Clustering in Ordered Dissimilarity Data

Timothy C. Havens,[1,*] James C. Bezdek,[1,†] James M. Keller,[1,‡]
Mihail Popescu[2,§]
[1]*Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211*
[2]*Health Management and Informatics Department, University of Missouri, Columbia, MO 65211*

This paper presents a new technique for clustering either object or relational data. First, the data are represented as a matrix **D** of dissimilarity values. **D** is reordered to $\mathbf{D}^*$ using a visual assessment of cluster tendency algorithm. If the data contain clusters, they are suggested by visually apparent dark squares arrayed along the main diagonal of an image $I(\mathbf{D}^*)$ of $\mathbf{D}^*$. The suggested clusters in the object set underlying the reordered relational data are found by defining an objective function that recognizes this blocky structure in the reordered data. The objective function is optimized when the boundaries in $I(\mathbf{D}^*)$ are matched by those in an aligned partition of the objects. The objective function combines measures of contrast and edginess and is optimized by particle swarm optimization. We prove that the set of aligned partitions is exponentially smaller than the set of partitions that needs to be searched if clusters are sought in **D**. Six numerical examples are given to illustrate various facets of the algorithm. © 2009 Wiley Periodicals, Inc.

1. INTRODUCTION

Consider a set of n objects $\mathbf{O} = \{o_1, \ldots, o_n\}$. The objects might be types of malignant tumors, genes expressed in a microarray experiment, vintage acoustic guitars, Cuban cigars, American motorcycles—virtually anything. We assume that there are subsets of similar objects in **O** (the clusters), but that each object bears no class label, that is, **O** is a set of unlabeled objects, and so, numerical representations of **O** are called *unlabeled data*.

Numerical *object* data associated with **O** has the form $\mathbf{X} = \{\vec{x}_1, \ldots, \vec{x}_n\} \subset \mathbb{R}^p$, where the coordinates of $\vec{x}_i$ provide *feature values* (e.g., weight, length, gene regulation, wrapper shape, number of strings, type of exhaust pipes, and so on) describing object o_i. The second data structure commonly used to represent the objects in

[*] Author to whom all correspondence should be addressed; e-mail: havenst@gmail.com.

[†] James Bezdek is visiting the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211, e-mail: jbezdek@uwf.edu.

[‡] e-mail: kellerj@missouri.edu.

[§] e-mail: popescum@missouri.edu.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 24, 504–528 (2009)
© 2009 Wiley Periodicals, Inc. Published online in Wiley InterScience
(www.interscience.wiley.com). • DOI 10.1002/int.20344

O is numerical *relational data*, which consist of n^2 similarities (or dissimilarities) between pairs of objects in **O**, represented by an $n \times n$ relational matrix $\mathbf{R} = [r_{ij} = \text{relation}(o_i, o_j) | 1 \leq i, j \leq n]$. We can always convert **X** into *dissimilarity* data $\mathbf{D} = \mathbf{D}(\mathbf{X})$, where $d_{ij} = ||\vec{x}_i - \vec{x}_j||$ is any vector norm on $\mathbb{R}^p$; therefore, most relational clustering algorithms are (implicitly) applicable to object data. In some sense, pairwise dissimilarity data represent the "most general" form of input data for cluster analysis; the most general example being, but certainly not the most common, a suite of sensors that supply numbers that become object data. However, there are both similarity and dissimilarity relational data sets that do not begin as object data and, for these, we have no choice but to use a relational clustering algorithm. We will refer to these two types of data as **X** and **D**, respectively. Good general references on clustering in both cases include the texts (See Refs. 1–7).

Clustering in unlabeled data **X** or **D** is the assignment of *labels* to the objects in **O** that are groups of similar items. The two necessary ingredients of all attempts to cluster in **X** or **D** are the number of groups to seek and (a model that encapsulates) some mathematical way to assess or assign similarity between the various objects. To consider possible solutions for the clustering problem, let c be the integer number of clusters. We include $c = 1$ and $c = n$ so that algorithms such as the SAHN clustering methods,[7] which begin or end with singleton clusters $c = n$ or the universal cluster $c = 1$ are included in the general discussion.[2] The crisp (that is, nonsoft) c-partitions of **X** are sets of cn values $\{u_{ik}\}$ that can be conveniently arrayed as a $c \times n$ matrix $\mathbf{U} = [u_{ik}]$. The set of all nondegenerate (no zero rows) c-partition matrices for **O** is

$$\mathbf{M}_{hcn} = \{\mathbf{U} \in \mathbb{R}^{cn} | u_{ik} \in \{0, 1\} \forall i, k; \sum_{i=1}^{c} u_{ik} = 1 \,\forall k; \sum_{k=1}^{n} u_{ik} > 0 \,\forall i\}, \qquad (1)$$

where u_{ik} is the *membership* of object o_k in cluster i—the partition element $u_{ik} = 1$ if o_k is labeled i and is 0 otherwise. There are three other kinds of labels—fuzzy, probabilistic, and possibilistic—that can be associated with each object and, for each kind, there are many clustering algorithms.[3] However, this article concerns only a subset of the type of partitions represented in (1), which we discuss in Section 4.

Here is a preview of the new method. The VAT algorithm (visual assessment of cluster tendency[8]) reorders the rows and columns of any $n \times n$ scaled dissimilarity matrix **D** with a modified version of Prim's minimal spanning tree algorithm.[9] We denote (any) reordering of **D** as **D***. If the image $I(\mathbf{D}^*)$ has c dark blocks along its main diagonal, this suggests that **D** contains c (as yet unfound) clusters. The size of each block may even indicate the approximate size of the suggested cluster. Hence, VAT images *suggest* both the number of and approximate members of object clusters, but VAT does not *find* the clusters. That is the aim of the method developed here. Specifically, the goal is to partition the objects underlying **D** and **D*** by optimizing an objective function designed to extract aligned clusters from the dark blocks in the image of the ordered dissimilarity matrix $I(\mathbf{D}^*)$.

The remainder of this article is structured thus. Section 2 gives a brief review of visual clustering and related work. Section 3 offers a short description of the VAT

Table I. Symbol definitions.

Symbol	Definition
n	No. of objects
c	No. of clusters
$\mathbf{O}$	Object data
$\mathbf{X}$	Numerical object data
$\mathbf{D}$	$n \times n$ Dissimilarity matrix
$\mathbf{D}^*$	Ordered dissimilarity matrix
$I(\mathbf{D}^*)$	Image of scaled $\mathbf{D}^*$
$\mathbf{U}$	$c \times n$ Partition matrix
$\mathbf{E}$	CLODD objective function value
α	CLODD mixing coefficient
γ	CLODD spline inflection set point
$\vec{m}^{(q)}$	qth Particle in PSO

algorithm, which is used to reorder **D**. Section 4 contains the main contribution of this work—the definition and analysis of the aligned partitioning model. Our method seeks *clusters in ordered dissimilarity data*, hence its acronym—CLODD. Section 5 gives a formal statement of CLODD and describes its optimization by *particle swarm optimization* (PSO).[10] Section 6 contains numerical examples illustrating the new approach. Section 7 summarizes our results and offers some ideas for interesting and useful extensions of this work. Table I contains a list of symbols used throughout this paper.

2. VISUAL APPROACHES TO CLUSTERING PROBLEMS

For object data, visual clustering was initially performed by inspecting scatterplots in $p = 1, 2,$ and 3 dimensions. For $p > 3$, scatterplots cannot be made. Many computational schemes have been devised to represent higher dimensional object data so that it can be visualized (and hence, possibly formed into clusters from visual representations). Interesting examples include Andrews plots,[11] Chernoff faces,[12] and Trees and Castles.[13] There are many other approaches and Refs. 14–17 contain informative introductions on many of these approaches.

For relational data **D**, scatterplots are unavailable. Tryon[14] apparently presented the first method for extracting clusters from dissimilarity data by use of a visual approach. Here is a rough description of his method; (i) plot a graph of each row in the data—a matrix of pairwise correlation coefficients, (ii) visually aggregate subsets of the graphs into clusters, (iii) reorder the input data matrix **D** so that similar profiles have adjacent representations in the rows and columns of the reordered data set **D***, (iv) find the mean profile (a prototype graph representing the elements of a group) for each cluster of correlation profiles, and (v) present the final results as a set of clustered profile graphs with their prototypes. This procedure—almost 70 years old—contains all the elements of the current work on visual clustering: create a visual representation of **D**, reorder it to **D***, create a visual representation **D***, and, finally, extract clusters from $\mathbf{D}^*$ using the visual evidence. Tryon did this by hand in 1939 for a 20×20 data set collected at the University of California, Berkeley. For tiny data sets, methods such as this are useful. But for the data sets we typically encounter today, automation is essential.

In the decades subsequent to Tryon's work, the literature has included many visual schemes for each of the three main problems in cluster analysis: tendency, partitioning, and validity. Using $\mathbf{D}$ and $\mathbf{D}^*$ in various ways for any of the three clustering problems involves two basic issues: *finding* $\mathbf{D}^*$ (how shall we reorder $\mathbf{D} \to \mathbf{D}^*$?), and *displaying* $\mathbf{D}^*$ (how shall we "see" the information in $\mathbf{D}^*$?). The three problems and two principles have appeared in almost every combination.

Sneath introduced the idea of visual representation of $\mathbf{D}^*$ by an image in 1957.[18] Sneath's paper contains an image $I(\mathbf{D}^*)$ of $\mathbf{D}^*$ created by handshading the pixels of a matrix with one of eight "intensities"—reordering was done by an algorithm that had both computer and manual components. Subsequent refinements of his idea followed the general evolution of computers themselves. In 1963, Floodgate and Hayes[19] presented a hand-rendered image similar to Sneath's, but reordering of $\mathbf{D}$ was done computationally using single linkage clustering. Apparently Ling[20] was the first to automate the creation of the image $I(\mathbf{D}^*)$ with an algorithm called SHADE, which was used after application of the complete linkage hierarchical clustering scheme and served as an alternative to visual displays of hierarchically nested clusters via the standard dendrogram. SHADE used 15 level halftone intensities (created by overstriking standard printed characters) to approximate a digital representation of the lower triangular part of the reordered dissimilarity matrix. SHADE apparently represents the first completely automated approach to finding $\mathbf{D}^*$ and viewing $I(\mathbf{D}^*)$.

Closely related to SHADE, but presented more in the spirit of finding rather than displaying clusters found with a relational clustering algorithm, is the "graphical method of shading" described by Johnson and Wichern.[7] They provide this informal description: (i) arrange the pairwise distances between points in the data into several classes of 15 or fewer, based on their magnitudes, (ii) replace all distances in each class by a common symbol with a certain shade of gray, (iii) reorganize the distance matrix so that items with common symbols appear in contiguous locations along the main diagonal (darker symbols correspond to smaller distances), and (iv) identify groups of similar items by the corresponding patches of dark shadings. A more formal approach to this problem is the work of Tran-Luu,[21] who proposed reordering the data into an "acceptable" block form based on optimizing several mathematical criteria of image "blockiness." The reordered matrix is then imaged, and the number of clusters is deduced visually by a human observer.

