Enhanced Visual Analysis for Cluster Tendency Assessment and Data Partitioning

Liang Wang, *Senior Member*, *IEEE*, Xin Geng, James Bezdek, *Fellow*, *IEEE*, Christopher Leckie, and Kotagiri Ramamohanarao

Abstract—Visual methods have been widely studied and used in data cluster analysis. Given a pairwise dissimilarity matrix D of a set of n objects, visual methods such as the VAT algorithm generally represent D as an $n \times n$ image $\mathrm{I}(\tilde{D})$ where the objects are reordered to reveal hidden cluster structure as dark blocks along the diagonal of the image. A major limitation of such methods is their inability to highlight cluster structure when D contains highly complex clusters. This paper addresses this limitation by proposing a Spectral VAT algorithm, where D is mapped to D' in a graph embedding space and then reordered to $\tilde{D}'$ using the VAT algorithm. A strategy for automatic determination of the number of clusters in $\mathrm{I}(\tilde{D}')$ is then proposed, as well as a visual method for cluster formation from $\mathrm{I}(\tilde{D}')$ based on the difference between diagonal blocks and off-diagonal blocks. A sampling-based extended scheme is also proposed to enable visual cluster analysis for large data sets. Extensive experimental results on several synthetic and real-world data sets validate our algorithms.

Index Terms—Clustering, VAT, cluster tendency, spectral embedding, out-of-sample extension.

1 Introduction

A general question in the data mining community is how to organize observed data into meaningful structures (or taxonomies). As a tool of exploratory data analysis [36], cluster analysis aims at grouping objects of a similar kind into their respective categories. Given a data set $\mathcal{O}$ comprising n objects $\{o_1, o_2, \ldots, o_n\}$ (e.g., fish, flowers, beers, etc.), (crisp) clustering partitions the data into c groups $C_1, C_2, \ldots, C_c$, so that $C_i \cap C_j = \emptyset, \text{if } i \neq j$ and $C_1 \cup C_2 \cup \cdots \cup C_c = \mathcal{O}$. There have been a large number of data clustering algorithms in the recent literature [24]. In general, clustering of unlabeled data poses three major problems: 1) assessing cluster tendency, i.e., how many clusters to seek or what is the value of c?, 2) partitioning the data into c groups, and 3) validating the c clusters discovered. Given "only" a pairwise dissimilarity matrix $D \in \mathcal{R}^{n \times n}$ representing a data set of n objects (i.e., the original object data is not necessarily available), this paper addresses the first two problems, i.e., determining the number of clusters c *prior to clustering* and partitioning the data into c clusters.

Most clustering algorithms require the number of clusters c as an input parameter, so the quality of the resulting clusters is largely dependent on the estimation of c. For some applications, users can determine the number of clusters with domain-specific knowledge. However, in most real situations, the value of c is unknown and needs to be estimated from the data themselves. Various *postclustering* measures of cluster validity have been proposed to estimate c, e.g., [13], [40], [39], [37], [35], [27] and [9], by attempting to choose the best partition from a set of alternative partitions. In contrast, cluster tendency assessment attempts to estimate c before clustering occurs. Visual methods for various data analysis problems have been extensively studied [7]. In particular, the representation of data structures in an image format has a long and continuous history, e.g., [11], [2], [21], [19], and [34]. The visual representation of pairwise dissimilarity information about a set of n objects is usually depicted as an $n \times n$ image, where the objects are reordered so that the resulting image is able to highlight potential cluster structure in the data. A "useful" reordered dissimilarity image (RDI) highlights potential clusters as a set of "dark blocks" along the diagonal of the image, and can thus be viewed as a visual aid to tendency assessment.

