A New Formulation of the coVAT Algorithm for Visual Assessment of Clustering Tendency in Rectangular Data

Timothy C. Havens,[1,*] James C. Bezdek[2,†]
[1]*Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824*
[2]*Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Victoria 3010, Australia*

Since 1998, a graphical representation used in visual clustering called the *reordered dissimilarity image* or cluster heat map has appeared in more than 4000 biological or biomedical publications. These images are typically used to visually estimate the number of clusters in a data set, which is the most important input to most clustering algorithms, including the popularly chosen fuzzy c-means and crisp k-means. This paper presents a new formulation of a matrix reordering algorithm, coVAT, which is the only known method for providing visual clustering information on all four types of cluster structure in rectangular relational data. Finite rectangular relational data are an $m \times n$ array R of relational values between m row objects O_r and n column objects O_c. R presents four clustering problems: clusters in O_r, O_c, $O_{r\cup c}$, and coclusters containing some objects from each of O_r and O_c. coVAT1 is a clustering tendency algorithm that provides visual estimates of the number of clusters to seek in each of these problems by displaying reordered dissimilarity images. We provide several examples where coVAT1 fails to do its job. These examples justify the introduction of coVAT2, a modification of coVAT1 based on a different reordering scheme. We offer several examples to illustrate that coVAT2 may detect coclusters in R when coVAT1 does not. Furthermore, coVAT2 is not limited to just relational data R. The R matrix can also take the form of feature data, such as gene microarray data where each data element is a real number: Positive values indicate upregulation, and negative values indicate downregulation. We show examples of coVAT2 on microarray data that indicate coVAT2 shows cluster tendency in these data.

1. INTRODUCTION

Consider a set of objects $O = \{o_1, \ldots, o_n\}$. These objects can represent virtually anything—vintage bass guitars, pure-bred cats, cancer genes expressed in a microarray experiment, cake recipes, maduro belicoso cigars, or Web pages. The

*Author to whom all correspondence should be addressed: e-mail: havenst@gmail.com.
†e-mail: jcbezdek@gmail.com

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 27, 590–612 (2012)

View this article online at wileyonlinelibrary.com. • DOI 10.1002/int.21539

object set O is *unlabeled data*; that is, each object has no associated class label. However, it is assumed that there are subsets of similar objects in O. These subsets are called *clusters*.

Clustering is the process of grouping the objects in O in a sensible manner. This process is often performed to elucidate the similarity and dissimilarity among and between the grouped objects. Clustering has also been called unsupervised learning, typology, and partitioning,[1] Although clustering is typically thought of as only the act of separating objects into the proper groups, cluster analysis actually consists of three concise questions: (i) *cluster tendency*—how many clusters are there?, (ii) *partitioning*—which objects belong to which cluster and to what degree?, and (iii) *cluster validity*—are the partitions "good"?. In this paper, we specifically address tendency.

Numerical object data are represented as $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^p$, where each dimension of the vector $\mathbf{x}_i$ is a feature value of the associated object o_i. These features can be composed of almost any discrete or continuously valued numbers, e.g., Red-Green-Blue (RGB) values, gene expression, year of manufacture, and number of stripes. Another way to represent the objects in O is with numerical *relational* data, which consist of n^2 values that represent the (dis)similarity between pairs of objects. These data are commonly represented by a relational (or proximity) matrix $P = [P_{ij} = \text{relation}(o_i, o_j) | 1 \leq i, j \leq n]$. The relational matrix P often takes the form of a *dissimilarity* matrix D and, here, we denote $d(o_i, o_j) = \text{relation}(o_i, o_j)$. Dissimilarity can be interpreted as a distance between objects. For instance, numerical data X can always be converted to D by $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ (any vector norm on $\mathbb{R}^p$). There are, however, similarity and dissimilarity relational data sets that do not begin as numerical object data; for these, there is no choice but to use a relational algorithm. Hence, relational data represent the "most general" form of input data.

An even more general form of relational data is *rectangular*. These data are represented by an $m \times n$ dissimilarity matrix R, where the entries are the pairwise dissimilarity values between m row objects O_r and n column objects O_c. An example comes from Web-document analysis, where the row objects are m Web pages, the columns are n words, and the (dis)similarity entries are occurrence measures of words in Web pages.[2] In this case, the row and column objects are nonintersecting sets, such that the pairwise relation among row (or column) objects is unknown. Conventional relational clustering algorithms are ill-equipped to deal with rectangular data. In addition, the definition of a cluster as a group of similar objects takes on a new meaning. There can be groups of similar objects that are composed of only row objects, of only column objects, or of mixed objects (often called *coclusters*). In this paper, we consider all these types of clusters in rectangular data.

Consider the hypothetical example shown in Table I. The row objects O_r are magazines: *Time*, *National Geographic*, *Newsweek*, and *Smithsonian*. The column objects O_c are subjects: Guns, Celebrities, War, Lakes, Seas, Bombs, Mountains, Singers, and Dancers. The value in each entry indicates the relationship between the magazine and the subject: a value of 1 indicates that the magazine is very likely to cover the subject and a value of 0 indicates otherwise. These data represent the similarity between magazines and subjects. Because coVAT works on dissimilarity values, we converted these similarities to dissimilarities by

Table I. Hypothetical example of the four clustering problems in rectangular data.

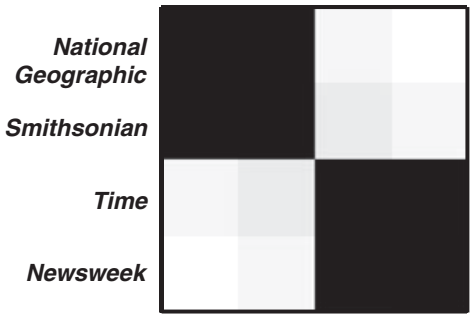
	Similarity data								
	Subjects								
Magazines	Guns	Celebrities	War	Lakes	Seas	Bombs	Mountains	Singers	Dancers
Time	1	0.5	1	0	0	1	0	0.5	0.4
National Geographic	0.2	0	0.3	1	1	0.2	1	0.1	0.1
Newsweek	1	0.5	1	0.1	0.1	1	0.1	0.3	0.5
Smithsonian	0	0	0.2	1	1	0.1	1	0.2	0.2
	coVAT1 reordered data								
	Subjects								
Magazines	Lakes	Seas	Mountains	Dancers	Singers	Celebrities	Guns	War	Bombs
National Geographic	1	1	1	0.1	0.1	0	0.2	0.3	0.2
Smithsonian	1	1	1	0.2	0.2	0	0	0.2	0.1
Time	0	0	0	0.4	0.5	0.5	1	1	1
Newsweek	0.1	0.1	0.1	0.4	0.3	0.5	1	1	1

similarity = 1-*dissimilarity*.[a] We reordered the dissimilarity data with coVAT1 and then applied this reordering to the original similarity data. Table I contains the results.

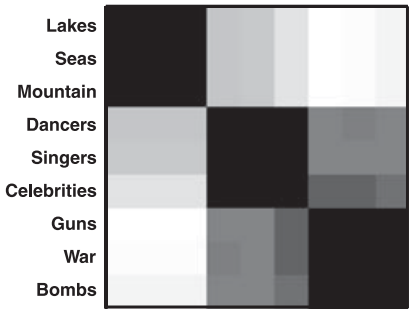
Table I shows the raw similarity data and the coVAT1-reordered data. The groups of similar values in these data are made evident by the coVAT1 reordering. There are two magazine groups, {travel} and {news} and three subject groups, {scenery words}, {people words}, and {military words}. These are the first two types of clusters in rectangular data: clusters in O_r and O_c. Figures 1a and 1b show the coVAT1 images that indicate the number of row (magazine) and column (subject) clusters—the number of clusters is indicated by the number of dark blocks on the diagonal of the image. The third cluster type is groups in the union of the objects $O_{r\cup c} = O_r \cup O_c$. Figure 1c shows the coVAT1 image of the dissimilarity matrix of the union of the magazines and subjects. The three groups are {scenery words and travel magazines}, {people words}, and {military words and news magazines}. The final type of cluster is what we call coclusters, or clusters that are composed of both row and column objects. Figure 1d shows the coVAT1 image that indicates the presence of coclusters by dark rectangular blocks. This image clearly shows two coclusters: {scenery words and travel magazines} and {military words and news magazines}.