Software for visualizing distance data is available at the GENLAB toolbox Web site.[22] Similarity-based intensity images, formed using kernel functions, were used in Refs. 23 and 24 to provide guidance in determining the number of clusters (tendency assessment, in spirit of the VAT algorithm), but no useful ordering scheme is offered there to facilitate the approach. Other representative studies include Refs. 25–29. Visual cluster validity includes the work presented in Refs. 30 and 31.

The main difference between the algorithms and methods described in this section and CLODD is that CLODD is a completely autonomous method for determining cluster tendency, extracting clusters from the image of the reordered dissimilarity data, and providing a cluster validity metric, as well. This leads to a distinct advantage of CLODD; namely, that CLODD is not tied directly to any one distance metric or reordering scheme. CLODD requires, as input, only an image of reordered dissimilarity data, such that the clusters appear as dark blocks along the diagonal.

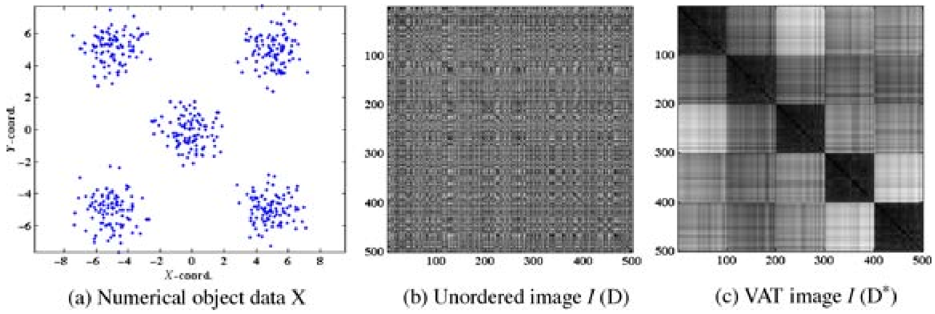
3. THE VAT IMAGE

The VAT algorithm displays an image of reordered and scaled dissimilarity data.[8] Each pixel of the grayscale VAT image $I(\mathbf{D}^*)$ displays the scaled dissimilarity value of two objects. White pixels represent high dissimilarity, whereas black represents low dissimilarity. Each object is exactly similar with itself, which results in zero-valued (black) diagonal elements of $I(\mathbf{D}^*)$. The off-diagonal elements of $I(\mathbf{D}^*)$ are scaled to the range [0, 1]. A dark block along the diagonal of the $I(\mathbf{D}^*)$ is a submatrix of "similarly small" dissimilarity values; hence, the dark block represents a cluster of objects that are relatively similar to each other. Thus, the cluster tendency is shown by the number of dark blocks along the diagonal of the VAT image. Algorithm 1 illustrates the steps of the VAT algorithm, where arg min and arg max in Equations 2 and 3 are set-valued.

Algorithm 1 VAT Ordering Algorithm[8]

```
Input: $\mathbf{D}$ - dissimilarity matrix
Data: $\mathbf{K} = \{1, 2, \ldots, n\}$; $\mathbf{I} = \mathbf{J} = \emptyset$; $P = (0, 0, \ldots, 0)$.
Select
    $$(i, j) \in \arg\max_{p\in\mathbf{K}, q\in\mathbf{K}} D_{pq}. \quad (2)$$
Set $P(1) = i$; $\mathbf{I} = \{i\}$; and $\mathbf{J} = \mathbf{K} - \{i\}$.
for $r = 2, \ldots, n$ do
    Select
        $$(i, j) \in \arg\max_{p\in\mathbf{I}, q\in\mathbf{J}} D_{pq}. \quad (3)$$
    Set $P(r) = j$; Replace $\mathbf{I} \leftarrow \mathbf{I} \cup \{j\}$ and $\mathbf{J} \leftarrow \mathbf{J} - \{j\}$.
Obtain the ordered dissimilarity matrix $\mathbf{D}^*$ using the ordering array $P$ as: $D^*_{pq} = D_{P(p),P(q)}$, for $1 \leq p, q \leq n$.
```

Figure 1a is a scatterplot of $n = 1000$ data points in $\mathbb{R}^2$ drawn from a mixture of five normal distributions. The means, mixing proportions, and number of samples in each cluster (i.e., the cardinality n_i, $i = 1, 2, 3, 4, 5$) are listed in Table II. The covariance matrices are $\Sigma_1 = \Sigma_2 = \Sigma_3 = \Sigma_4 = \Sigma_5 = \sigma^2 I$, where I is the 2×2 identity matrix $\mathbf{D}$ by computing $d_i j = ||\vec{x}_i = \vec{x}_j||$ with the Euclidean norm. The $c = 5$ visually apparent clusters in Figure 1a are suggested by the five distinct dark



(a) Numerical object data X (b) Unordered image I (D) (c) VAT image I (D*)

Figure 1. Example of how VAT image suggests cluster tendency by the number of dark blocks along diagonal

Table II. Data set **X** shown in scatterplot of Figure 1a.

Mean	Mixing proportions	n_i
$\mu_1 = (0, 0)$	$\alpha_1 = 0.21$	225
$\mu_1 = (8, 8)$	$\alpha_1 = 0.21$	203
$\mu_1 = (16, 0)$	$\alpha_1 = 0.21$	197
$\mu_1 = (0, 16)$	$\alpha_1 = 0.21$	200
$\mu_1 = (16, 16)$	$\alpha_1 = 0.16$	175

diagonal blocks in Figure 1c, which is the VAT image $I(\mathbf{D}^*)$ of the data after VAT reordering of $\mathbf{D}$ to $\mathbf{D}^*$. Comparing this to view 1b, which is the image $I(\mathbf{D})$ of the dissimilarities in input order, it is clear that reordering is essential to reveal the structure of the underlying data. The fact was clear to Tryon[14] in 1939 and to Sneath[18] in 1957, but our ability to process and display information of this kind is, of course, quite a bit better than that which was available to those early pioneers of visual clustering methods.

VAT in its original form was limited to approximately $n = 5000$, and was $\mathbf{O}(n^2)$. A scalable version of VAT (sVAT)[32] removes the size limitation and reduces the complexity to $\mathbf{O}(n)$. A rectangular version of VAT (coVAT)[33] yields images like that in Figure 1c from nonsquare relational data and is also scalable to arbitrary sized data sets. Three questions associated with the VAT-based methods of finding and displaying $\mathbf{D}^*$ (or $I(\mathbf{D}^*)$ are

1. (Q1) How closely related is $I(\mathbf{D}^*)$ to image representations of single linkage clusters? The fact that *single linkage* (SL) clusters can be realized by cutting a *minimal spanning tree* (MST) in $\mathbf{D}$, coupled with the fact that VAT reorders $\mathbf{D}$ with a modification of Prim's MST algorithm[9] suggests that there is a close relationship. We also know that both VAT and SL can fail: Do these failures occur in the same circumstances, and is there a property of $\mathbf{D}$ that would enable us to at least be wary of failures? Consideration of these two issues is nearly a paper unto itself and would take us far afield from our present objective; hence, this question is taken up in Refs. 34 and 35.
2. (Q2) Can we automatically extract c, the number of clusters to look for, as suggested by the visual evidence in $I(\mathbf{D}^*)$ without looking at the visual display? This problem is driven

by a desire to capitalize on the information possessed by the VAT image without actually having to view it. For even loadable values of n, $I(\mathbf{D}^*)$ becomes difficult, if not impossible to actually display. Moreover, different viewers may have different opinions, making this a somewhat subjective method in exactly the cases where it is most important to be correct (i.e., cases where the clusters are not sharply delineated). Two papers provide positive answers for this second question. The CCE[36] and DBE[37] algorithms extract the number of apparent clusters from VAT images using similar image-processing approaches that differ mainly in the details of the image processing itself. But these two methods stop short of answering the last question.

3. (Q3) Can we automatically extract $\mathbf{U}$, a crisp c-partition of $\mathbf{O}$, as suggested by the visual evidence in $I(\mathbf{D}^*)$? This last question has, to our knowledge, not been answered and forms the basis for the rest of this article. The algorithm developed in the next section answers (Q3) and, as a bonus, provides a third approach for addressing (Q2) as well.

4. PARTITIONING OBJECTS REPRESENTED BY A BLOCK DIAGONAL MATRIX

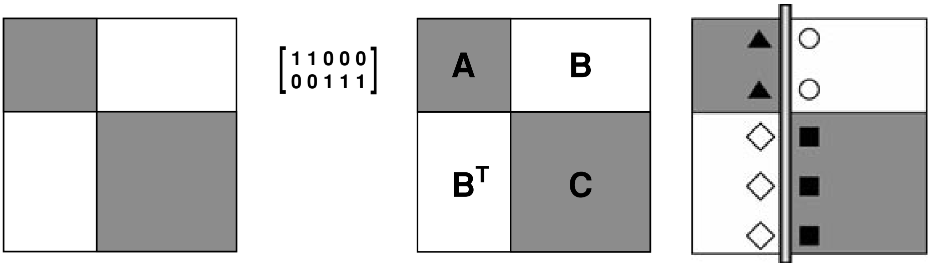
We assume as input a normalized (entries between 0 and 1) dissimilarity matrix $\mathbf{D}^*$—equivalently, $I(\mathbf{D}^*)$ — that is symmetric with diagonal elements that are zero. The superscript (*) indicates that $\mathbf{D}$ has been reordered by some algorithm to produce a "VAT-like" image, as in Figure 1. The important property of $I(\mathbf{D}^*)$ is that it has, beginning in the upper left corner, dark blocks along its diagonal. Accordingly, we constrain our search through $\mathbf{M}_{hcn}$ for each c under consideration to those partitions that mimic the blocky structure in $I(\mathbf{D}^*)$. We call these partitions, $\mathbf{U} \subsetneq \mathbf{M}_{hcn}$, *aligned* partitions. Aligned c-partitions of $\mathbf{O}$ have c contiguous blocks of 1s in $\mathbf{U}$, ordered to begin with the upper left corner and proceeding down and to the right. The set of all aligned c-partitions is

$$\mathbf{M}^*_{hcn} = \{\mathbf{U} \in \mathbf{M}_{hcn} | u_{1k} = 1, 1 \le k \le n_1 : u_{ik} = 1, n_{i-1} \le k \le n_i, 2 \le i \le c\}. \quad (4)$$

For example, $\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$ are aligned partitions, whereas $\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$, $\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$, and $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$ are not.