We could generate reordered dissimilarity images using any of the existing schemes in [11], [21], [2], [19], and [34]. For concreteness, this paper focuses on one method for generating RDIs, namely visual assessment of cluster tendency (VAT) of Bezdek and Hathaway [2], although our approach can also be applied to any method that generates RDIs. Several algorithms have extended VAT for related assessment problems [10], [3], [38], [8], [45], e.g., bigVAT [10] offers a way to approximate the VAT-reordered dissimilarity image for very large data sets, and coVAT [3] extends the VAT idea to rectangular dissimilarity data to enable tendency assessment for coclustering. While RDIs have been widely used for data analysis, they are usually only effective at highlighting cluster tendency in data sets that contain compact well-separated clusters. However, many practical applications involve data sets with highly complex structure, which invalidate the

- *L. Wang is with National Lab of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: wangliangnlpr@gmail.com.*
- *J. Bezdek, C. Leckie, and K. Ramamohanarao are with the Department of Computer Science and Software Engineering, The University of Melbourne, Melbourne, Vic 3010, Australia. E-mail: jcbezdek@gmail.com, caleckie@csse.unimelb.edu.au, rao@csse.unimelb.edu.au.*
- *X. Geng is with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and the School of Mathematical Sciences, Monash University, VIC 3800, Australia. E-mail: xgeng@seu.edu.cn.*

Manuscript received 21 Oct. 2008; revised 12 Mar. 2009; accepted 23 Aug. 2009; published online 12 Oct. 2009.
Recommended for acceptance by D. Talia.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-10-0554.
Digital Object Identifier no. 10.1109/TKDE.2009.192.

1041-4347/10/$26.00 © 2010 IEEE Published by the IEEE Computer Society

assumption of compact, well-separated clusters. In this paper, we propose a new approach to generating RDIs that combines VAT with spectral analysis of pairwise data. The resulting Spectral VAT (SpecVAT) images can clearly show the number of clusters c and the approximate sizes of each cluster for data sets with highly irregular cluster structures. Based on spectral VAT, the cluster structure in the data can be reliably estimated by visual inspection. Also, we propose an effective strategy to measure the "goodness" of spectral VAT images for automated determination of the number of clusters c. We also derive a visual clustering algorithm based on the spectral VAT image and its unique block-structured property to partition the data into c groups. By integrating cluster tendency assessment and cluster formation using an RDI, we provide a natural environment for visual cluster validation and interpretation. To handle large data sets, we further propose a feasible approximate solution in a "sampling plus extension" manner to enable both visual cluster tendency estimation and partitioning. A wide range of primary and comparative experiments on synthetic and real-world data sets demonstrate the effectiveness of our algorithms.

In summary, the major contributions of this paper include:

1. the enhanced SpecVAT algorithm for better revealing the hidden cluster structure of complex-shaped data sets;
2. the effective "goodness" measure of the SpecVAT images for automatic assessment of cluster tendency;
3. the effective visual data partitioning algorithm based on the SpecVAT image; and
4. the extended strategy based on sampling plus out-of-sample extension for scaling the SpecVAT algorithm to larger data sets.

It should be mentioned that a preliminary version of this work has appeared in [44]. In contrast to the previous version, major changes of this paper are summarized as follows:

a. We modify the organization of the paper for better readability, as well as describing each of the proposed algorithms in more detail, including the theoretical analysis of runtime complexity.
b. We provide several additional real-world data sets to evaluate our previously proposed algorithms.
c. We propose a new strategy to extend the previous algorithms to make them feasible for use with truly large data sets.
d. We provide extensive experiments on several synthetic and real data sets to evaluate this new algorithm, and the results demonstrate its effectiveness.

The remainder of the paper is organized as follows: Section 2 illustrates our spectral VAT algorithm. Section 3 presents our strategy for automatically determining the number of clusters c from the SpecVAT images, and Section 4 proposes how to find the c clusters from the SpecVAT image. Section 5 extends the spectral VAT algorithm to deal with large data sets. The experimental results are given in Section 6, prior to discussion and conclusion in Section 7.