Assessment of cluster tendency is important to all clustering algorithms that require, as input, a choice of the number of clusters, which includes the well-known k-means[3] and fuzzy c-means.[4] The coVAT algorithms allow one to assess the number

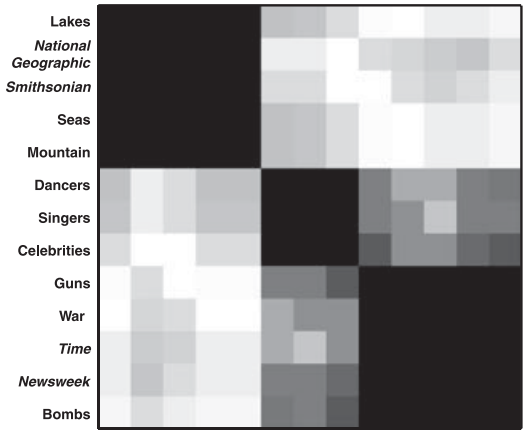
[a] The coVAT and VAT algorithms could easily be adapted to work directly on similarity values by changing the $arg\,max$ operator to $arg\,min$ and vice versa.



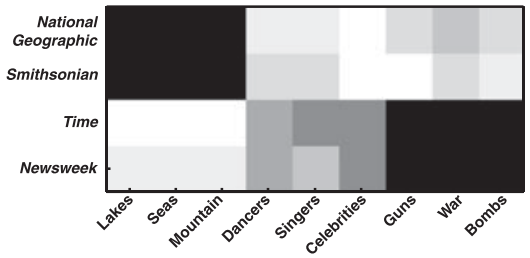
(a) coVAT1 image of magazine dissimilarity



(b) coVAT1 image of subject dissimilarity



(c) coVAT1 image of magazine union subject dissimilarity



(d) coVAT1 image of magazine-subject dissimilarity

Figure 1. coVAT1 images of magazines-subjects relational data.

of the different types of clusters in rectangular data R. This assessment method can therefore be used as a visual preprocessing step for fuzzy coclustering algorithms, such as those proposed in Refs. 5–8.

Section 3 presents a new method for finding the reordering of the rectangular dissimilarity matrix R. Section 4 presents numerical examples of the new formulation, coVAT2, and comparisons with coVAT1. Section 5 concludes this paper. We now discuss related research on visual clustering and introduce the VAT algorithms.

2. BACKGROUND

Visual clustering is the search for structure in graphical representations of numerical data. There are three important functions connected with this enterprise: display functions, reordering functions, and clustering functions. Visual methods have been used for all three of the canonical problems: tendency assessment, clustering, and validation. There are two branches of visual clustering algorithms, depending on the input data: those that take feature vectors as input and those that take relational data as input. The focus of this paper is primarily relational data. Good references on visual clustering methods for feature vectors include Refs. 9–14.

Visual clustering of relational data has a long and rich history, much of which is described in Ref. 15. The earliest publications we know of on this topic include the 1873 work by Loua,[16] which presented a hand-shaded dissimilarity matrix image describing maps in Paris, the 1909 work by Czekanowski,[17] apparently the first method for clustering in dissimilarity data using a visual, approach, and the 1939 work by Tryon,[18] who clustered a 20×20 correlation matrix by visually aggregating plots of matrix rows that Tryon called "correlation profiles." Cattell[19] pioneered the idea of image-based visual clustering by representing dissimilarity data as hand-shaded pixels with three possible "intensities." Other notable early contributions include those by Guttman,[20] Mayr,[21] and Torgerson[22] Sneath[23] advanced the field by building the dissimilarity matrix with a computer, but the image of the matrix was still constructed by hand. All these methods share the commonality that clustering or reordering was subsequently done manually.

In the early 1960s, Floodgate and Hayes[24] proposed a computational method for reordering dissimilarity data; the image of the reordered data was still hand-rendered. Ling,[25] in 1973, proposed the first completely automated computational method for reordering dissimilarity data and producing an image. Ling's algorithm, called SHADE, used complete-linkage hierarchical clustering to reorder the data, and the image was created by overstriking standard printed characters to produce 15 halftone intensities. Later methods for reordering and visualizing dissimilarity data include the "graphical method of shading" proposed by Johnson and Wichern[26] and Tran-Luu's[27] idea of searching for the most "acceptable" block-diagonal matrix by optimizing an objective function that measures "blockiness."

Other related reordering methods include graph-based schemes, such as depth-first search,[28] and schemes specific to unweighted connected graphs, such as Reverse Cuthill-Mckee,[29] King's algorithm,[30] and Sloan's algorithm.[31] A recent innovation for weighted connected graphs is spectral ordering,[28] which reorders the connection

matrix according to an Eigen-based decomposition of the graph's Laplacian matrix. This method is very effective and stable, but suffers from a very high computational complexity for fully connected graphs, $O(n^6)$.

For visualization, by far the most popular graphical representation has been the *reordered dissimilarity image* (RDI), which compacts large amounts of information into a small space to bring out coherent patterns in the data. Since 1998, RDIs have appeared in *well over 4000 biological or biomedical publications.*[32]

2.1. VAT Family

Unlike the coVAT algorithms, the methods described in Section 2. only work with square relational data or feature vectors. But in coclustering, the identification of submatrices of low dissimilarity values in rectangular data R is an important problem. This process is equivalent to finding subsets of similar objects in $O = O_r \cup O_c$ that contain both O_r and O_c. Representative works on the process of finding the coclusters include Refs. 2,5,33–37. The coVAT algorithms can be used with any of these coclustering algorithms to assist the user in defining the number and type of clusters in the rectangular data. The proposed coVAT2 algorithm represents a significant step forward in assessing clustering tendency in rectangular relational data.

The VAT family of algorithms that address the issue of cluster tendency include

1. **VAT**: The *Visual Assessment of clustering Tendency* (VAT) algorithm shows tendency for square dissimilarity data.[38] For large-scale data sets, *scalable* VAT can be used.[39] VAT has been shown to fail on some instances of "clearly structured data." The *improved* VAT (iVAT) algorithm[40] uses a path-based distance,[41] which is heavily used in spectral feature extraction and elsewhere. iVAT has been shown to successfully show cluster tendency in many of the instances where VAT fails.
2. **Efficient iVAT**: The original iVAT formulation had a complexity of $O(n^3)$. An $O(n^2)$ formulation was proposed in Ref. 42. Efficient iVAT was shown to be effective in real applications.[43,44]
3. **coVAT1**: This algorithm is useful for estimating the number of groups in coclustering problems.[45] For large-scale data, *scalable* coVAT1 can be used.[46] A version of coVAT1 that uses the iVAT path-based distance was proposed in Ref. 47.
4. **coVAT2**: This paper proposes a new formulation of coVAT1, which is shown to be effective in instances where coVAT1 fails. The examples in Figures 6–9 illustrate this fact.

We now describe in detail the VAT-family algorithms used in this paper. The notation used is shown in Table II.

2.1.1. VAT Algorithm

The VAT algorithm is based on Prim's algorithm[48] for finding the *minimum spanning tree* (MST) of a weighted connected graph.[38] VAT reorders the input data S according to the edge order of the MST, resulting in the reordered data matrix $\hat{S}$. Each pixel of the grayscale VAT image $I(\hat{S})$ displays the scaled dissimilarity value of two objects. For the sake of brevity and readability, from this point on we will denote

Table II. Notation used in this paper.

Symbol	Description
O	Set of objects
O_r	Set of "row" objects
O_c	Set of "column" objects (note: $O_r \cap O_c = \emptyset$)
S	Square dissimilarity data, $S_{ij} = \text{relation}(o_i, o_j)$
R	Rectangular dissimilarity data, $R_{ij} = \text{relation}((O_r)_i, (O_c)_j)$
$\hat{S}$	VAT-reordered S
$\hat{S}'$	iVAT-reordered S
R^*	coVAT1-reordered R
R^{**}	coVAT2-reordered R

the images $I(\cdot)$ simply by the matrix that is being imaged. 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 $\hat{S}$. The off-diagonal elements of $\hat{S}$ are scaled to the range [0, 1]. A dark block along the diagonal of $\hat{S}$ is a submatrix of "similarly small" dissimilarity values; hence, a dark block represents a cluster of objects that are relatively similar to each other. Thus, cluster tendency is shown by the number of dark blocks along the diagonal of the VAT image. VAT *suggests* cluster structure, but does not *find* it. Algorithm 1 lists the steps of the VAT algorithm.