The special nature of aligned partitions enables us to specify them in an alternative form. Every member of $\mathbf{M}^*_{hcn}$ is isomorphic to the unique set of c distinct integers (which are the cardinalities of the c clusters in $\mathbf{U}$) that satisfy $\{n_i | 1 \le n_i; 1 \le i \le c; \sum_{i=1}^{c} n_i = n\}$, so *aligned* partitions are completely specified by $\{n_1 : \ldots : n_c\}$. For example,

$$\mathbf{U} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \{2 : 1 : 2\}. \quad (5)$$



(a) Ideal I (D*)

(b) Optimal partition U ∈ M^*_{h25}

(c) E_{sq} (U) "squareness"

(d) E_{edge}(U) "edginess"

Figure 2. The components of CLODD objective function $E(\mathbf{U})$.

The important characteristics of $I(\mathbf{D}^*)$ that we shall exploit for finding a **U** that seems to match it are (i) the contrast between the dark blocks along the main diagonal and the lighter off-diagonal blocks and (ii) the visually apparent edges of those dark blocks. Our algorithm generates candidate partitions in $\mathbf{M}^*_{hcn}$ and tests their fit to the clusters suggested by the aligned dark blocks in $I(\mathbf{D}^*)$. To accomplish this, we define an objective function on $\mathbf{M}^*_{hcn}$ that computes a measure of two properties of blocky images $I(\mathbf{D}^*)$—"squareness" and "edginess". Figure 2a shows an idealized case of $I(\mathbf{D}^*)$ for $c = 2$ which, for illustration purposes, assumes that $n = 5$.

Figure 2b shows the presumably optimal aligned partition that provides the best fit to the image in 2a. Figure 2c shows the "squareness" component of the objective function that measures the contrast between diagonal dark blocks **A** and **C** and the off-diagonal blocks **B** and $\mathbf{B}^\mathbf{T}$ according to the **U** in 2b. An intuitively appealing measure is the difference of the average dissimilarity values *between* apparent clusters (i.e., dissimilarities in [(**A**,**B**)] and [($\mathbf{B}^\mathbf{T}$,**C**)]) and those *within* apparent clusters (i.e., dissimilarities in [(**A**,**A**)] and [(**C**,**C**)]). Let **U** be a candidate partition in $\mathbf{M}^*_{hcn}$; let $\{\mathbf{O}_i : 1 \le i \le c\}$ be the crisp c-partition of **O** corresponding to **U**. The cardinality $|\mathbf{O}_i| = n_i \forall i$, and we abbreviate the membership $o_s \in \mathbf{O}_i$ simply as $s \in i$. With these heuristics, the "squareness" component of the objective function for a given $\mathbf{D}^*$ is

$$E_{\text{sq}}(\mathbf{U};\mathbf{D}^*) = \underbrace{\left(\frac{\sum_{i=1}^{c}\sum_{s\in i,t\notin i} d^*_{st}}{\sum_{i=1}^{c}(n-n_i)n_i}\right)}_{\substack{\text{ave. dissimilarity } \textit{between} \text{ dark}\\ \text{and non-dark regions in } I(\mathbf{D}^*)}} - \underbrace{\left(\frac{\sum_{i=1}^{c}\sum_{s,t\in i,s\neq t} d^*_{st}}{\sum_{i=1}^{c}(n_i^2-n_i)}\right)}_{\substack{\text{ave. dissimilarity } \textit{within}\\ \text{dark regions in } I(\mathbf{D}^*)}} . \tag{6}$$

Good candidate partitions **U** should maximize Equation 6. This equation is a measure of contrast between the on-diagonal dark blocks and the off-diagonal nondark blocks.

The "edginess" of the dark blocks in $\mathbf{D}^*$ is computed by averaging the values of the first-order estimate of the horizontal digital gradient across each vertical boundary imposed by a candidate $\mathbf{U}$ in $\mathbf{M}^*_{hcn}$. Figure 2d shows the edges that are considered for this part of the objective function. The symbols along the vertical boundary separating the dark from the nondark blocks represent dissimilarity values in the columns of $\mathbf{D}^*$ adjacent to the boundary. The "edginess" value for the example in 2d is computed by

$$E_{\text{edge}}(\mathbf{U}) = \left(\frac{\sum | \bigcirc - \triangle | + \sum | \Diamond - \Box |}{2+3} \right).$$

For the c blocks in $\mathbf{D}^*$, there are $(c-1)$ interior vertical boundaries between dark blocks and adjacent blocks of lighter intensities. Each vertical edge spans the right face of an upper block and the left face of the block immediately below it. Let $\mathbf{U} = \{n_1 : \ldots : n_c\} \in \mathbf{M}^*_{hcn}$, a candidate-aligned partition. For $j = 1$ to $c-1$, let $m_j = \sum_{k=1}^{j} n_k$, and $m_0 = 1$. We defined the "edginess" measure as

$$E_{\text{edge}}(\mathbf{U}; \mathbf{D}^*) = \frac{1}{c-1} \sum_{j=1}^{c-1} \frac{\displaystyle\sum_{i=m_{j-1}}^{m_j} |d^*_{i,m_j} - d^*_{i,m_j+1}| + \sum_{i=m_j+1}^{m_{j+1}} |d^*_{i,m_j} - d^*_{i,m_j+1}|}{n_j + n_{j+1}}. \tag{7}$$

Good candidate partitions $\mathbf{U}$ should maximize Equation 7. Although this equation looks complicated, it is merely the average horizontal gradient across vertical edges separating dark blocks from nondark blocks in $I(\mathbf{D}^*)$. Good candidate partitions $\mathbf{U}$ maximize both Equations 6 and 7, which allows us to add them together to produce a composite objective function. To make the resulting sum flexible in terms of the balance between contrast and edginess, we use the convex combination of Equations 6 and 7. Let α be the *mixing coefficient*, and

$$E_\alpha(\mathbf{U}; \mathbf{D}^*) = \alpha E_{\text{sq}}(\mathbf{U}; \mathbf{D}^*) + (1-\alpha) E_{\text{edge}}(\mathbf{U}; \mathbf{D}^*); 0 \le \alpha \le 1. \tag{8}$$

If contextual information is unavailable to suggest that one factor, contrast or edginess, is more important than the other, one may take $\alpha = 1/2$, which gives equal weight to contrast and edginess in $\mathbf{D}^*$.

The final component of the objective function controls the size of the smallest cluster allowed in the search over $\mathbf{M}^*_{hcn}$. We use the spline function,

$$s(x, a) = \begin{cases} 0 & ; x \le 1 \\ 2\left(\frac{x}{a}\right)^2 & ; 1 < x \le \frac{a}{2} \\ 1 - 2\left(\frac{a-x}{a}\right)^2 & ; \frac{a}{2} < x < a \\ 1 & ; a \le x \end{cases}, \tag{9}$$

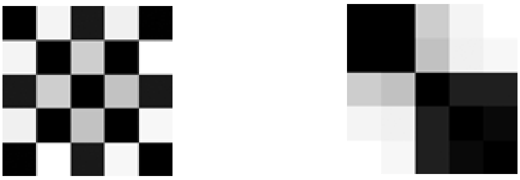
for this purpose. This function is a typical s-curve valued in [0, 1] with points of inflection at $a/2$ and a. For $\mathbf{U} = \{n_1 : \ldots : n_c\} \in \mathbf{M}^*_{hcn}$, we set the inflection points by choosing $a = \gamma n$, $2/n < \gamma < 1$, and then evaluate s at $x = \min_{1 \le i \le c}\{n_i\}$.

$$\mathbf{D} = \begin{bmatrix} 0 & 0.73 & 0.19 & 0.71 & 0.16 \\ 0.73 & 0 & 0.59 & 0.12 & 0.78 \\ 0.19 & 0.59 & 0 & 0.55 & 0.19 \\ 0.71 & 0.12 & 0.55 & 0 & 0.74 \\ 0.16 & 0.78 & 0.19 & 0.74 & 0 \end{bmatrix}$$

(a) Dissimilarity matrix D

$$\mathbf{D} = \begin{bmatrix} 0 & 0.12 & 0.59 & 0.73 & 0.78 \\ 0.12 & 0 & 0.55 & 0.71 & 0.74 \\ 0.59 & 0.55 & 0 & 0.19 & 0.19 \\ 0.73 & 0.71 & 0.19 & 0 & 0.16 \\ 0.78 & 0.74 & 0.19 & 0.16 & 0 \end{bmatrix}$$

(b) VAT reordered dissimilarity matrix D*



(c) Image I (D)

(d) VAT image I (D*)

Figure 3. Dissimilarity data used in CLODD example 1.

Finally, we multiply the function in Equation 8 by

$$S_\gamma(\mathbf{U}) = s\left(\min_{1\le i\le c}\{n_i\}, \gamma n\right). \tag{10}$$

This scales Equation 8 in a way that enables us to damp very small clusters in candidates partitions when none are apparent in $\mathbf{D}^*$. The objective function is now complete, so we define an optimal partition of $\mathbf{D}^*$ as one that maximizes

$$E(\mathbf{U};\mathbf{D}^*) = s\left(\min_{1\le i\le c}\{n_i\}, \gamma n\right)\cdot E_\alpha(\mathbf{U};\mathbf{D}^*) = S_\gamma(\mathbf{U})\cdot E_\alpha(\mathbf{U};\mathbf{D}^*). \tag{11}$$

Finally, we want to search for the best partition at various values of c, so let $\mathbf{C} = \{2, 3, \ldots, c_{\max}\}$. The optimization problem that the CLODD algorithm attempts to solve is

$$\max_{\mathbf{U}\in\mathbf{M}^*_{hcn}, c\in\mathbf{C}}\{E(\mathbf{U};\mathbf{D}^*)\} \tag{12}$$

We denote an approximate global solution of Equation 12 by $\mathbf{U}_{c^*}$. We need to choose two model parameters (α, γ), and then solve the optimization problem in Equation 12. Before we turn to the solution of Equation 12, we give an example that illustrates the basic ideas of this approach.

Example 1. Shown in Figures 3a and 3b are a matrix $\mathbf{D}$ and the image $I(\mathbf{D})$ of dissimilarities between five objects $\mathbf{O} = \{o_1, \ldots, o_5\}$. Figures 3c and 3d show the VAT reordering $\mathbf{D}^*$ of $\mathbf{D}$, and the VAT image $I(\mathbf{D}^*)$ corresponding to this reordering.