2 Spectral VAT

Our work is built upon the VAT algorithm [2] (see Appendix). Two important points about VAT are noted here: 1) Only a pairwise dissimilarity matrix D is required as the input. When vectorial forms of objects are available, it is easy to convert them into D using some form of dissimilarity measures. Even when vectorial data are not explicitly available, it is still feasible to use certain flexible metrics to compute a pairwise dissimilarity matrix, e.g., using Dynamic Time Warping (DTW) to match sequences of different lengths. 2) Although the VAT image suggests both the number of and approximate members of object clusters, matrix reordering produces neither a partition nor a hierarchy of clusters. It merely reorders the data to reveal its hidden structure, which can be viewed as an illustrative data visualization for estimating the number of clusters prior to clustering. However, hierarchical structure could be detected from the reordered matrix if the diagonal sub-blocks exist within larger diagonal blocks.

At first glance, a viewer can estimate the number of clusters c from a VAT image by counting the number of dark blocks along the diagonal if these dark blocks possess visual clarity. However, this is not always possible. We find that a dark block appears in the VAT image only when a tight (or ellipsoidal) group exists in the data. For complex-shaped data sets where the boundaries between clusters become less distinct due to either significant overlap or irregular geometries between different clusters, the resulting VAT images will degrade. See Figs. 4a and 5a for examples. Accordingly, viewers may deduce different numbers of clusters from such poor-quality images, or even cannot estimate c at all. This naturally raises a problem of whether we can transform D into a new form D' so that the VAT image of D' can become clearer and more informative about the cluster structure. In this paper, we address this problem by combining the VAT algorithm with spectral analysis of the proximity matrix of the data.

Recently, a number of researchers have used spectral analysis of a graph in applications such as dimensionality reduction [1], image segmentation [23], [20], and data clustering [22]. These spectral methods generally use the eigenvectors of a graph's adjacency (or Laplacian matrix) to construct a geometric representation of the graph. Different methods are strongly connected, e.g., Laplacian Eigenmaps [1] are very similar to the mapping procedure used in a spectral clustering algorithm described in [15]. Let $\mathcal{G}(\mathcal{V}, E, \boldsymbol{W})$ be a weighted undirected graph, where $\mathcal{V}$ is a set of n vertices (e.g., corresponding to n objects $\{o_1, o_2, \ldots, o_n\}$ to be analyzed), $E = [e_{ij}]$ is the edge set with $e_{ij} = 1$ showing that there is a link between vertices i and j and 0 otherwise, and $\boldsymbol{W} = [w_{ij}]$, an $n \times n$ affinity matrix, includes the edge weights, with w_{ij} representing the relation of the edge connecting vertices i and j. Most spectral methods differ in terms of *graph construction* (reflected in E, e.g., the ε-neighborhood graph, the K-nearest neighbor graph, and the fully connected graph), *weighting functions* (reflected in $\boldsymbol{W}$, e.g., simple 0-1 weighting and the commonly used Gaussian function), or *graph Laplacians* (e.g., the unnormalized Laplacian matrix $\boldsymbol{L} = \boldsymbol{M} - \boldsymbol{W}$ and the normalized version $\hat{\boldsymbol{L}} = \boldsymbol{M}^{-1/2}\boldsymbol{L}\boldsymbol{M}^{-1/2}$, where $\boldsymbol{M}$ is a diagonal degree matrix of $\mathcal{G}$, i.e., $m_{ii} = \sum_{j=1}^{n} w_{ij}$).

TABLE 1
Algorithm I: SpecVAT

Input: $\boldsymbol{D} = [d_{ij}]$: an $n \times n$ scaled matrix of pairwise dissimilarities
k: the number of eigenvectors used

(1): Compute a local scale parameter σ_i for object o_i using $\sigma_i = d(o_i, o_K) = d_{iK}$ where o_K is the K-th nearest neighbor of o_i.

(2): Construct the weighting matrix $\boldsymbol{W} \in \mathcal{R}^{n \times n}$ by defining $w_{ij} = \exp(-d_{ij}d_{ji}/(\sigma_i\sigma_j))$ for $i \neq j$, and $w_{ii} = 0$.