Algorithm 1: VAT Reordering Algorithm[38]

Input: S — $n \times n$ dissimilarity matrix
Data: $K = \{1, 2, \ldots, n\}$; $I = J = \emptyset$; $P = (0, 0, \ldots, 0)$.
Select $(i, j) \in arg\,max_{p \in K, q \in K} S_{pq}$.
Set $P(1) = i$; $I = \{i\}$; and $J = K - \{i\}$.
for $r = 2, \ldots, n$ **do**
  Select $(i, j) \in arg\,min_{p \in I, q \in J} S_{pq}$.
  Set $P(r) = j$; Replace $I \leftarrow I \cup \{j\}$ and $J \leftarrow J - \{j\}$.

Obtain the ordered dissimilarity matrix $\hat{S}$ using the ordering array P as:
$\hat{S}_{pq} = S_{P(p),P(q)}$, for $1 \le p, q \le n$.
Output: Reordered dissimilarity $\hat{S}$

2.1.2. coVAT1 Algorithm

The coVAT1 algorithm begins by creating a square matrix $S_{r \cup c}$, part of which is composed of the input data, viz., a rectangular dissimilarity matrix R. $S_{r \cup c}$ is created by first estimating the dissimilarity matrices S_r and S_c, which are, respectively, square dissimilarity matrices that relate the objects in O_r and O_c to themselves — i.e. $[S_r]_{ij} \approx d(o_i, o_j)$ and $[S_c]_{ij} \approx d(o_{m+i}, o_{m+j})$. $S_{r \cup c}$ is organized as in Equation 1.

$$S_{r\cup c} = \begin{bmatrix} S_r & R \\ R^T & S_c \end{bmatrix} \approx$$

$$\begin{bmatrix} \begin{bmatrix} d(o_1,o_1) & \cdots & d(o_1,o_m) \\ \vdots & \ddots & \vdots \\ d(o_m,o_1) & \cdots & d(o_m,o_m) \end{bmatrix} & \begin{bmatrix} d(o_1,o_{m+1}) & \cdots & d(o_1,o_{m+n}) \\ \vdots & \ddots & \vdots \\ d(o_m,o_{m+1}) & \cdots & d(o_m,o_{m+n}) \end{bmatrix} \\ \begin{bmatrix} d(o_1,o_{m+1}) & \cdots & d(o_m,o_{m+1}) \\ \vdots & \ddots & \vdots \\ d(o_1,o_{m+n}) & \cdots & d(o_m,o_{m+n}) \end{bmatrix} & \begin{bmatrix} d(o_{m+1},o_{m+1}) & \cdots & d(o_{m+1},o_{m+n}) \\ \vdots & \ddots & \vdots \\ d(o_{m+n},o_{m+1}) & \cdots & d(o_{m+n},o_{m+n}) \end{bmatrix} \end{bmatrix} \quad (1)$$

The elements in S_r and S_c are estimated from R using any vector norms on $\mathbb{R}^n$ and $\mathbb{R}^m$,

$$[S_r]_{ij} = \lambda_r ||\mathbf{r}_{i*} - \mathbf{r}_{j*}||, \ 1 \le i, j \le m, \quad (2)$$

$$[S_c]_{ij} = \lambda_c ||\mathbf{r}_{*i} - \mathbf{r}_{*j}||, \ 1 \le i, j \le n, \quad (3)$$

where $\mathbf{r}_{i*}$ is the ith row of R, $\mathbf{r}_{*j}$ is the jth column of R, and λ_r and λ_c are scale factors such that the mean of the off-diagonal elements of S_r and S_c is equal to the mean of R. For the examples in this paper, we use the vector 2-norm to calculate S_r and S_c. Algorithm 2 lists the steps in the coVAT1 algorithm.[b]

3. coVAT2 ALGORITHM

coVAT1 reorders the rectangular matrix R by shuffling the VAT-reordering indices of $S_{r\cup c}$. Thus, coVAT1 is very dependent on the construction of $S_{r\cup c}$. We have discovered that coVAT1 fails to show cluster tendency in certain cases; see Figures 7 and 8 for examples. Algorithm 3 presents a reordering scheme that is not dependent on the reordering of $S_{r\cup c}$ — this matrix does not even need to be constructed. However, you still need the matrix $\hat{S}_{r\cup c}$ if you intend to assess cluster tendency in $O_{r\cup c}$. In the proposed coVAT2, the reordering of the row indices of R are taken from the VAT-reordering of S_r and the reordering of the column indices are taken from the VAT-reordering of S_c.

[b] Line 6 in Algorithm 2 was erroneously written as $CP(cc) = P(t)$ in the original coVAT article.[45]

Algorithm 2: coVAT1 Algorithm[45]

Input: R - $m \times n$ rectangular dissimilarity matrix
Build estimates of S_r and S_c using Equations 2 and 3, respectively.
Build $S_{r\cup c}$ using Equation 1.
Run VAT on $S_{r\cup c}$, saving permutation array $P_{r\cup c} = \{P(1), \ldots, P(m+n)\}$
Initialize $rc = cc = 0$; $RP = CP = 0$.
for $t = 1, \ldots, m+n$ **do**
  if $P(t) \leq m$ **then**
    $rc = rc + 1$, rc is row component
    $RP(rc) = P(t)$, RP are row indices
  else
    $cc = cc + 1$, cc is column component
    $CP(cc) = P(t) - m$, CP are column indices
Form the coVAT1 ordered rectangular dissimilarity matrix,
$R^* = [R^*_{ij}] = [R_{RP(i)CP(j)}]$, $1 \leq i \leq m$; $1 \leq j \leq n$
Output: Reordered dissimilarity matrices R^*, $\hat{S}_r$, $\hat{S}_c$, and $\hat{S}_{r\cup c}$

Another advantage of this alternate reordering scheme is that the scale factors λ_r and λ_c in (2) and (3) can be ignored.

Algorithm 3: coVAT2 Reordering Scheme

Input: R - $m \times n$ rectangular dissimilarity matrix
Build estimates of S_r and S_c using Equations 2 and 3, respectively (with $\lambda_r = \lambda_c = 1$).
1 Run VAT on S_r, saving permutation array, $RP = \{RP(1), \ldots, RP(m)\}$
2 Run VAT on S_c, saving permutation array, $CP = \{CP(1), \ldots, CP(n)\}$
3 Form the coVAT2 ordered rectangular dissimilarity matrix,
$R^{**} = [R^{**}_{ij}] = [R_{RP(i)CP(j)}]$, $1 \leq i \leq m$; $1 \leq j \leq n$
Output: Reordered dissimilarity matrices, R^{**}, $\hat{S}_r$, and $\hat{S}_c$
Optional: Build and output $\hat{S}_{r\cup c}$ with Equation 1

3.1. Analysis

Consider a partition matrix of n objects $U \in M_{hcn}$, where M_{hcn} is the set of all $c \times n$ partition matrices,

$$M_{hcn} = \left\{ U \in \mathbb{R}^{cn} | u_{ij} \in \{0, 1\}\ \forall i, j; \sum_{i=1}^{c} u_{ij} = 1\ \forall j; \sum_{j=1}^{n} u_{ij} > 0\ \forall i \right\}.$$

The partition element $u_{ij} = 1$ if the jth object is in the ith cluster. In Ref. 49, we proved that all *single-linkage* (SL) partitions are represented as *aligned partitions* in

the VAT-reordered data. In other words, every c-partition, $c = 1, \ldots, n$, produced by SL is represented by a partition matrix U that has c contiguous blocks of 1s beginning in the upper left corner and proceeding down to the right. The set of all aligned partitions are

$$M^*_{hcn} = \left\{U \in M_{hcn} | u_{1j} = 1, 1 \le j \le n_1; u_{ij} = 1, n_{i-1} + 1 \le j \le n_i, 2 \le i \le c\right\}.$$

In both coVAT1 and coVAT2, VAT is used on both the row and column vectors of R, where the result is a VAT-reordered matrix $\hat{S}_r$ of the rowwise relational data and a VAT-reordered matrix $\hat{S}_c$ of the columnwise relational data. In coVAT2, the resulting reordering indices are then applied to R to produce R^{**}.