Visual inspection of $I(\mathbf{D})$ does not reveal whether the objects represented by pairwise dissimilarities in $\mathbf{D}$ might form clusters in $\mathbf{O}$. In addition, it is easy to see that cluster structure *is* suggested by the two dark blocks in the VAT image $I(\mathbf{D}^*)$. The strong impression given by $I(\mathbf{D}^*)$ is that this is an instance for which the ideal case is shown in Figure 2a. Thus, the aligned 2-partition of $\mathbf{O}$ that should provide a best match to $I(\mathbf{D}^*)$ is the one shown in Figure 2b corresponding to $o = \{o_1^*, o_2^*\} \cup \{o_3^*, o_4^*, o_5^*\}$. At this point, VAT has done its job. We could apply CCE[36] or DBE[37] to $I(\mathbf{D}^*)$, and those algorithms would return the value $c = 2$, telling us to look for two clusters in $\mathbf{O}$. Despite this, these algorithms (VAT, CCE, and/or DBE) still have not defined cluster partitions. To obtain the $\mathbf{U}$ in Figure 2b that is suggested by $I(\mathbf{D}^*)$, we apply CLODD to $\mathbf{D}^*$.

To see how the CLODD objective function $E(\mathbf{U}; \mathbf{D}^*)$ compares candidates, consider the aligned 2-partitions,

$$\mathbf{U} = \{2:3\} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

and

$$\mathbf{V} = \{3:2\} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix},$$

and their transformations under f and g,

$$f(\mathbf{U}) = \mathbf{U}^T\mathbf{U} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix};$$

$$g(\mathbf{U}) = [1] - f(\mathbf{U}) = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}, \tag{13}$$

$$f(\mathbf{V}) = \mathbf{V}^T\mathbf{V} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix};$$

$$g(\mathbf{V}) = [1] - f(\mathbf{V}) = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}. \tag{14}$$

$$\mathbf{U} \leftrightarrow \{2:3\} \in \mathbf{M}^*_{h25} \qquad\qquad \mathbf{U} \leftrightarrow \{3:2\} \in \mathbf{M}^*_{h25}$$

$$\mathbf{D}^* = \left[\begin{array}{cc|ccc} 0 & 0.12 & 0.59 & 0.73 & 0.78 \\ 0.12 & 0 & 0.55 & 0.71 & 0.74 \\ \hline 0.59 & 0.55 & 0 & 0.19 & 0.19 \\ 0.73 & 0.71 & 0.19 & 0 & 0.16 \\ 0.78 & 0.74 & 0.19 & 0.16 & 0 \end{array}\right] \qquad \mathbf{D}^* = \left[\begin{array}{ccc|cc} 0 & 0.12 & 0.59 & 0.73 & 0.78 \\ 0.12 & 0 & 0.55 & 0.71 & 0.74 \\ 0.59 & 0.55 & 0 & 0.19 & 0.19 \\ \hline 0.73 & 0.71 & 0.19 & 0 & 0.16 \\ 0.78 & 0.74 & 0.19 & 0.16 & 0 \end{array}\right]$$

Figure 4. Boundaries imposed on $\mathbf{D}^*$ by choosing $\mathbf{U} = \{n_1 : n_2\} \in \mathbf{M}^*_{h25}$.

The blocks of 1s in $f(\mathbf{U}) = \mathbf{U}^T\mathbf{U}$ and $f(\mathbf{V}) = \mathbf{V}^T\mathbf{V}$ show the regions in $\mathbf{D}^*$ over which the CLODD calculations are made (as do $g(\mathbf{U})$ and $g(\mathbf{V})$). The partition parameters $\{2 : 3\}$ and $\{3 : 2\}$ set up "boundaries" in $\mathbf{D}^*$ as shown in Figure 4.

For this example, Equations (6) and (7) yield,

$$\begin{aligned} E_{\text{sq}}(\mathbf{U};\mathbf{D}^*) &= (0.59 + 0.73 + 0.78 + 0.55 + 0.71 + 0.74)/6 \\ &\qquad -(0.12 + 0.19 + 0.19 + 0.16)/4 = 0.52, \\ E_{\text{sq}}(\mathbf{V};\mathbf{D}^*) &= (0.73 + 0.78 + 0.71 + 0.74 + 0.19 + 0.19)/6 \\ &\qquad -(0.12 + 0.59 + 0.55 + 0.1)/4 = 0.20, \end{aligned}$$

$$\begin{aligned} E_{\text{edge}}(\mathbf{U};\mathbf{D}^*) &= [|0.12 - 0.59| + |0 - 0.55| \\ &\qquad +|0.55 - 0| + |0.71 - 0.19| + |0.74 - 0.19|]/5 \\ &= 0.53, \end{aligned}$$

$$\begin{aligned} E_{\text{edge}}(\mathbf{U};\mathbf{D}^*) &= [|0.59 - 0.73| + |0.55 - 0.71| + |0 - 0.19| \\ &\qquad +|0.19 - 0| + |0.19 - 0.16|]/5 \\ &= 0.14. \end{aligned}$$

In this example the smallest $n_i = 2$ and $n = 5$ for both $\mathbf{U}$ and $\mathbf{V}$, so the spline factor in Equation 9 has the same value for any choice of γ; without loss we take $S_\gamma(\mathbf{U}) = 1$. Choosing $\alpha = 0.5$ in Equation 11, we arrive at the final values,

$$E(\mathbf{U};\mathbf{D}^*) = E_{0.5}(\mathbf{U};\mathbf{D}^*) = (0.52 + 0.53)/2 = 0.53,$$

$$E(\mathbf{V};\mathbf{D}^*) = E_{0.5}(\mathbf{V};\mathbf{D}^*) = (0.20 + 0.14)/2 = 0.17.$$

For the two candidates **U** and **V**, our expectation is correct: E clearly favors **U** to **V**, that is, $\mathbf{U}_{2^*} = \mathbf{U}$. □

The objective function $E(\mathbf{U}; \mathbf{D}^*)$ is always valued in [0, 1]. $E(\mathbf{U}; \mathbf{D}^*) = 0$ if and only if $I(\mathbf{D}^*)$ has only one intensity, which can occur if and only if $\mathbf{D}^*$ has all zero-valued off-diagonal elements. $E(\mathbf{U}; \mathbf{D}^*) = 1$ if and only if $I(\mathbf{D}^*)$ has c perfect (i.e., zero-valued intensities) diagonal blocks with all other off-diagonal intensities equal to 1. If the diagonal blocks in Figure 2a were pure black, then the partition in Figure 2b would result in $E(\mathbf{U}; \mathbf{D}^*) = 1$. $\mathbf{M}^*_{hcn}$ is finite, and much smaller than the finite set $\mathbf{M}_{hcn}$, but how much smaller? The following proposition answers this question.

PROPOSITION 1. *The cardinality of* $\mathbf{M}^*_{hcn}$*, the set of aligned c-partitions of n objects into* $2 \le c < n$ *crisp subsets in* $\mathbf{M}_{hcn}$*, is*

$$|\mathbf{M}^*_{hcn}| = \binom{n-1}{c-1}. \tag{15}$$

Proof. Recall that aligned partitions can be completely specified by $\{n_1 : \ldots : n_c\}$. Hence, the cardinality of $\mathbf{M}^*_{hcn}$ is equal to the cardinality of $\{n_1 : \ldots : n_c\}$, under the constraints

$$n_i \in \mathbb{Z}; 1 \le n_i \le (n - c + 1) \forall i; \sum_{i=1}^{c} n_i = n. \tag{16}$$

Consider n_i to be the number of marbles in a bag or container, where there are c bags. You are given n marbles to put in those bags under the constraint that you must place at least one marble in each bag and you cannot be left with any marbles. How many different ways could you place the marbles in the bags? Solving this problem is equivalent to proving Proposition 1.

Begin by placing one marble in each bag. There are $(n - c)$ marbles left over. Hence, the maximum number of marbles that could be in any one bag is $(n - c + 1)$. Now, choose a bag at random and add one marble to its contents. Continue until all marbles are placed. Thus, you have c objects (bags) to choose from and you choose $(n - c)$ times. The order does not matter, and the objects (bags) can be chosen more than once. Thus, this is a well-known combinatorics problem, where we are choosing an unordered sample of size $(n - c)$ with repetition from a population of c elements.[38] The number of combinations is the value of the binomial coefficient,

$$\binom{c + (n - c) - 1}{c - 1} = \binom{n-1}{c-1}, \tag{17}$$

which is Equation 15. ■

Remark 1. For $c << n$, $|\mathbf{M}^*_{hcn}| \approx n^{c-1}/(c-1)!$ is a good approximation to the exact value in Equation 15. The exact cardinality of $\mathbf{M}_{hcn}$ is known, $|\mathbf{M}_{hcn}|$

$= \frac{1}{c!}\sum_{j=1}^{c}\begin{pmatrix} c \\ j \end{pmatrix}(-1)^{c-j} j^n$. For $c << n$, the last term dominates this sum, and the approximation $|\mathbf{M}_{hcn}| \approx c^n/c!$ can be used. It is instructive to compare the size of $\mathbf{M}^*_{hcn}$ to that of $\mathbf{M}_{hcn}$, by the ratio

$$\frac{|\mathbf{M}^*_{hcn}|}{|\mathbf{M}_{hcn}|} \approx \frac{n^{c-1}/(c-1)!}{c^n/c!} = \left(\frac{n^{c-1}}{(c-1)!}\right)\left(\frac{c!}{c^n}\right) = \frac{n^{c-1}}{c^{n-1}},\ c << n. \tag{18}$$

Applying this ratio for the fairly typical problem of $c = 10$ and $n = 10,000$ yields $|\mathbf{M}^*_{hcn}|/|\mathbf{M}_{hcn}| \approx 1/10^{9963}$—a *very* small number. This shows that algorithms that search for a crisp partition of **D** over $\mathbf{M}^*_{hcn}$ have a significantly smaller set of solutions to examine. We note, however, that the size of $\mathbf{M}^*_{hcn}$ is still quite large: for $c = 10$ and $n = 10,000$, $|\mathbf{M}^*_{hcn}| \approx 10^{36}/9! = 2.7557 \times 10^{30}$. Hence, even though $\mathbf{M}^*_{hcn}$ is relatively small, it is still far too big for exhaustive search. This leads us to methods for approximating a solution to Equation 12, which is the topic we turn to next.

5. PARTICLE SWARM OPTIMIZATION AND THE CLODD ALGORITHM

We stress that, in principle, any number of optimization algorithms could be used. We use *particle swarm optimization* (PSO)[10] because it is simple, and because it has been shown to be relatively successful at optimizing highly modal nonlinear objective functions. For a given c in **C**, each candidate $\mathbf{U} = \{n_1 : \ldots : n_c\} \in \mathbf{M}^*_{hcn}$ is completely specified by the c integer indices $\{n_1 : \ldots : n_c\}$, which in turn can be used to specify the locations along the columns of $\mathbf{D}^*$ where trial boundaries are matched to the boundaries in $\mathbf{D}^*$. The integers $m_j = \sum_{k=1}^{j} n_k$, $j = 1, 2, \ldots, t-1$, are the locations of the right edges (boundaries) of the first $t-1$ blocks in $\mathbf{D}^*$—the right edge of the last block is at location $m_n = n$, which is the right edge of the matrix or image of the matrix. Because we can recover the c integer $\{n_i\}$ from the $c-1$ integers $\{m_i\}$, we write $\mathbf{U} = \{n_1 : \ldots : n_c\} = \vec{m} \in \mathbf{M}^*_{hcn}$. The vector $\vec{m} = (m_1, \ldots, m_{t-1}) \in \mathbb{R}^{t-1}$ plays a central role in CLODD.