(3): Construct the normalized Laplacian matrix $\boldsymbol{L'} = \boldsymbol{M}^{-1/2}\boldsymbol{W}\boldsymbol{M}^{-1/2}$.

(4): Choose the k largest eigenvectors of $\boldsymbol{L'}$ to form the matrix $\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k] \in \mathcal{R}^{n \times k}$ by stacking the eigenvectors in columns.

(5): Normalize the rows of $\boldsymbol{V}$ with unit Euclidean norm to generate $\boldsymbol{V'}$.

(6): For $i = 1, 2, \cdots, n$, let $\boldsymbol{u}_i \in \mathcal{R}^k$ be the vector corresponding to the i-th row of $\boldsymbol{V'}$ and treat it as a new instance (corresponding to o_i). Then construct a new pairwise dissimilarity matrix $\boldsymbol{D'}$ between instances.

(7): Apply the VAT algorithm to $\boldsymbol{D'}$ to obtain the image $\mathrm{I}(\tilde{\boldsymbol{D}}')$.

Output: Spectrally-mapped and reordered dissimilarity matrix $\tilde{\boldsymbol{D}}'$ and its corresponding scaled gray-scale image $\mathrm{I}(\tilde{\boldsymbol{D}}')$

The spectral decomposition of the Laplacian matrix provides useful information about the properties of the graph. It has been shown experimentally that natural groups in the original data space may not correspond to convex regions, but once they are mapped to a spectral space spanned by the eigenvectors of the Laplacian matrix, they are more likely to be transformed into tight clusters [22], [6]. Based on this observation, we wish to embed $\boldsymbol{D}$ in a k-dimensional spectral space, where k is the number of eigenvectors used, such that each original data point is implicitly replaced with a new vector instance in this new space. After a comprehensive study of recent spectral methods, we adopt a combination of adjacency graph, weighting function, and graph Laplacian for obtaining a better graph embedding (and thus, better SpecVAT images, see Figs. 4b and 5b for examples). We summarize our spectral VAT algorithm in Table 1. Several points about this algorithm are noted as follows:

- Using a specific local scaling parameter allows self-tuning of the object-to-object distance according to the local statistics of the neighborhood surrounding objects i and j, resulting in high affinities within clusters and low affinities across clusters, which has been demonstrated in [26] to be advantageous for clustering.
- The normalized Laplacian matrix

$$\hat{L} = \boldsymbol{M}^{(-1/2)}(\boldsymbol{M} - \boldsymbol{W})\boldsymbol{M}^{(-1/2)} = \boldsymbol{I} - \boldsymbol{L}',$$

  where $\boldsymbol{I}$ is the identity matrix. Replacing $\boldsymbol{I} - \boldsymbol{L}'$ with $\boldsymbol{L}'$ in our algorithm only changes the eigenvalues from $1 - \lambda_i$ to λ_i and not the eigenvectors. The importance of normalization has been analytically demonstrated when the matrix is potentially block diagonal with nonconstant blocks [15], [23].
- Step (6) is actually a graph-embedding representation [1], [33]. That is, the space spanned by the top k eigenvectors of $\boldsymbol{L}'$ is the rank-k subspace that best approximates $\boldsymbol{W}$, in which the original objects are implicitly transformed into new representations.
- The computational complexity of the SpecVAT algorithm depends mainly on three parts, i.e., computing the local scale parameters σ_i, the eigendecomposition of the normalized Laplacian matrix $\boldsymbol{L}'$, and performing the VAT algorithm. The corresponding (worst case) runtime complexities for these three parts are, respectively, $\mathrm{O}(Kn^2)$, $\mathrm{O}(n^3)$, and $\mathrm{O}(n^2)$. Thus, the total computational complexity of the SpecVAT algorithm is $\mathrm{O}(n^3 + (K+1)n^2)$.