PROPOSITION 1. *For a rectangular matrix R, coVAT2 produces a reordered matrix* R^{**} *where all the SL c-partitions of the row vectors* $\mathbf{r}^{**}_{i,\cdot}$ *are aligned partitions for* $c = 1, \ldots, m$.

PROPOSITION 2. *For a rectangular matrix R, coVAT2 produces a reordered matrix* R^{**} *where all the SL c-partitions of the column vectors* $\mathbf{r}^{**}_{\cdot,j}$ *are aligned partitions for* $c = 1, \ldots, n$.

Proof. The proofs of these propositions follow directly from Proposition 1 in Ref. 49. ■

Remark. It follows from Propositions 1 and 2 that the SL dendrograms of the row vectors and column vectors can be superimposed on the coVAT2 image of R^{**}. Furthermore, the computation elements of producing these dendrograms are already included in the coVAT2 algorithm. This feature of coVAT2 could be especially useful for microarray analysis if you wish to visualize the dendrogram of gene clusters or treatment clusters independently.

Note that Propositions 1 and 2 do not apply to coVAT1 because coVAT1 uses the reordering indices of $\hat{S}_{r \cup c}$ to produce R^*. Hence, this is another reason why we believe that coVAT2 is the preferred method.

Another property of coVAT2 that separates it from coVAT1 is that coVAT2 works on both relational and feature data, where coVAT1 only works on relational data. Because the row and column reordering in coVAT2 is performed separately by VAT, the data in R can be real numbers. A good example of data for which coVAT2 works, but coVAT1 does not, is microarray data. These data can take both positive and negative values, where positive indicates upregulation of a gene and negative indicates downregulation. The absolute values of these data can be computed to produce relational data, but we argue that this operation discards meaningful information. The advantage of coVAT2 is that it can produce tendency visualizations for both the real-valued microarray data and the absolute-value processed relational data. Extending this line of thought, coVAT2 is also able to process tertiary relational data: data that have supporting, negating, and no-information values (e.g., yea, nay,

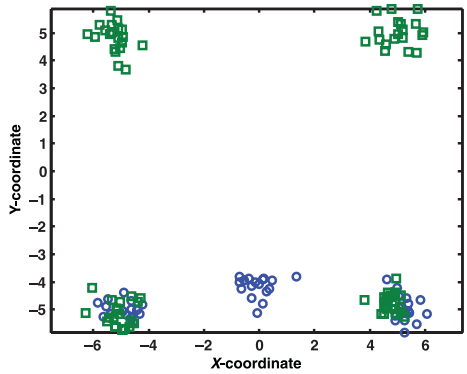
and no-vote). We now move on to examples that demonstrate why we believe that coVAT2 is a more effective cluster tendency visualization tool than coVAT1.

4. EXAMPLES

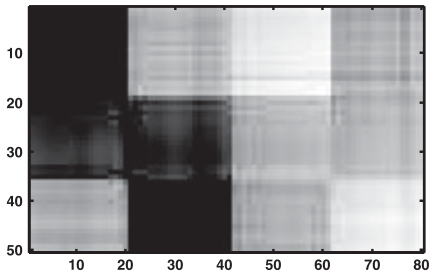
4.1. Comparison of coVAT1 and coVAT2 on Original Examples

The results in this section compare coVAT2 to the original formulation, coVAT1, using selected examples from the original coVAT paper. The data shown here are not exactly the same as the data presented in Ref. 45 as the original data sets were unavailable. We mimicked the examples from Ref. 45 as closely as possible. We only show the image of the reordered rectangular dissimilarity matrix, using both coVAT1 and coVAT2, as the method for building $\hat{S}_r$, $\hat{S}_c$, and $\hat{S}_{r\cup c}$ is unchanged. For these examples, a Euclidean distance was used to compute R.

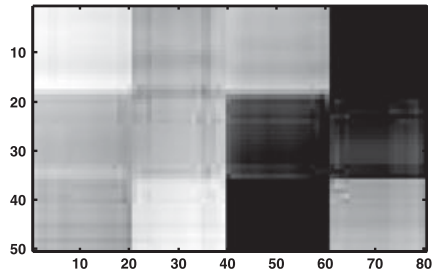
Example 1. Figure 2 compares the (coclustering only) results of coVAT1 and coVAT2 for the data set shown in Figure 1 in Ref. 45. Both formulations are



(a) Object data - row objects (○), column objects (□)

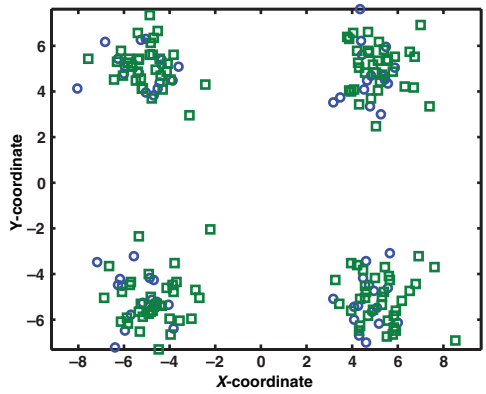


(b) coVAT1 R^*

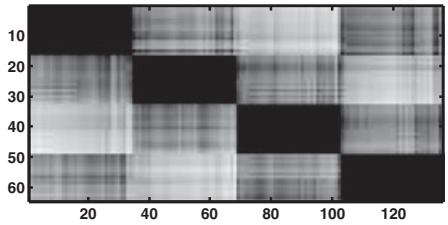


(c) coVAT2 R^{**}

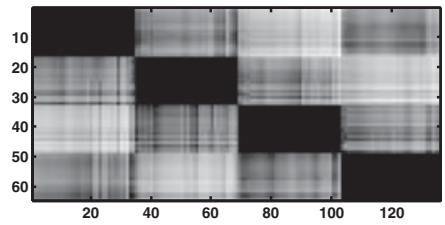
Figure 2. Example from Figure 1 in Ref. 45: (a) object data, (b) coVAT1 reordering, and (c) coVAT2 reordering.



(a) Object data - row objects (○), column objects (□)



(b) coVAT1 R^*



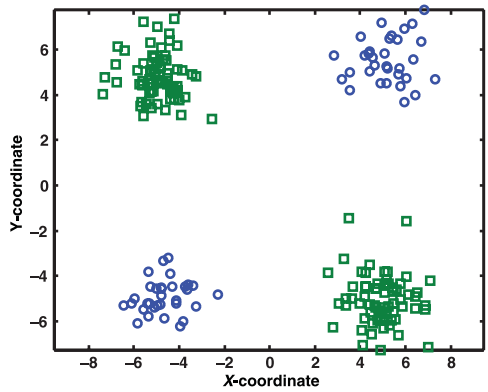
(c) coVAT2 R^{**}

Figure 3. Example from Figure 4 in Ref. 45: (a) object data, (b) coVAT1 reordering, and (c) coVAT2 reordering.

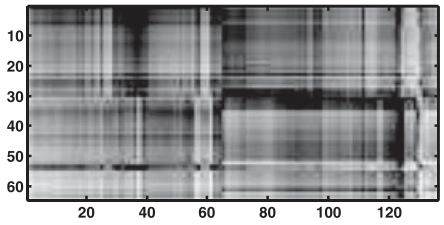
effective in showing that this data set contains two coclusters—the left and right clusters at the bottom of Figure 2a. Note that although the images are not exactly similar, they both show the same cluster structure.

Example 2. The results of the algorithms on the data set shown in Figure 4 in Ref. 45 are shown in Figure 3. This data set was called $R_{64\times136}$ as there are 64 row objects and 136 column objects. This represents the "best case" scenario for coVAT as the imputed matrices S_r and S_c will have the least amount of error. Visual inspection of Figure 3a shows that there are four coclusters in this data, and Figures 3b and 3c both contain four dark blocks, showing this to be the case.