Fix $c = t$. Let $\mathbf{U}_{it} = \{n_{i1} : \ldots : n_{ic}\} \in \mathbf{M}^*_{htn}$. Construct the vector $\vec{m}_{it} = (m_{i1}, \ldots, m_{i(t-1)}) \in \mathbb{R}^{t-1}$. This vector of $t-1$ integers has strictly increasing components, $m_{i1} < m_{i2} < \cdots < m_{i(t-1)}$, that specify the $t-1$ locations of the interior boundaries imposed on $\mathbf{D}^*$ by $\mathbf{U}_{it}$. The vector $\vec{m}_{it}$ is thought of as a particle having velocity $\vec{v}_{it} = (v_{i1}, \ldots, v_{i(t-1)}) \in \mathbb{R}^{t-1}$. Let N_p be the number of particles—the number of trial partitions of **O**—in each swarm, where each swarm represents a different choice of the number of clusters t. Let $\hat{m}_{it}$ denote the current best position of each particle in swarm t, let $\hat{\hat{m}}_t$ denote the current best position of all N_p particles in swarm t, and let $\hat{\hat{G}}$ be the best particle over all swarms. In our specification, $\mathbf{rand}([a, b])$ is a random vector, each component distributed uniformly on $[a, b]$. With these conventions, we are ready to state the CLODD algorithm, displayed in Algorithm 2.

Algorithm 2 CLODD: Extraction of clusters from ordered dissimilarity data

```
Input: An $n \times n$ matrix of ordered (from, e.g., VAT) dissimilarities,
$\mathbf{D}^* = [d^*_{ij}]; \forall i, j : 0 \le d^*_{ij} \le 1, d^*_{ij} = d^*_{ji}, d^*_{ii} = 0$.
Parameters:
$\mathbf{C} = \{2, 3, \ldots, c_{max}\}$ = range of values for search over $\mathbf{M}^*_{hcn}$
$N_p$ = no. of particles for each swarm $c \in \mathbf{C}$
$\alpha$ = mixing coefficient for $E_\alpha(\mathbf{U}; \mathbf{D}^*)$, $0 \le \alpha \le 1$
$\gamma$ = set point control for $S_\gamma(\mathbf{U})$, $2/n < \gamma \le 1$
$q_{max}$ = maximum number of swarm iterations
$\epsilon$ = threshold multiplier
$\epsilon_c = \epsilon N_p(c-1), c \in \mathbf{C}$ = termination threshold at each value of $c$
PSO parameters:
$K$ = inertial constant, $0 < K < 1$
$A_{local}$ = local influence constant, $0 < A_{local} < 4$
$A_{global}$ = global influence constant, $0 < A_{global} < 4$
Main Loop:
1  for $t = 2$ to $c_{max}$ do
2      Initialize particles, $(i, t), i = 1, 2, \ldots, N_p$
3      for $q = 1$ to $q_{max}$ do
4          for $i = 1$ to $N_p$
5              if $\vec{m}^{(q)}_{it}$ produces a valid partition then
6                  Build the partition matrices $\mathbf{U}^{(q)}_{it}$, $\hat{\mathbf{U}}_{it}$, and $\hat{\hat{\mathbf{U}}}_t$ equivalent to
                   $\vec{m}^{(q)}_{it}, \hat{m}_{it}, \hat{\hat{m}}_t$
7                  if $E(\mathbf{U}^{(q)}_{it}) > E(\hat{\mathbf{U}}_{it})$ then $\hat{m}_{it} = \vec{m}^{(q)}_{it}$
8                  if $E(\mathbf{U}^{(q)}_{it}) > E(\hat{\hat{\mathbf{U}}}_t)$ then $\hat{\hat{m}}_t = \vec{m}^{(q)}_{it}$
9              $\vec{v}^{(q+1)}_{it} = K\vec{v}^{(q)}_{it} + A_{local} \cdot \mathbf{rand}([0, 1])(.*)(\hat{m}_{it} - \vec{m}^{(q)}_{it}) +$
               $A_{global} \cdot \mathbf{rand}([0, 1])(.*)(\hat{\hat{m}}_t - \vec{m}^{(q)}_{it})$
10             $\vec{m}^{(q+1)}_{it} = \mathrm{Round}(\vec{m}^{(q)}_{it} + \vec{v}^{(q+1)}_{it})$
11             CLIP $\vec{m}^{(q+1)}_{it}$, constrain the elements of $\vec{m}^{(q+1)}_{it}$ to the
               interval $[1, n-1]$
12             SORT $\vec{m}^{(q+1)}_{it}$, sort $\vec{m}^{(q+1)}_{it}$ such that $m_{(it)_1} \le m_{(it)_2} \le \ldots \le m_{(it)_{t-1}}$
13         if $\left[ \sum_{s=1}^{t-1} \sum_{i=1}^{N_p} |\vec{v}^{(q+1)}_{is}| < \epsilon_t = \epsilon N_p(t-1) \textbf{ OR } q = q_{max} \right]$ then STOP
14     if $E(\hat{\hat{\mathbf{U}}}_t) > E(\mathbf{U}_{\hat{\hat{G}}})$ then $\hat{\hat{G}} = \hat{\hat{m}}_t$
```

Although the CLODD algorithm looks complex, it is really quite simple. Line 2 initializes the particles according to the following procedure:

1. Randomly choose $\vec{m}^{(1)}_{it}$ so that

$$\vec{m}^{(1)}_{it} \neq \vec{m}^{(1)}_{st}, i \neq s,$$

and

$$\vec{m}_{it}^{(1)} \leftarrow \mathbf{U}_{it}^{(1)} \in \mathbf{M}_{htn}^{*}.$$

2. $\vec{v}_{it}^{(1)} = \mathbf{rand}([-1, 1]) : \hat{m}_{it} = \vec{m}_{it}^{(1)} : \hat{\hat{m}}_t = \vec{m}_{1t}^{(1)}$

Line 6 builds the candidate partitions according to the particles, including the particles' current location, previous best personal location, and previous best overall location. Although, in our algorithm outline we show that candidate partitions are built at every iteration of the particle swarm, because this problem is discrete in nature, candidate partitions *only* need to built when new particle locations are explored. If a candidate partition has been tested in a previous iteration, the objective function does not need to be calculated again. Lines 7 and 8 test to see whether candidate partitions are better than the best previously found candidate partitions. Line 9 is the PSO update equation, which updates the velocity of each particle. Line 10 calculates the new location of each particle. Lines 11 and 12 are of particular interest and lead to the following remark. Line 12 sorts the elements of $\vec{m}^{(q+1)}$ such that the elements are ordered and increasing. Line 13 is the termination criterion for the PSO. Finally, line 14 keeps track of the best candidate partition over all values of t, the number of clusters.

Remark 2. It is possible that at the end of the Round operation, $\vec{m}^{(q+1)}$ could have one or more negative entries. This would be not be a valid partition. For example, we might have $\vec{m}^{(q+1)} = (-2, -1, 0, 3, 1)$ *before* clipping. This condition is only temporary, because $\vec{m}^{(q+1)}$ is clipped before it has a chance to reach the objective function. Thus, CLIP $(-2, -1, 0, 3, 1) = (1, 1, 1, 3, 1)$. In this example, there are several equal elements in the clipped $\vec{m}^{(q+1)}$. This is *NOT* a valid partition, because it violates the condition that $m_1 < m_2 < \cdots < m_{t-1}$. When this occurs, CLODD will not evaluate the objective function and, subsequently, will not update the local or best particle positions. The particle is allowed to stay in its location (which is invalid) but does not contribute. If the particle is lucky, it will be updated to a *valid* location at the next iteration.

Remark 3. If the termination criterion $\sum_{s=1}^{t-1} \sum_{i=1}^{N_p} |\vec{v}_{is}^{(q+1)}| < \epsilon_t = \epsilon N_p(t-1)$ is met, the average value of the magnitude of the particle velocities is less than ϵ. There are $(t-1)$ velocity elements in each particle. The particles can only move in discrete jumps (integers; see line 10); hence, an average velocity less than $\epsilon = 0.5$ virtually ensures that all particles have converged to a solution—usually, but not necessarily, the globally best solution of Equation 11.

Remark 4. Two or more particles can occupy the same location. In fact, as a swarm approaches termination by the velocity criterion, many particles may be located at the global maximum. As a results of the formulation of the update equation (line 9), once the particles arrive at the global maximum (with minimal momentum), they stay.

The specification we have given for CLODD looks pretty intimidating, but this algorithm is simple to describe verbally. For each c

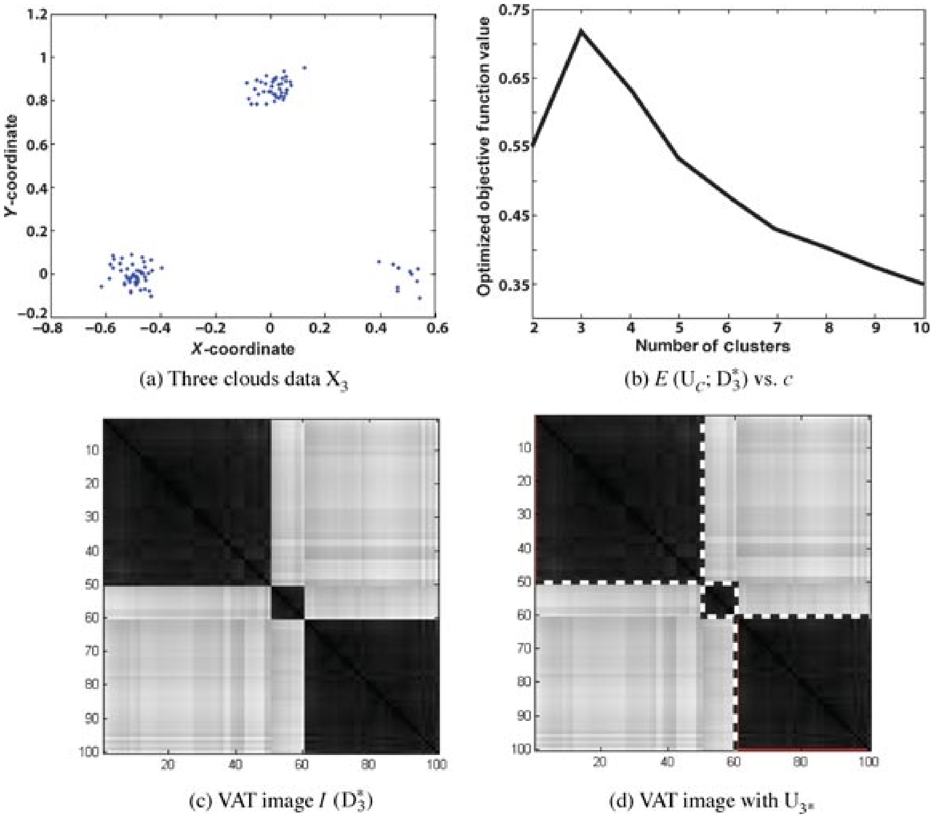
1. Guess a bunch of particles, each of which represents a candidate aligned c-partition of n objects;
2. Test the fit of each guess to the image $I(\mathbf{D}^*)$ using $E(\mathbf{U};\mathbf{D}^*)$;
3. Adjust each particle by moving the interior boundaries according to the standard PSO delta rule;
4. GOTO 2. until termination condition is satisfied.