3 Automatic Cluster Tendency Assessment

Clustering in unlabeled data $\mathcal{O}$ is the assignment of labels to the objects in $\mathcal{O}$, where two necessary ingredients are, respectively, the number of groups to seek, c, and a partitioning method to discover the c clusters. In this section, we explore the use of spectral VAT for the problem of cluster tendency assessment. That is, can we automatically determine the number of clusters c, as suggested by $\mathrm{I}(\tilde{D}')$, in an objective manner, without viewing the visual display? Before designing an automatic method for estimating the numbers of clusters from the SpecVAT images, let us examine the characteristics of the SpecVAT images first. Figs. 1a and 1c show two examples of the original VAT image and SpecVAT images with different numbers of eigenvectors (corresponding to synthetic data sets S-1 and S-4 shown in Fig. 3, where S-1 is a combination of three circles with the same centroid but different radii, i.e., $c = 3$, and S-4 is a combination of four compact groups hidden in one cluttered background, i.e., $c = 5$). From Fig. 1, we can see that the SpecVAT images are generally clearer than the original VAT image in revealing real data structure. See Figs. 4 and 5 for more examples.

To enable automatic determination of the number of clusters, we need to find a "best" SpecVAT image in terms of "clarity" and "block structure." Each of the "block regions" in the image corresponds to either intracluster or intercluster dissimilarity values, while the "clarity" is relevant to the degree of the brightness difference between such blocks. The corresponding gray-scale histograms in Figs. 1b and 1d suggest that a good SpecVAT image should, ideally, include two explicit modalities in the gray-scale histogram, with a narrow distribution of each modality and a large distance between the two modalities. It is easily understood that the two modalities in the histogram implicitly correspond to "within-cluster distances" (diagonal dark-block regions) and "between-cluster distances" (off-diagonal non-dark-block regions). A narrow distribution for any one modality means that values in either "within-cluster distances" or "between-cluster distances" are close, whereas a big distance between two modalities means that these two modalities are easily distinguished. A nonparametric thresholding method for image binarization is proposed in [17], where only the gray-level histogram suffices without other prior knowledge. We borrow its idea of deriving an optimal threshold to establish an appropriate criterion for evaluating the "goodness" of the SpecVAT images from a more general standpoint.

Following [17], let the pixels of an image be represented in L gray levels. The number of pixels at level l is denoted by m_l and the total number of pixels by $N = \sum_{l=1}^{L} m_l$. Such a gray-level histogram may be normalized and regarded as a probability distribution, i.e., $p_l = m_l/N, p_l > 0, \sum_{l=1}^{L} p_l = 1$.