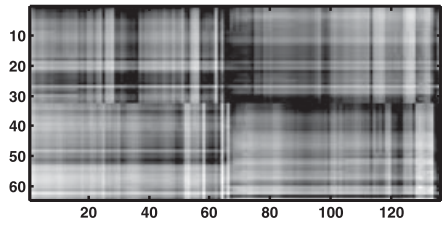
Example 3. Finally, Figure 4 compares the results of coVAT1 and coVAT2 for the data set shown in Figure 7 in Ref. 45. This example represents the opposite extreme to the data in Figure 3a. The data in Figure 4a have no coclusters (none of the four clusters there contain both squares and circles). In this case, we expect coVAT1 and coVAT2 to show no tendency toward coclusters. Neither Figures 4b or 4c contain



(a) Object data - row objects (○), column objects (□)



(b) coVAT1 R^*



(c) coVAT2 R^{**}

Figure 4. Example from Figure 7 in Ref. 45: (a) object data, (b) coVAT1 reordering, and (c) coVAT2 reordering.

any dark block structure, so both algorithms are successful in their attempt to assess coclustering tendency—there is none!

4.2. Synthetic Relational Data

The examples shown in this section show *pure* relational data, or data for which no object representation exists. These are in contrast to the examples shown in the original coVAT paper, where the dissimilarity matrices were built from Euclidean distances between object data pairs. In these examples, we start with the rectangular matrix R.

The three rectangular relational matrices used in this section were built by assigning rectangular blocks within the matrix a dissimilarity of 0 (or perfect similarity) and the remaining object pairs a dissimilarity of 1. A low-level additive uniform noise was then applied to the elements of each matrix. Finally, the row and column objects were randomly permuted as to "break up" the contiguous blocks of similar objects. If the reordering algorithms succeed, we would expect to see the original rectangular blocks of low dissimilarity to be reconstructed in the images.

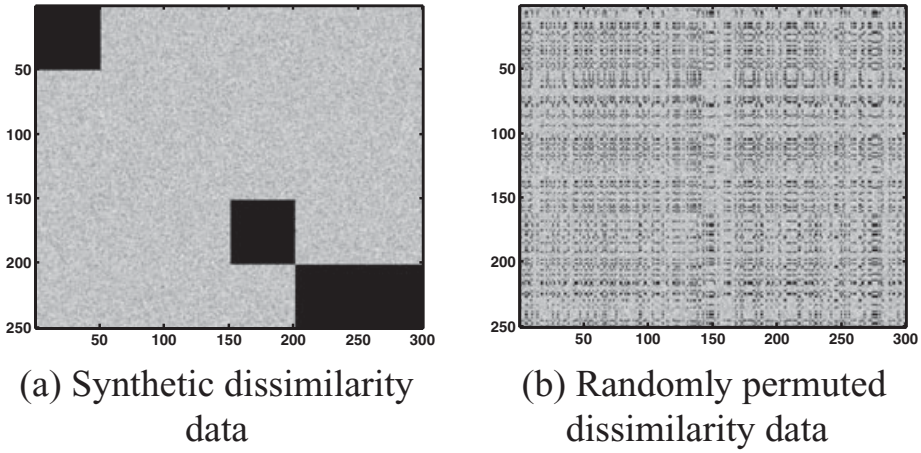


Figure 5. Example of synthetic data construction.

Figure 5 shows an example of a synthetic rectangular relational matrix we constructed that has three rectangular dark blocks and the randomly permuted version that is used as input to the coVAT algorithms.

Example 4. This example shows the utility of coVAT2 on relational data that are not derived from object data (as were shown in the original article and in the previous examples). Figure 6a shows a dissimilarity matrix of 250 row objects and 300 column objects. By our construction, the input data contain four row clusters, four column clusters, five clusters in the union of rows and columns, and three coclusters. Figures 6b–6f show the coVAT images of this dissimilarity data. Both coVAT1, shown in Figure 6b, and coVAT2, in Figure 6c, clearly show that there are three co-clusters. Figure 6d indicates that there are four clusters of row objects; Figure 6e indicates that there are four clusters of column objects. Finally, Figure 6f illustrates that in the union of the row and column objects there are five clusters.

We show this example first because this clearly shows how both coVAT1 and coVAT2 provide a visualization of the clustering tendency of the different types of clusters in pure relational data. It also provides a basis for our claims described in the next examples.

Example 5. This example is on a pure relational data set that has 250 row objects and 300 column objects. We constructed this data to have three row clusters, five column clusters, and five rectangles of strong pairwise similarity in the rectangular matrix R. The dissimilarity data for these objects is shown in Figure 7a. Clearly, Figure 7a does not indicate any discernable cluster structure. Similarly, the coVAT1 image in Figure 7b also does not show any cluster structure. At the very least, this image shows that there is a group of column objects (indexed 1–50) that do not have a strong similarity to *any* row objects.

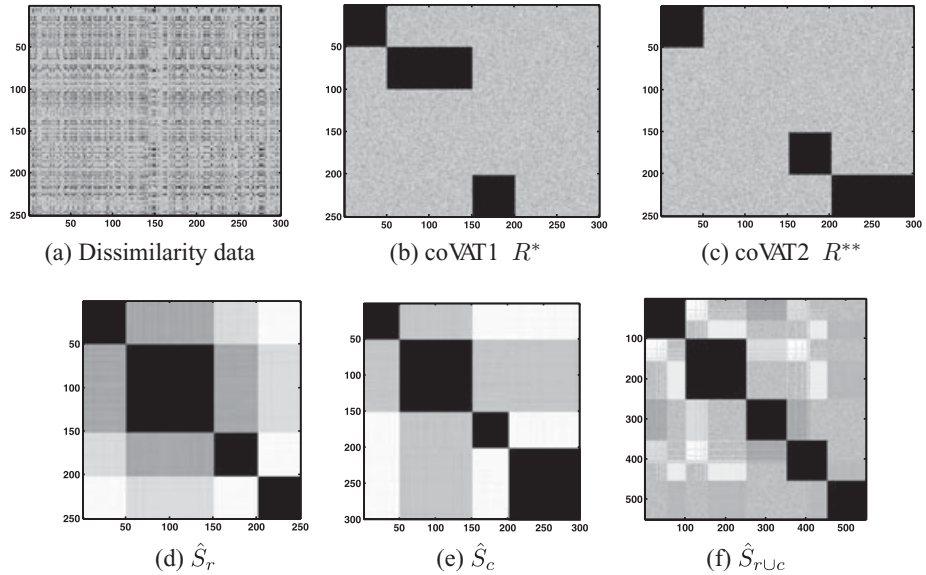


Figure 6. Example 4. Pure rectangular relational data: (a) dissimilarity data, (b) coVAT1, (c) coVAT2, and (d)–(f) reordered square dissimilarity data matrices.

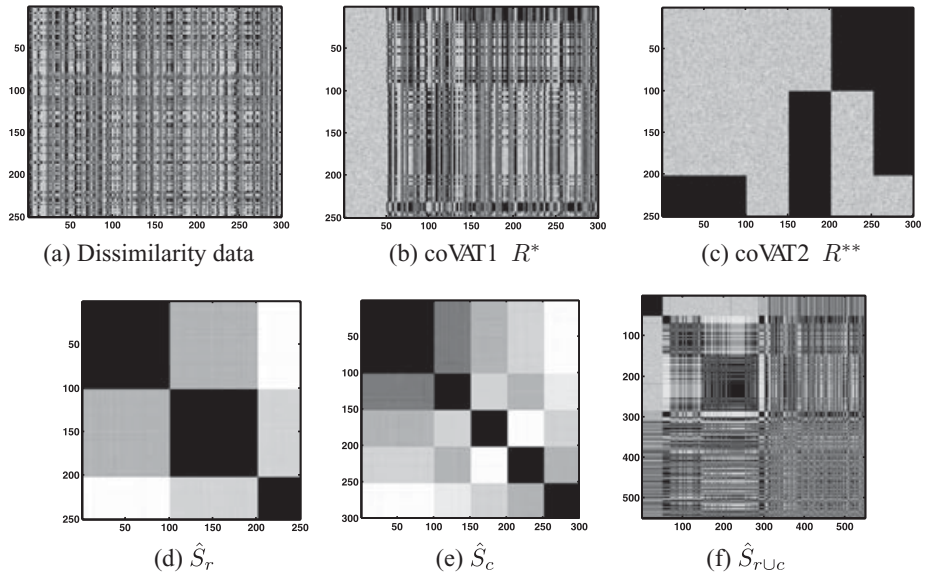


Figure 7. Example 5. Pure rectangular relational data: (a) dissimilarity data, (b) coVAT1, (c) coVAT2, and (d)–(f) reordered square dissimilarity data matrices.