6. NUMERICAL EXAMPLES

This section contains a number of examples that illustrate various facets of the CLODD algorithm. First, we list the computing protocols (for all examples except where noted). $\mathbf{C} = \{2, 3, \ldots, c_{\max}\}$ varies from example to example; $\alpha = 0.5$, $\gamma = 0.05$; $N_p = 20$ particles per swarm; $q_{\max} = 1000$; $\epsilon = 0.5 =$ termination threshold multiplier; $K = 0.75$; $A_{\text{local}} = A_{\text{global}} = 2$. Many papers attempt to establish "best" choices for the PSO parameters. We chose the values shown after a limited amount of experimentation with each. A given problem may warrant other choices, but here, we concentrate on the showing the basic points of CLODD.

Example 2. (*Three Gaussian Clouds*). Figure 5a shows $n = 100$ object vectors $\mathbf{X}_3 \subset \mathbb{R}^2$. Figure 5c is the VAT image $I(\mathbf{D}_3^*)$ of the corresponding Euclidean dissimilarity data $\mathbf{D}_3$. The well-defined cluster structure that is visually evident in $\mathbf{X}_3$ is represented exactly in $I(\mathbf{D}_3^*)$, so we expect CLODD to find a perfect match to the boundaries in the VAT image. Figure 5b is a plot of the values of the objective function $E(\mathbf{U}_c;\mathbf{D}_3^*)$ for the PSO winners at each $c = 2, 3, \ldots, 10$. The aligned partition $\mathbf{U}_{3*}$ has a strong maximum of 0.72 in Figure 5b. This partition—the expected perfect match—is superimposed on $I(\mathbf{D}_3^*)$ in Figure 5d.

Example 3. (*Three Lines*). Figure 6a shows $n = 100$ object vectors $\mathbf{X}_{3L} \subset \mathbb{R}^2$. Figure 6c is the VAT image $I(\mathbf{D}_{3L}^*)$ of the corresponding Euclidean dissimilarity data $\mathbf{D}_{3L}$. Most observers would agree that there is a well-defined cluster structure, which is visually evident in $\mathbf{X}_{3L}$, but view 6c shows that VAT does not elicit this from these data. The visual impression given by $I(\mathbf{D}_{3L}^*)$ is that $\mathbf{X}_{3L}$ has $c = 5$ clusters, and we see that CLODD agrees. The PSO winners at each c, shown in Figure 6b, have a clear maximum at $c = 5$. Note that the corresponding aligned partition $\mathbf{U}_{5*}$, which solves Equation 12, has a very weak maximum of 0.23. This partition of $\mathbf{X}_{3L}$ is shown in Figure 6d. What went wrong? VAT failed to reorder the distance matrix to show the $c = 3$ linear clusters. As discussed in Ref. 35, the ability of VAT to show "proper" cluster tendency is directly linked to Dunn's cluster validity index.[39] Dunn's index for the visually apparent 3-partition of $\mathbf{X}_{3L}$ is approximately 0.3, which is less than 1; hence, the contrast of the VAT image is not sufficient to show a cluster tendency of $c = 3$.



(a) Three clouds data X_3

(b) $E\ (U_C; D_3^*)$ vs. c

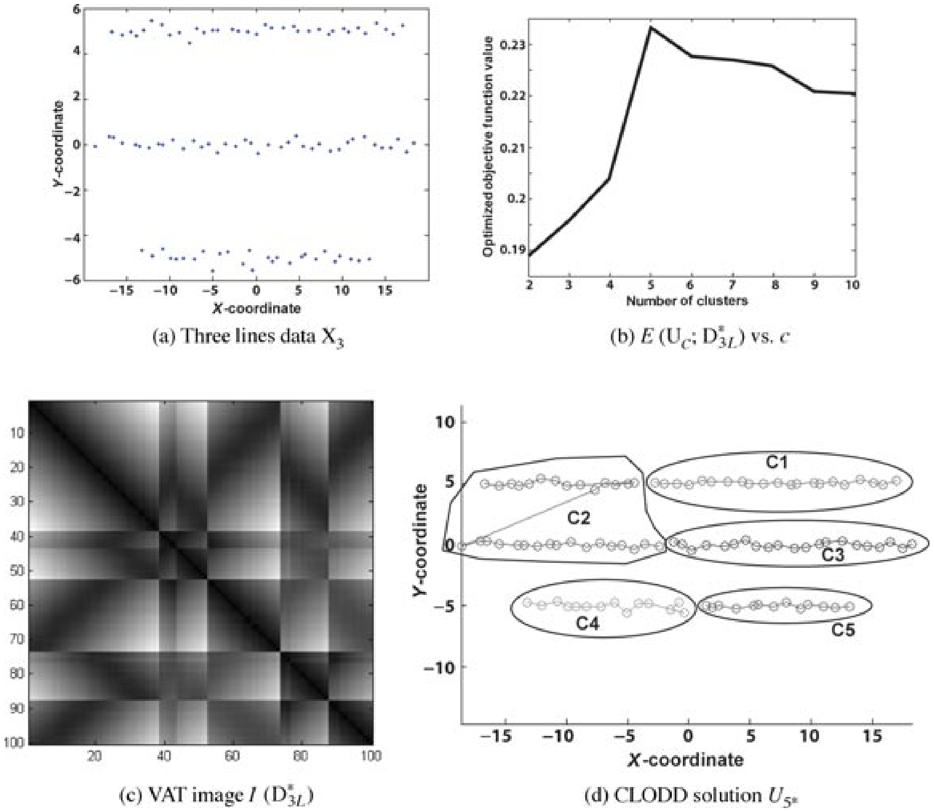
(c) VAT image $I\ (D_3^*)$

(d) VAT image with U_{3*}

Figure 5. Object data scatterplot, PSO winners, VAT image, and optimal CLODD partition for the Three Clouds data $\mathbf{X}_3$—dotted line in view (d) indicates partition boundaries.

Example 4. (*Uniform Random Field*). To study the candidate partitions that CLODD might suggest when there are *no* visible clusters in the data, we generated a set of 500 object vectors $\mathbf{X}_u$, uniformly distributed in [0, 1] × [0, 1], and converted them to Euclidean dissimilarity data $\mathbf{D}_u$. What would you conjecture, based only on the visual evidence in the VAT image $I(\mathbf{D}_u^*)$, shown in Figure 7c? There are several dark blocks in the lower part of this image that attract the eye, and there are quite a few smaller dark blocks along the diagonal, so you might speculate that there is some type of cluster substructure in the data—albeit weak and perhaps not distinguishable by the reordering procedure used by VAT.

The solution of the CLODD objective function Equation 12 for these data is indicated by the maximum on the graph in Figure 7b. CLODD finds $c = 5$ clusters, and the corresponding partition is shown in Figure 7d. The optimal CLODD partition $\mathbf{U}_{5*}$ is not an unreasonable fit to the VAT image. Although it certainly could be argued that there is *NO* cluster structure in these data. Hence, does CLODD fail for these data? No. CLODD finds an aligned partition that is a pretty good match to the VAT image it has to work with. The failure in this case, as in the three lines example, is



(a) Three lines data X_3

(b) $E\ (U_C;\ D^*_{3L})$ vs. c

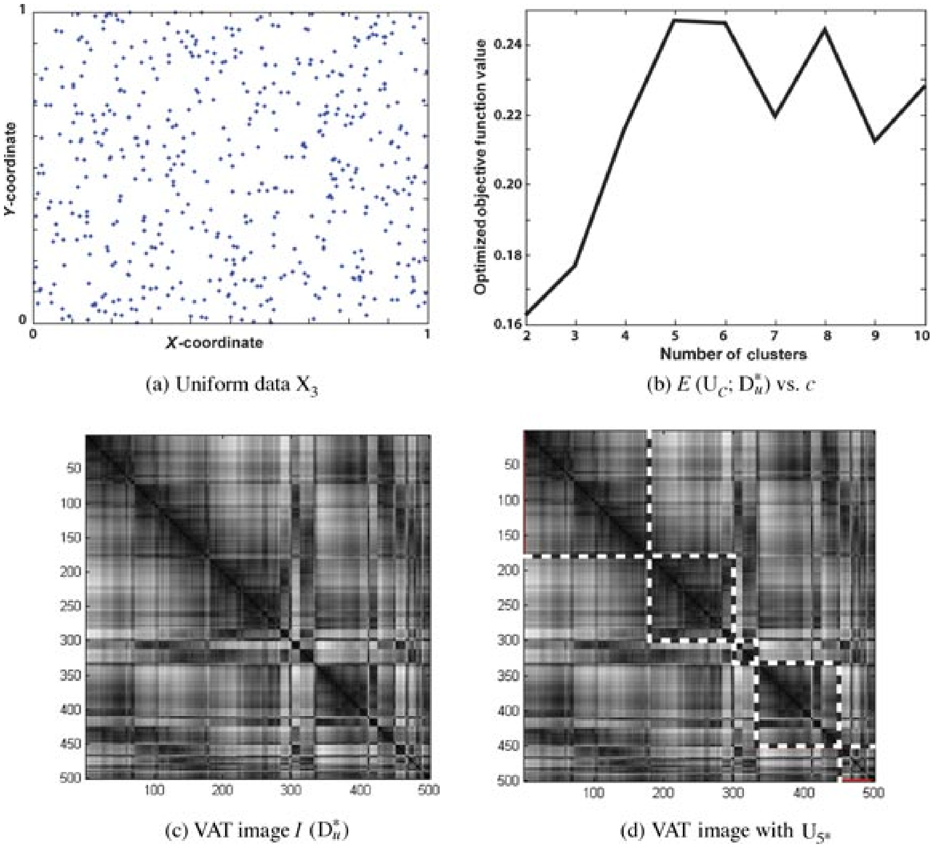
(c) VAT image $I\ (D^*_{3L})$

(d) CLODD solution U_{5*}

Figure 6. VAT image, PSO winners, and optimal CLODD partition for the 3 lines data $\mathbf{X}_{3L}$.

due to VAT, which produces a reordered image that seems to have more structure than the scatterplot of these data suggest. This reminds us that the job of every clustering algorithm is to find clusters, and CLODD is not different from all other clustering algorithms in this respect: CLODD does its job—namely, finding clusters where none seem to exist.