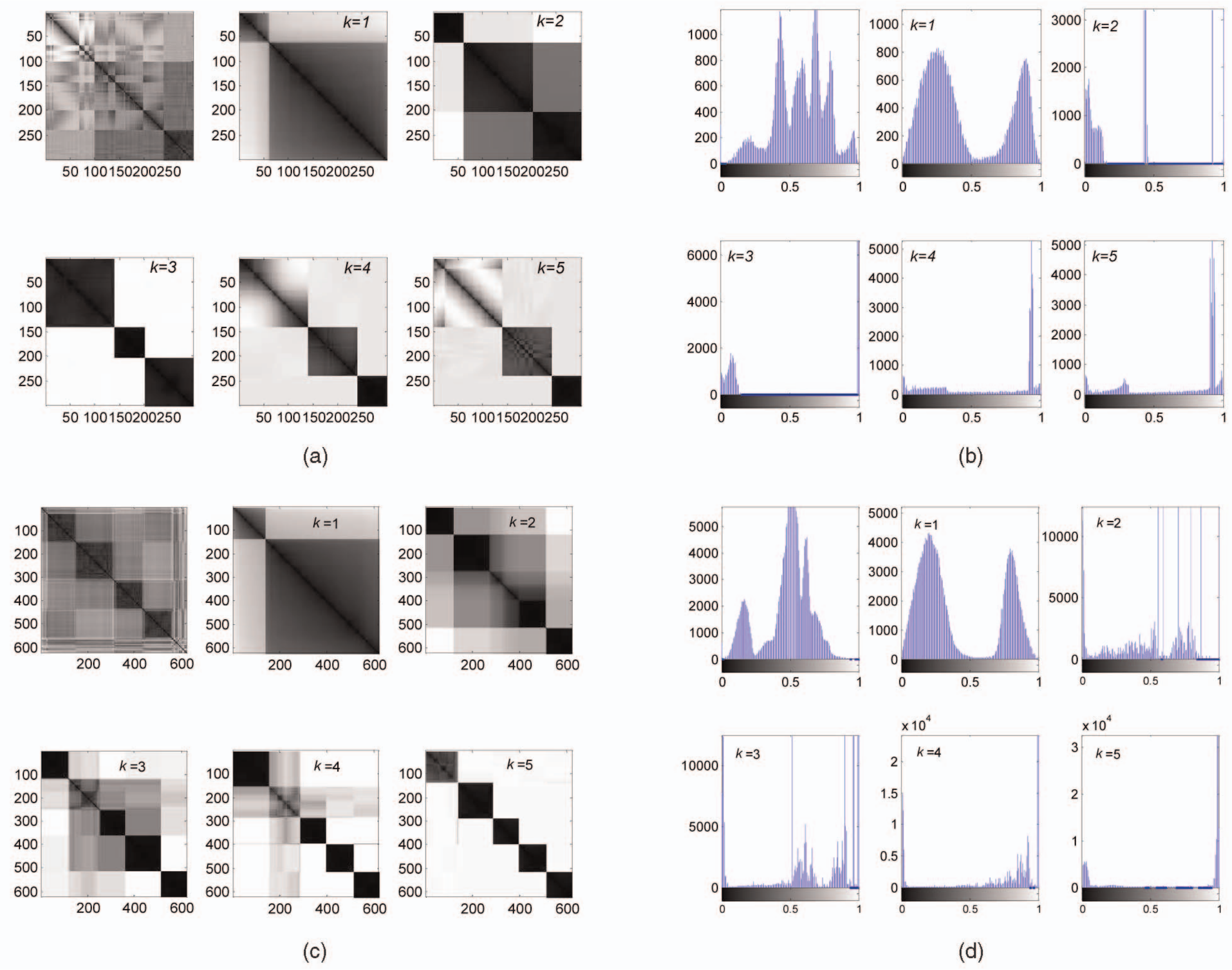


Fig. 1. (a) and (c): Original VAT (*top-left*) and SpecVAT images with different k for data sets S-1 (*top*) and S-4 (*bottom*), and (b) and (d): their corresponding gray-scale histograms. (a) S-1, (b) S-1, (c) S-4, and (d) S-4.

Suppose that the image pixels can be divided into two classes C_1 and C_2 (e.g., corresponding to "within-cluster blocks" and "between-cluster blocks" in the VAT or SpecVAT image) by a threshold at level T. Assume that C_1 denotes pixels with levels $[1, \ldots, T]$ and C_2 denotes pixels with levels $[T+1, \ldots, L]$. Then, the probabilities of class occurrence are, respectively, $\omega_1 = P(C_1) = \sum_{l=1}^{T} p_l$ and $\omega_2 = P(C_2) = \sum_{l=T+1}^{L} p_l$. The class mean levels are

$$\mu_1 = \sum_{l=1}^{T} lP(l|C_1) = \sum_{l=1}^{T} lp_l/\omega_1 = \mu(T)/\omega(T), \tag{1}$$

$$\mu_2 = \sum_{l=T+1}^{L} lP(l|C_2) = \sum_{l=T+1}^{L} lp_l/\omega_2 = \frac{\mu_L - \mu(T)}{1 - \omega(T)}, \tag{2}$$