Interestingly, Figures 7d and 7e of $\hat{S}_r$ and $\hat{S}_c$, respectively, show the cluster tendency of the row and column objects rather well: three clusters in the row objects and five clusters in the column objects. But, the reason that coVAT1 "fails" on this example is because the reordering of $\hat{S}_{r\cup c}$ "fails". Figure 7f shows $\hat{S}_{r\cup c}$. This image shows no clear cluster structure in the union of the row and column objects. Thus, when the reordering indices are reshuffled to produce the coVAT1 image, this image also fails to show the structure.

However, if we use coVAT2 to produce the reordering of R, shown in Figure 7c, the cluster structure of these data is very clear. coVAT2 is able to reconstruct the five rectangles of strong similarity in R^{**}; however, notice that there is overlap between all of the objects that have a strong similarity to another object (the column objects indexed 100–150 are the same column objects indexed 1–50 in the coVAT1 image). The row objects indexed 200–250 have a strong similarity to column objects 1–100 and 150–200. However, the column objects 150–200 are also strongly similar to the row objects 100–200. And these row objects are strongly similar to the column objects 250–300. Thus, we believe that a question remains as to the number of pure coclusters in this data set. Also, it is our conjecture that these overlaps, which did not exist in the previous example, are the reason that coVAT1 "fails" for this data set.

Example 6. This example is on a pure relational data set that has 250 row objects and 300 column objects. By design, the "true, but unknown" numbers of row and column clusters are six and five, respectively, with five rectangles of strong similarity in R. The dissimilarity data for these objects is shown in Figure 8a. As in the previous example, the coVAT1 image shown in Figure 8b does not suggest any cluster structure. Again, the images of $\hat{S}_r$ and $\hat{S}_c$ clearly show structure: six clusters in the row objects and five clusters in the column-objects. The image of $\hat{S}_{r\cup c}$ again shows no clear cluster structure.

As shown in Figure 8c, coVAT2 produces a visually pleasing and informative image, reconstructing five rectangles of strong pairwise similarity. However, as we saw in the previous example, there is overlap between the dark blocks—row objects 175–200 are strongly similar to column objects 250–300. Thus, the question remains as to how many pure coclusters these five rectangles represent. This example also provides more evidence for our conjecture that overlapping is the cause of the failure of coVAT1.

4.3. Real Data

For our next example, we turn to a set of real data.

Example 7. These data consist of 1984 records of 16 key votes by the 435 members of the U.S. House of Representatives (available on the UCI data depository[50]). We coded the data as 1 for yea, 0 for nay, and 0.5 for unknown. We chose this coding so that coVAT1 could also be used with these data. But please note that a more conventional $\{-1, 0, 1\}$ coding could be used with coVAT2, producing the same visualization results. Figure 9a shows the raw rectangular data for this example.

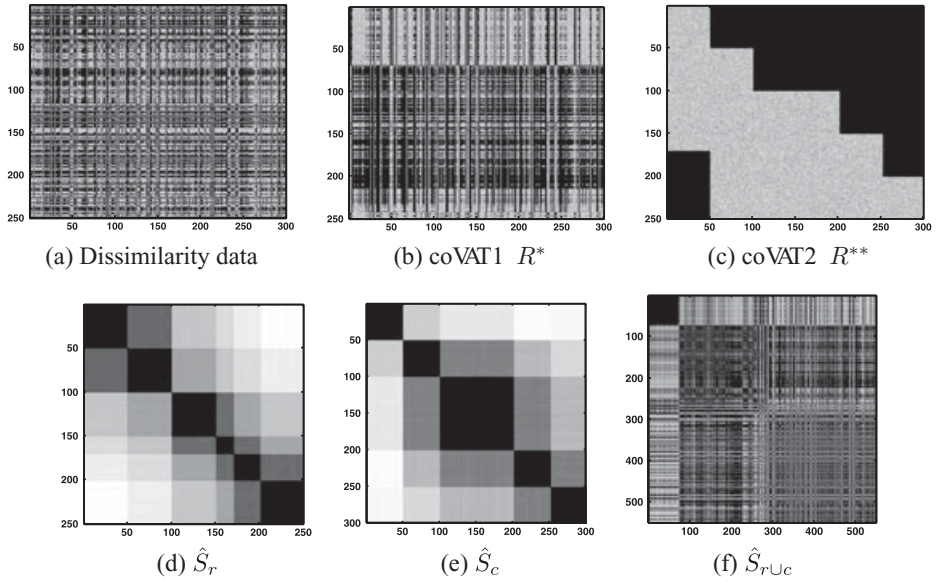


Figure 8. Example 6. Pure rectangular relational data: (a) dissimilarity data, (b) coVAT1, (c) coVAT2, and (d)–(f) reordered square dissimilarity data matrices.

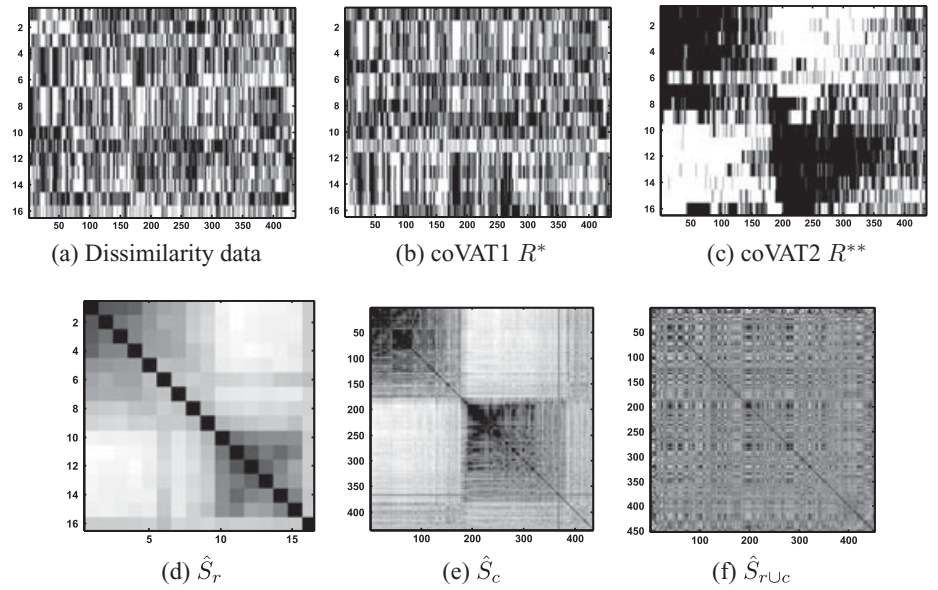
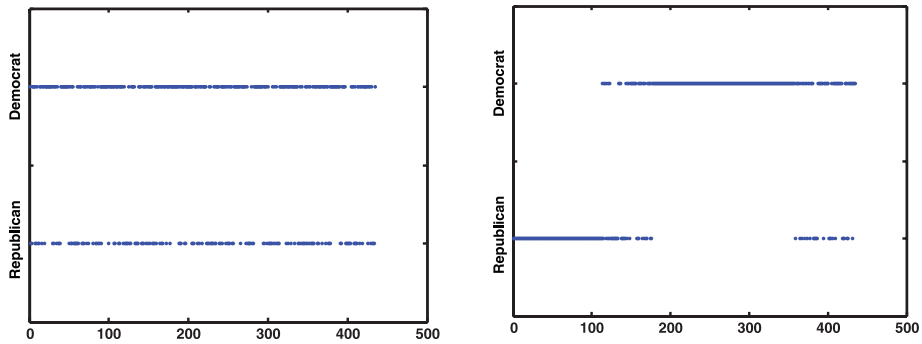


Figure 9. Example 7. (a) Rectangular VOTE data, (b) coVAT1, (c) coVAT2, and (d)–(f) reordered square dissimilarity data matrices.



(a) coVAT1-reordered column object labels

(b) coVAT2-reordered column object labels

Figure 10. Labels ordered according to column object reordering of coVAT1 and coVAT2—shows that coVAT2 discovers party affiliation patterns through voting records.