Example 5. (*"VOTE" Data*). This example uses the real-world VOTE data set, downloaded from the UCI Machine Learning Repository.[40] The data are generated from Congressional voting records and consist of the 1984 records of the 435 members of the United States House of Representatives on 16 key votes. These data consist of "y" for yea, "n" for nay, and "?" for unknown disposition. To represent these data numerically, we chose the values 0.5 for yea, −0.5 for nay, and 0 for unknown. Thus, the voting records are represented by an object data set $\mathbf{X}_{VOTE} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{435}\} \subset \mathbb{R}^{16}$. We use Euclidean and squared Euclidean distances to generate relational data sets $\mathbf{D}_e$ and $\mathbf{D}_{e^2}$ from $\mathbf{X}_{VOTE}$. Figure 8a shows the VAT



(a) Uniform data X_3

(b) $E(U_c; D_u^*)$ vs. c

(c) VAT image $I(D_u^*)$

(d) VAT image with U_{5*}

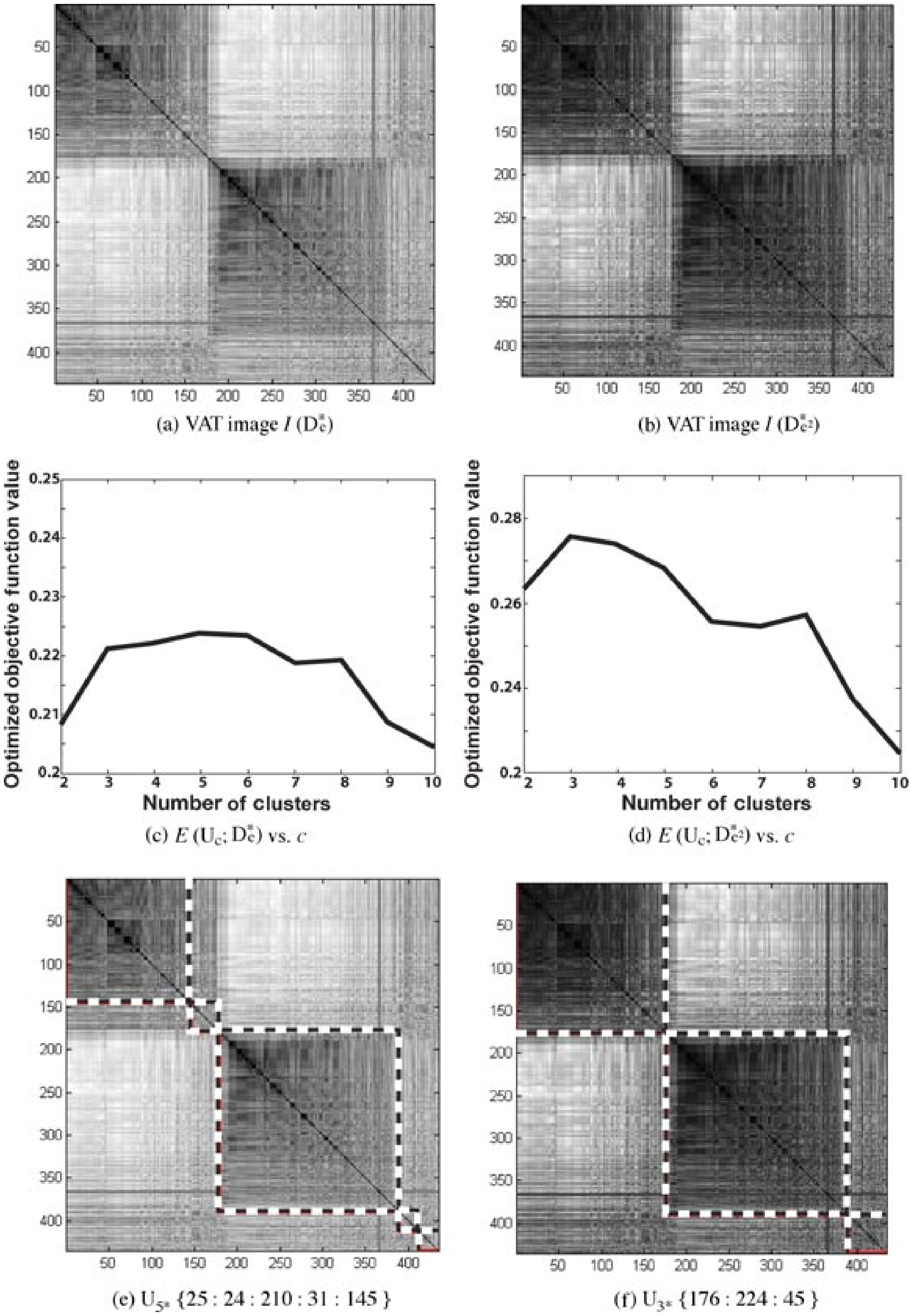
Figure 7. VAT image, PSO winners, and optimal CLODD partition for the uniform data $\mathbf{X}_u$—dotted line in view (d) indicates partition boundaries.

image $I(\mathbf{D}_e^*)$. This image gives the impression that there are two clusters in the data, but the intensities at the edges of the dark regions fade into neighboring pixels more or less continuously, and the lower corner, along the diagonal of the lower block, simply disappears. Figure 8c plots the values of the objective function for the winner of each PSO competition, where, recall, each PSO competition is for a different number of clusters. The range of values of the vertical axis of Figure 8c is very compressed and is relatively small—$E(\mathbf{U}; \mathbf{D}_e^*)$ is valued in [0.208, 0.223]. The graph from $c = 3$ to $c = 6$ is nearly flat, so while there is a maximum at $c = 5$, it is relatively weak. This indicates that the optimal CLODD partition $\mathbf{U}_{5*}$ is not *clearly* preferable, just better than those at other values of c.

Figure 8b, the VAT image $I(\mathbf{D}_{e2}^*)$, has improved visual contrast. The dark blocks are darker and the boundaries seem more distinct, but we still see a gray area along the bottom and right edge of the VAT image. Figure 8(d) plots the winning objective function value at each c. Figures 8c and 8d show that changing the input data from $\mathbf{D}_e$ to $\mathbf{D}_{e^2}$ changes the number of optimal clusters from $c = 5$ to $c = 3$. This demonstrates the ability of the edginess and contrast factors, which comprise $E(\mathbf{U}; \mathbf{D}^*)$, to track changes in contrast and edge definition in the VAT image $I(\mathbf{D}^*)$. The 3-partition chosen as the best match for $I(\mathbf{D}^*_{e^2})$ is $\mathbf{U}_{3*} = \{176 : 224 : 45\}$. This is a somewhat more satisfying result than the partition $\mathbf{U}_{5*} = \{145 : 31 : 210 : 24 : 25\}$ that CLODD matches to $I(\mathbf{D}^*_e)$. The two identified classes in these data are Republicans (54.8%) and Democrats (45.2%), but this does not guarantee that the numerical data contain two geometrically well-defined clusters. Our conjecture is that the two apparent clusters correspond to Democrats and Republicans voting along party lines, while the poorly defined region in the bottom right corner of $I(\mathbf{D}^*_{e^2})$ corresponds to 45 voters who crossed party lines on these 16 votes.

Example 6. (*Bioinformatics Data*). Our last example uses one version of the real-world data $GPD194_{12.10.03}$, denoted here as $\mathbf{D}_{194}$. These data are different from the previous examples in that they are not derived from object data. Rather, they are derived directly from a (dis)similarity relation built with a fuzzy measure applied to annotations of 194 human gene products which appear in the Gene Ontology.[40] Popescu et al.[41] contains a detailed description of the construction of this data. These data comprise 21 gene products from the Myotubularin protein family, 87 gene products from the Receptor Precursor protein family, and 86 gene products from the Collagen Alpha Chain protein family. The three protein families are clearly visible in the image of $\mathbf{D}_{194}$ shown in Figure 9a; the upper left block is the Myotubularins, the middle block is the Receptor Precursors, and the lower right block is the Collagens. Note the strong substructure within the Collagen protein family dissimilarity data. This substructure has been corroborated in Ref. 43 and, as you will see, is also supported by CLODD.

Figure 9a displays an image of $\mathbf{D}_{194}$, and if you compare this image to the VAT image $I(\mathbf{D}^*_{194})$ in Figure 9c, you will see that they are similar, but not exactly equal. However, both these images seem to suggest that there are more than just three clusters, with $c = 5$–7 main clusters being our estimate from the VAT image. In this regard, CLODD agrees. Figure 9b shows a slight maximum in the objective function at $c = 6$, and the corresponding partition $\mathbf{U}_{6*}$ is shown superimposed in Figure 9d. In this example, the three highest values of the objective function, which occur at $c = 5$, 6, and 7, are all about 0.64. Compare this to the best values of the objective function in the previous examples. In the Three Clouds data, the maximum objective function value is larger than 0.6; in this example CLODD (arguably) found a good partition of these data. But in the Three Lines, Uniform, and VOTE data, where either VAT or CLODD performed less reliably, the value of the objective function is below 0.25. Hence, we believe that CLODD supports the substructure found in the collagen family. Also, please note that within the six main clusters found by CLODD in the $GPD194_{12.10.03}$ data (which, for lack of a better term, we call *first order* clusters), there are visually apparent subclusters (*second-order* clusters).