where $\omega(T) = \sum_{l=1}^{T} p_l$ and $\mu(T) = \sum_{l=1}^{T} lp_l$ are the zeroth- and the first-order cumulative moments of the histogram up to the Tth level, respectively, and $\mu_L = \sum_{l=1}^{L} lp_l$ is the total mean level of the original image. Note that $\omega_1\mu_1 + \omega_2\mu_2 = \mu_L$ and $\omega_1 + \omega_2 = 1$. The class variances are then given by

$$\sigma_1^2 = \sum_{l=1}^{T} (l - \mu_1)^2 P(l|C_1) = \sum_{l=1}^{T} (l - \mu_1)^2 p_l/\omega_1, \tag{3}$$

$$\sigma_2^2 = \sum_{l=T+1}^{L} (l - \mu_2)^2 P(l|C_2) = \sum_{l=T+1}^{L} (l - \mu_2)^2 p_l/\omega_2. \tag{4}$$

Based on the discriminant criteria [14], Otsu used the following measures for evaluating the class separability [17]:

$$\eta = \sigma_B^2/\sigma_W^2, \quad \gamma = \sigma_T^2/\sigma_W^2, \quad \xi = \sigma_B^2/\sigma_T^2, \tag{5}$$

where

$$\sigma_W^2 = \omega_1\sigma_1^2 + \omega_2\sigma_2^2, \tag{6}$$

$$\sigma_B^2 = \omega_1\omega_2(\mu_2 - \mu_1)^2, \tag{7}$$

$$\sigma_T^2 = \sum_{l=1}^{L} (l - \mu_L)^2 p_l = \sigma_W^2 + \sigma_B^2, \tag{8}$$

are the within-class variance, the between-class variance, and the total variance of levels, respectively. A good choice of threshold is to solve the optimization problem by maximizing ξ, γ, or η. σ_W^2 and σ_B^2 are functions of T, but

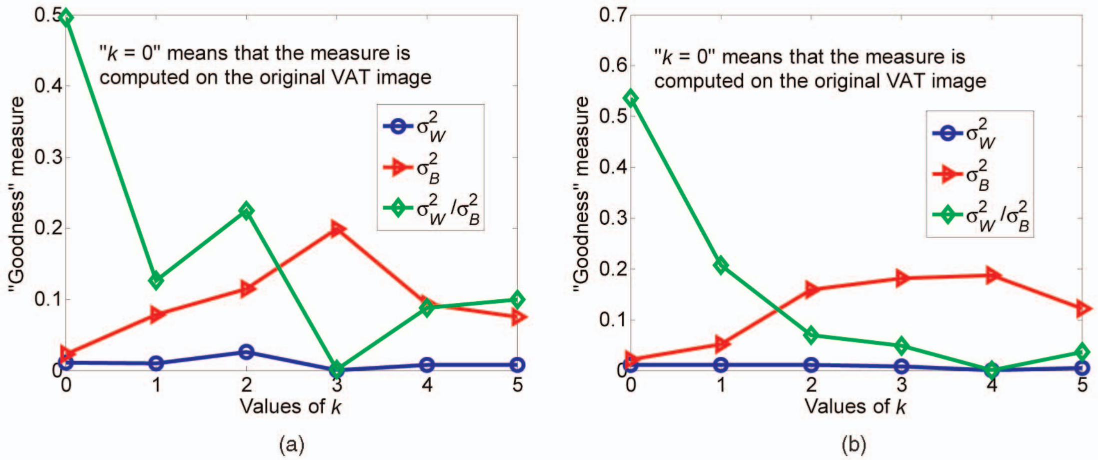


Fig. 2. Examples of "goodness" measures of original VAT and SpecVAT images with different k for data sets (a) S-1 and (b) S-5.

σ_T^2 is independent of T. In particular, σ_W^2 is based on the second-order statistics (class variances), while σ_B^2 is based on the first-order statistics (class means). Thus, ξ is the simplest measure to obtain an optimal threshold T^*.