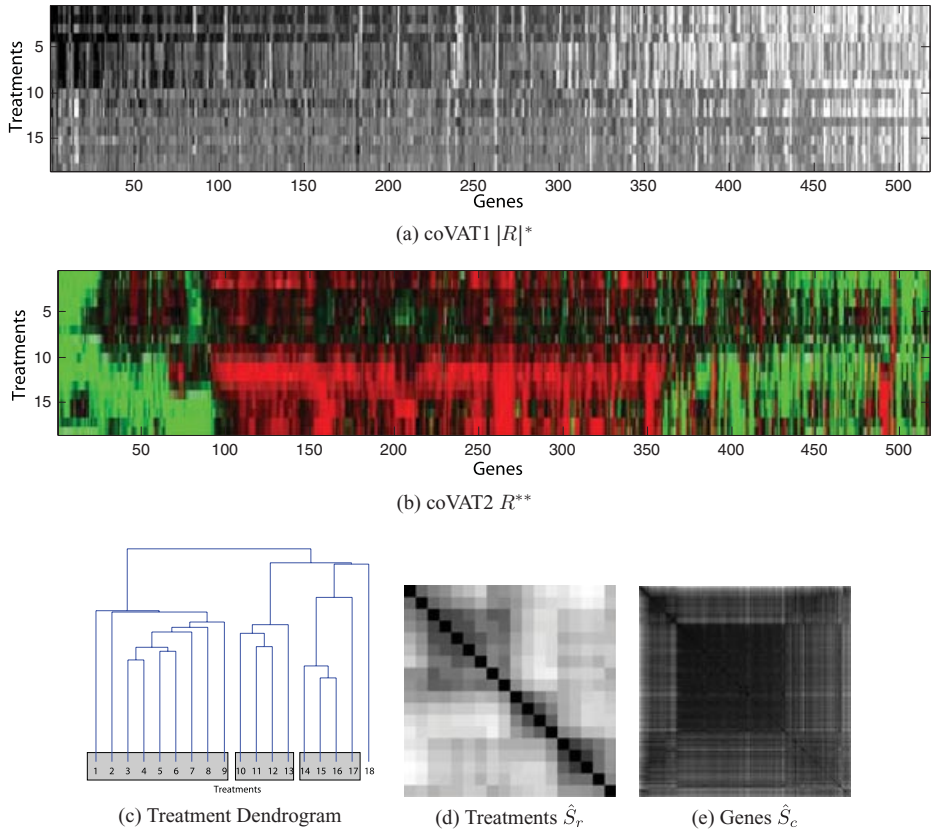
Figures 9b and 9c show the coVAT1 and coVAT2 reordered rectangular matrices, respectively. Again, the coVAT2 reordering produces a more appealing image, from the standpoint of assessing cluster tendency. There appear to be the underpinnings for two larger coclusters—one dark blob at the upper-left and one at the middle-bottom of Figure 9c. These two coclusters correspond to Republicans and Democrats, respectively, that tend to vote along party lines as well as the votes (the columns) on which they tend to agree on. Figure 9b, produced by coVAT1, clearly does not provide as pleasing a result as Figure 9c does.

Figure 10 shows the labels of the column objects, plotted in the order imposed by the coVAT1 and coVAT2 reorderings—Figures 9b and 9c—respectively. The plots in Figure 10 clearly support the visual evidence that the coVAT2 reordering is superior to that of coVAT1. As the figures show, coVAT2 is able to show how the members of each party cluster together; where as the coVAT1 reordering shows no such pattern.

Figures 9d–9f show the other coVAT images. Figure 9e shows the two clusters of party members that tend to vote along party lines (Figure 10b shows the labels (Republican and Democrat) according the reordering of S_c). Figure 9f shows the cluster tendency in the union of the members and votes. This view does not seem to suggest any cluster structure in the union of the objects. The failure of VAT to elucidate the cluster structure from $S_{r\cup c}$ is the reason that the coVAT1 image in Figure 9b is inferior to the coVAT2 image shown in Figure 9c.

Example 8. The data in this example were measured in a microarray experiment. They are expression values of 517 genes in the presence of 18 treatments.[c] Expression values are negative for downregulated genes and positive for upregulated

[c] This data set can be downloaded from http://genome-www.stanford.edu/serum/. Cluster analysis was first performed on these data in Ref. 51.



(a) coVAT1 $|R|^*$

(b) coVAT2 R^{**}

(c) Treatment Dendrogram

(d) Treatments $\hat{S}_r$

(e) Genes $\hat{S}_c$

Figure 11. Example 8: (a) coVAT1 of $|R|$, (b) coVAT2 of R, green indicates upregulation and red indicates downregulation, (c) SL dendrogram of treatments, (d) reordered treatment dissimilarity, and (e) reordered gene dissimilarity.

genes. These values can be thought of as relational data by considering large values (both positive and negative) as indicating a strong relationship between a gene and a treatment; values near 0 indicate no relationship. We created two relational matrices for these data: $|R|$, which is the absolute value of the expression, and R, which is the unmodified expression values.

Figure 11a sows the coVAT1 image of $|R|$ (recall that coVAT1 cannot be used with negative relation values). This image appears to show some cluster structure in the upper left and in the upper right; however, if you compare the coVAT1 image in Figure 11a to the coVAT2 Figure 11b, it is clear that coVAT1 is not nearly as effective as coVAT2. Figure 11b shows the coVAT2 image of R, where green indicates upregulation and red indicates downregulation—the brighter the color the more the gene expressed. The coVAT2 image shows clear cluster structure of genes that upregulate on the left and right of the image, and a large group of downregulating genes in the middle of the image. Also interesting is the group of genes centered

at index 50 that downregulate for some treatments and upregulate for others, all behaving similarly.

Figure 11c shows the dendrogram of the 18 treatments in coVAT2 order; hence, the ordering of the treatments in Figures 11b and 11c are the same. This dendrogram clearly shows that there are three major clusters of treatments and one outlier, treatment 18. The coVAT view of $\hat{S}_r$ in Figure 11d supports this notion by showing three darker blocks along the diagonal. Finally, Figure 11e shows the coVAT image of $\hat{S}_c$, which suggests that there is one large group of genes that are very similar and then several smaller groups. The large dark block in Figure 11e corresponds to large red block of gene expressions indexed 100-350 in Figure 11b.

The images in Figure 11 show that coVAT2 is clearly preferable for examining the cluster tendency of microarray data or other data that has both positive- and negative-valued relations. Furthermore, it is additionally helpful in that the dendrogram can be created for the coVAT2 image, providing yet another look at how the data should cluster.

5. CONCLUSIONS

First, we reviewed several algorithms that provide visual assessment of clustering tendency in square (VAT) and rectangular (coVAT1) dissimilarity data. Then we introduced a new algorithm, coVAT2, for the rectangular case, that reorders the input data in a simple and more direct way than coVAT1. Our first three examples show that coVAT2 does the same job as coVAT1 for all of the original examples in Ref. 45. Example 4 shows them to agree for a simple case of synthetic data, whereas Examples 5 and 6 illustrate cases where coVAT1 fails to indicate the ("true but unknown") cluster structure in the synthetic data, while coVAT2 succeeds. Our last two examples show that coVAT2 works much better than coVAT1 on a small real data set, the VOTE data, and on a small set of microarray data.

To our knowledge, no other methods exist that estimate the potential numbers of clusters in each of the four clustering problems inherent in rectangular data; hence, either of coVAT1 or coVAT2 provide useful information to the coclustering community. coVAT2 enjoys a small advantage in complexity over coVAT1 because it does not require the construction or processing of the (estimate of) the union matrix $\hat{S}_{r\cup c}$. And our examples show that coVAT2 is a better choice than coVAT1 for visual assessment of coclustering tendency. coVAT2 also works with relational data that has both positive (affirming) and negative (dissenting) values, providing an additional reason why coVAT2 is preferred.

Because the coVAT algorithms utilize the VAT algorithm, and VAT has been extended to very large data sets by sVAT,[39] coVAT1 and coVAT2 are also useful for processing large-scale data sets. The use of sVAT with coVAT1 was called scoVAT1.[46] Our next project will be to benchmark scoVAT1 against the obvious extension of coVAT2 to scoVAT2, and compare the results on large-scale dissimilarity data.