(a) VAT image I (D_c^*)

(b) VAT image I ($D_{c^2}^*$)

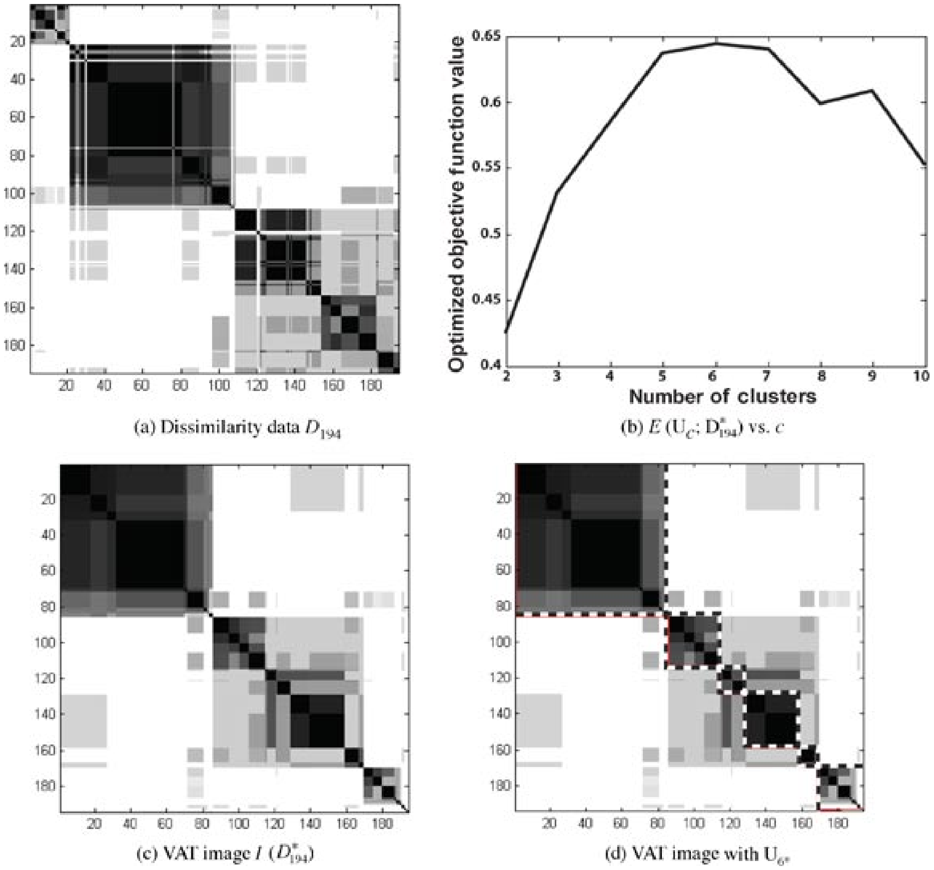
(c) E (U_c; D_c^*) vs. c

(d) E (U_c; $D_{c^2}^*$) vs. c

(e) U_{5*} {25 : 24 : 210 : 31 : 145 }

(f) U_{3*} {176 : 224 : 45 }

Figure 8. VAT images, PSO winners, and optimal CLODD partitions for the VOTE data—dotted line in views (e,f) indicates partition boundaries. Views (a,c,e) use Euclidean dissimilarity relation, and views (b,d,f) use squared Euclidean dissimilarity relation.



(a) Dissimilarity data D_{194}

(b) E (U_C; D^*_{194}) vs. c

(c) VAT image I (D^*_{194})

(d) VAT image with U_{6^*}

Figure 9. VAT image, PSO winners and optimal CLODD partition for the $GPD194_{12.10.03}$ data—dotted line in view (d) indicates partition boundaries

7. CONCLUSIONS AND FUTURE RESEARCH

Our examples demonstrate that when **D** has "good" clusters, CLODD will find them. In our examples when CLODD finds a good match to a good VAT image of the data, the value of the objective function is larger than 0.6. But in the examples where either VAT or CLODD is less reliable, the value of the objective function is below 0.25. This indicates that CLODD is useful for both finding clusters in unlabeled data and, also, presenting a cluster validity index of those clusters.

There are algorithms besides VAT that produce block diagonal images: some are displays of clusters already found[7,19,20,25,26] others are constructed, like VAT, to assess structure prior to clustering[24,26]; still others are used to simultaneously find and display clusters[7,18,21]; and, finally, images with this type of structure are used to attack the validity question.[30,31] Consequently, CLODD is much more widely useful than it might appear. However, many good questions remain. For example, we have ignored the possibility that Equation 12 may not have a solution, or that it has more than one. These questions are interesting, but the objective function in Equation 11 is discontinuous on its domain; hence, these questions are indeed formidable.

On a more practical note, we ask whether there is a better way than trial and error to find a reliable pair of CLODD parameters (α, γ)? Our initial attempts at approaching this question have centered on computational ways to make CLODD "adaptive," but so far, we have met with little success. Another interesting question concerns the reliance of CLODD on VAT. Certainly, CLODD will fail when VAT does, and we have illustrated here that this can happen. It is possible that other reordering methods might be useful "front-end" partners for CLODD in such cases. This leads to a related question concerning the size of the data **O**. VAT is a useful reordering scheme for small- to medium-sized data sets ($n \leq 10{,}000$). The scalable version of VAT[32] produces a sample-based estimate of the VAT image $I(\mathbf{D}^*)$ for very large n, but does not reorder the very large data in preparation for CLODD clustering. What is the bottom line? As with all research, we are left interesting, unanswered questions.

References

1. Duda R, Hart P, Stork D. Pattern classification. 2nd ed., New York: Wiley-Interscience; 2000.
2. Theodoridis S, Koutroumbas K. Pattern recognition, 3rd ed.; San Diego, CA, Academic Press; 2006.
3. Bezdek J. Pattern recognition with fuzzy objective function algorithms. New York: Plenum; 1981.
4. Bezdek J, Keller J, Krishnapuram R, Pal N. Fuzzy models and algorithms for pattern recognition and image processing. Norwell, MA: Kluwer; 1999.
5. Hartigan J. Clustering algorithms. New York: Wiley; 1975.
6. Jain A, Dubes R. Algorithms for clustering data. Englewood Cliffs, NJ: Prentice-Hall; 1988.
7. Johnson D, Wichern D. Applied multivariate statistical analysis. 6 ed. Englewood Cliffs, NJ: Prentice Hall; 2007.
8. Bezdek J, Hathaway R. VAT: A tool for visual assessment of (cluster) tendency. In: Proc. IJCNN 2002, Piscataway, NJ, 2002. pp 2225–2230.
9. Prim R. Shortest connection networks and some generalisations. Bell System Tech J 1957;36:1389–1401.
10. Clerc M, Kennedy J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. IEEE Trans Evolut Comput 2002;6(1):58–73.
11. Andrews D. Plots of high dimensional data. Biometrics 1972;28:125–136.
12. Chernoff H. The use of faces to represent points in k-dimensional space. J Am Stat Assoc 1973;68:361–368.
13. Kleiner B, Hartigan J. Representing points in many dimensions by trees and castles. J Am Stat Asooc 1981;76:260–269.
14. Tryon R. Cluster analysis. Ann Arbor, MI: Edwards Bros.; 1939.
15. Tukey J. Exploratory data analysis. Reading, MA: Addison-Wesley; 1977.
16. Everitt B. Graphical techniques for multivariate data. New York: Elsevier; 1978.
17. Cleveland W. Visualizing data. Summit, NJ: Hobart Press; 1993.
18. Sneath P. A computer approach to numerical taxonomy. J Gen Microbiol 1957;17:201–226.
19. Floodgate G, Hayes P. The Adansonian taxonomy of some yellow pigmented marine bacteria. J Gen Microbiol 1963;30:237–244.
20. Ling R. A computer generated aid for cluster analysis. CACM 1973;16(6):355–361.

21. Tran-Luu T. Mathematical concepts and novel heuristic methods for data clustering and visualization. PhD Thesis, University of Maryland, College Park, MD; 1996.
22. van Someren E, Wessels L, Reinders M. GENLAB toolbox. Available at http://genlab.tudelft.nl 2000.
23. Girolami M. Mercer kernel-based clustering in feature space. IEEE Trans Neural Networks 2002;13:780–784.
24. Zhang D, Chen S. Clustering incomplete data using kernel-based fuzzy *c*-means algorithm. Neural Process Lett 2003;18:155–162.
25. Baumgartner R, Somorjai R, Summers R, Richter W. Ranking fMRI time courses by minimum spanning trees: Assessing coactivation in fMRI. NeuroImage 2001;13:734–742.
26. Baumgartner R, Somorjai R, Summers R, Richter W, Ryner L. Correlator beware: correlation has limited selectivity for fMRI data analysis. NeuroImage 2000;12:240–243.
27. Strehl A, Ghosh J. A scalable approach to balanced, high-dimensional clustering of market-baskets. In: Proc HiPC. LNCS vol 1970, New York: Springer; 2000. pp 525–536.
28. Strehl A, Ghosh J. Value-based customer grouping from large retail data-sets. In: Proc SPIE Conf on Data Mining and Knowledge Discovery, vol. 4057, Bellingham, WA: SPIE Press; 2000. pp 33–42.
29. Dhillon I, Modha D, Spangler W. Visualizing class structure of multidimensional data. In Proc 30th Symp on the Interface, Computing Science, and Statistics, Weisberg S, editor, 1998.
30. Hathaway R, Bezdek J. Visual cluster validity for prototype generator clustering models. Pattern Recog Lett 2003;24:1563–1569.
31. Huband J, Bezdek J. VCV2–Visual cluster validity. In: Computational intelligence: research frontiers. Springer: Berlin; 2008. pp 293–308.
32. Hathaway R, Bezdek J, Huband J. Scalable visual asseessment of cluster tendency for large data sets. Pattern Recog 2006;39(7):1315–1324.
33. Bezdek J, Hathaway R, Huband J. Visual assessment of clustering tendency for rectangular dissimilarity matrices. IEEE Trans Fuzzy Syst 2007;15(5):890–903.
34. Havens T, Bezdek J, Keller J, Popescu M, Huband J. Is VAT really single linkage in disguise? Ann Math Artif Intell. In review.
35. Havens T, Bezdek J, Keller J, Popescu M. Dunns cluster validity index as a contrast measure of VAT image. In: Proc ICPR, Tampa, FL, 2008.
36. Sledge I, Huband J, Bezdek J. (Automatic) Cluster count extraction from unlabeled datasets. In: Proc ICNC/FSKD, Jinan Shandong, China, 2008. pp 3–13.
37. Wang L, Leckie C, Rao K, Bezdek J. Automatically determining the number of clusters from unlabeled data sets. IEEE Trans Knowl Eng 2009;21(3):335–350.
38. Epp S. Discrete mathematics with applications. Boston, MA: Brooks/Cole Publishing; 2004.
39. Dunn J. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J Cybernet 1974;3(3):32–57.
40. Asuncion A, Newman D. UCI machine learning repository. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html 2007.
41. The Gene Ontology Consortium. The gene ontology (GO) database and informatics resource. Nucleic Acids Res 2004;32:D258–D261.
42. Popescu M, Keller J, Mitchell J, Bezdek J. Functional summarization of gene product clusters using Gene Ontology similarity measures. In: Proc 2004 ISSNIP. Piscataway, NJ: IEEE Press, 2004. pp 553–559.
43. Myllyharju J, Kivirikko K. Collagens, modifying enzymes, and their mutation in humans, flies, and worms. Trends Genet 2004;20(1):33–43.