Finally, the issue of interpreting the "proper" number of coclusters that is raised in each of Examples 5–7 is an interesting one. If we define "dark blocks"

as squares or rectangles of uniform intensity, we remove ambiguity in several of these examples but also lose any semblance of interpretability when the images do not possess uniformity, as seen in Figure 9c. This also happens in the square case. Therefore, different viewers may well see different "answers" in the same image. So this branch of cluster analysis is very much the same as other forms of the art: A human will always be needed to judge whether or not an algorithmic output is acceptable. The same test always applies—is it useful, to you, today?

Acknowledgments

Havens is supported by the National Science Foundation under Grant #1019343 to the Computing Research Association for the CI Fellows Project.

References

1. Theodoridis S, Koutroumbas K. Pattern recognition. San Diego, CA: Academic Press; 2009.
2. Dhillon IS. Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc 7th ACM SIGKDD Int Conf on Knowledge Discovery Data Mining 2001. pp 269–274.
3. Hartigan JA, Wong MA. A *K*-means clustering algorithm. Appl Stat 1979;28:100–108.
4. Bezdek JC. Pattern recognition with fuzzy objective function algorithms. New York: Plenum; 1981.
5. Kummamuru K, Dhawale A, Krishnapuram R. Fuzzy co-clustering of documents and keywords. In: Proc IEEE Int Conf Fuzzy Systems 2003. pp 772–777.
6. Tjhi W, Chen L. A heuristic-based fuzzy co-clustering algorithm for categorization of high-dimensional data. Fuzzy Sets Syst 2008;159:371–389.
7. Tjhi W, Chen L. Minimum sum-squared residue for fuzzy co-clustering. Intell Data Anal 2006;10:237–249.
8. Tjhi W, Chen L. Dual fuzzy-possibilistic coclustering for categorization of documents. IEEE Trans Fuzzy Syst 2009;17:532–543.
9. Kohonen T. Self-organizing maps. Berlin, Germany: Springer; 2001.
10. Chernoff H. The use of faces to represent points in *k*-dimensional space. J Am Stat Assoc 1973; 68:361–368.
11. Kleiner B, Hartigan JA. Representing points in many dimensions by trees and castles. J Am Stat Assoc 1981;76:260–269.
12. Sammons JW. A nonlinear mapping for data structure analysis. IEEE Trans Comput 1969;18:401–409.
13. Hathaway RJ, Bezdek JC. Visual cluster validity for prototype generator clustering models. Pattern Recog Lett 2003;24:1563–1569.
14. Bezdek JC, Chiou EW. Core zone scatterplots: a new approach to feature extraction for visual displays. Comput Vision Graph Image Process 1988; 41:186–209.
15. Wilkinson L, Friendly M. The history of the cluster heat map. Am Stat 2009;63:179–184.
16. Loua T. Atlas statistique de la population de paris. Paris, France: J. Dejey; 1973.
17. Czekanowski J. Zur differentialdiagnose der Neandertalgruppe. Korrespondenzblatt der Deutschen Gesellschaft fÂ‚r Anthropologie, Ethnologie und Urgeschichte 1909;40:44-47.
18. Tryon RC. Cluster analysis. Ann Arbor, MI: Edwards Bros.; 1939.
19. Cattell RB. A note on correlation clusters and cluster search methods. Psychometrika 1944;9:169–184.
20. Guttman L. Measurement and prediction. The American soldier. New York: Wiley; 1950.

21. Mayr E, Linsley E, Usinger R. Methods and principles of systematic zoology. New York: McGraw-Hill; 1953.
22. Torgerson WS. Theory and methods of scaling. New York: Wiley; 1958.
23. Sneath PHA. A computer approach to numerical taxonomy. J Gen Microbiol 1957; 17:201–226.
24. Floodgate GD, Hayes PR. The Adansonian taxonomy of some yellow pigmented marine bacteria. J Gen Microbiol 1963;30:237–244.
25. Ling RF. A computer generated aid for cluster analysis. Commun ACM 1973;16:355–361.
26. Johnson DA, Wichern DW. Applied multivariate statistical analysis. Englewood Cliffs, NJ: Prentice Hall; 2007.
27. Tran-Luu T.D. Mathematical concepts and novel heuristic methods for data clustering and visualization. Ph.D. Dissertation, University of Maryland, College Park, MD, 1996.
28. West DB. Introduction to graph theory. Englewood Cliffs, NJ: Prentice Hall; 2001.
29. George A, Liu J. Computer solution of large sparse positive definite systems. Englewood Cliffs, NJ: Prentice-Hall; 1981.
30. King IP. An automatic reordering scheme for simultaneous equations derived from network analysis. Int J Numer Methods Eng 1970;2:523–533.
31. Sloan SW. An algorithm for profile and wavefront reduction of sparse matrices. Int J Numer Methods Eng 1986;23:239–251.
32. Weinstein J. A postgenomic visual icon. Science 2008;319:1172–1173.
33. Dhillon IS, Mallela S, Modha DS. Information-theoretic co-clustering. In: Proc 9th ACM SIGKDD Int Conf on Knowledge Discovery Data Mining; 2003. pp 89–98.
34. Cho H, Dhillon IS, Guan Y, Sra S. Minimum sumsquared residue co-clustering of gene expression data. In: Proc 4th SIAM Int Conf Data Mining 2004. pp 114–125.
35. Frigui H, Nasraoui O. Simultaneous clustering and attribute discrimination. In: Proc FUZZ-IEEE 2000. pp 158–163.
36. Frigui H. In: Oliveira JV, Pedrycz W, editors. Advances in fuzzy clustering and feature discrimination with applications. New York: Wiley; 2007. pp 285–312.
37. Banerjee A, Dhillon IS, Ghosh J et al. A generalized maximum entropy to Bregman co-clustering and matrix approximation. J Mach Learn Res 2007;8:1919–1986.
38. Bezdek JC, Hathaway RJ. VAT: a tool for visual assessment of (cluster) tendency. In: Proc. IJCNN 2002; pp. 2225–2230.
39. Hathaway RJ, Bezdek JC, Huband JM. Scalable visual asseessment of cluster tendency for large data sets. Pattern Recog 2006;39:1315–1324.
40. Wang L, Nguyen TVU, Bezdek JC et al. iVAT and aVAT: enhanced visual analysis for cluster tendency assessment. In: Proc. PAKDD 2010; 2010.
41. Fisher B, Zoller T, Buhmann J. Path based pairwise data clustering with application to texture segmentation. In: Proc Third Int Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition, Sophia Antipolis, France, September 3–5, 2001. Lecture Notes Comput Sci 2001;2134:235–250.
42. Havens TC, Bezdek JC. An efficient formulation of the improved visual assessment of cluster tendency (iVAT) algorithm. IEEE Trans Knowl Data Eng 2012;24(5). doi: 10.1109/TKDE.2011.33.
43. Bezdek JC, Havens TC, Keller JM, Leckie CA, Park L, Palaniswami M. Clustering elliptical anomalies in sensor networks. In: Proc IEEE Int Conf on Fuzzy Systems; 2010. pp 1–8.
44. Moshtaghi M, Havens TC, Bezdek JC, Park L, Leckie CA, Rajasegarar S, Keller JM, Palaniswami M. Clustering ellipses for anomaly detection. Pattern Recog 2010;44:55–69.
45. Bezdek JC, Hathaway RJ, Huband JM. Visual assessment of clustering tendency for rectangular dissimilarity matrices. IEEE Trans Fuzzy Syst 2007;15:890–903.
46. Park L, Bezdek JC, Leckie CA. Visualization of clusters in very large rectangular dissimilarity data. In: Proc 4th Int Conf on Autonomous Robots and Agents; 2009. pp 251–256.
47. Havens TC, Bezdek JC, Keller JM. A new implementation of the co-VAT algorithm for visual assessment of clusters in rectangular relational data. In: Artificial intelligence and soft computing, Part I. Springer-Verlag: Berlin. 2010. pp 363–371.

48. Prim RC. Shortest connection networks and some generalisations. Bell Syst Tech J 1957;36:1389–1401.
49. Havens TC, Bezdek JC, Keller JM, Popescu M, Huband JM. Is VAT really single linkage in disguise?. Ann Math Artif Intell 2009;55:237–251.
50. Asuncion A, Newman DJ. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. Available at http://www.ics.uci.edu/ mlearn/MLRepository.html. 2007.
51. Iyer VR, Eisen MB, Ross DT, et al. The transcriptional program in the response of human fibroblasts to serum. Science 1999;283:83–87.