Naturally, the maximum value $\xi(T^*)$ can be used as a measure to evaluate the separability of classes [17]. Fortunately, such a measure is semantically consistent with our knowledge of the field in question. For each SpecVAT image with respect to different k (e.g., $k = 1$ to k_{max}), we can find an optimal threshold T^* that maximizes ξ (or, equivalently, maximizes σ_B^2). We denote the value of $\sigma_B^2(T^*, k)$ as the "goodness" measure of the image with respect to both clarity and blockyness. Accordingly, we select the best SpecVAT image as the one with the maximum goodness value, and determine the number of clusters as

$$c = \arg \max_{k\in\{1,\ldots,k_{max}\}} \sigma_B^2(T^*, k). \qquad (9)$$

Fig. 2 gives two examples of "goodness" values on the data sets S-1 and S-5, as shown in Fig. 3, thus depicting the effectiveness of such a measure in determining a good SpecVAT image, e.g., $k = 3$ for S-1 (three-circle data set) and $k = 4$ for S-5 (four-line data set). Table 2 summarizes the algorithm for automatically assessing the number of clusters (called A-SpecVAT) from a series of SpecVAT images derived from $\boldsymbol{D}$. The computational complexity of this algorithm mainly depends on the computation of grayscale histograms of images and optimal thresholds, which is $O(k_{max}Ln^2)$.

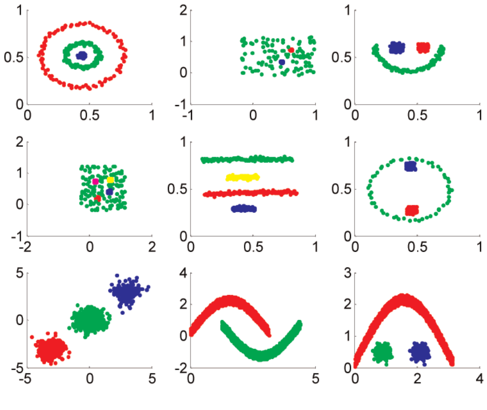


Fig. 3. Scatter plots of synthetic data sets: from left to right and from top to bottom: S-1 to S-9.

4 Visual Data Partitioning

In this section, we further explore the use of spectral VAT for the problem of visual data partitioning. That is, can we automatically extract a crisp c-partition of $\mathcal{O}$ directly from the visual evidence in $\mathrm{I}(\tilde{\boldsymbol{D}}')$? If so, how well does it perform? In general, the c-partitions of a data set $\mathcal{O}$ are sets of $c \cdot n$ values u_{ik} that can be conveniently arrayed as a $c \times n$ matrix $\boldsymbol{U} = [u_{ik}]$. The set of all nondegenerate c-partition matrices for $\mathcal{O}$ is

$$H_{hcn} = \{\boldsymbol{U} \in \mathcal{R}^{c\times n} | 0 \le u_{ik} \le 1,\ \forall i, k\}, \qquad (10)$$

$$\text{with} \quad \sum_{i=1}^{c} u_{i,k} = 1, \forall k \quad \text{and} \quad \sum_{k=1}^{n} u_{ik} > 0, \forall i. \qquad (11)$$

Element u_{ik} of $\boldsymbol{U}$ is the membership of object k in cluster i. In the case of a "crisp" (or hard) partition (not fuzzy or probabilistic), $u_{ik} = 1$ if o_k is labeled i and 0 otherwise.

The important property of $\mathrm{I}(\tilde{\boldsymbol{D}}')$ is that it has, beginning in the upper left corner, dark blocks along its main diagonal. Accordingly, we can constrain our search through H_{hcn} to those partitions that mimic the block structure in $\mathrm{I}(\tilde{\boldsymbol{D}}')$ [48], i.e.,

TABLE 2
Algorithm II: A-SpecVAT

Input: $\boldsymbol{D} = [d_{ij}]$: an $n \times n$ scaled matrix of pairwise dissimilarities
k_{max}: the user-specified maximum of the number of the eigenvectors used in SpecVAT

(1): Perform the SpecVAT algorithm to obtain a series of SpecVAT images $\mathrm{I}(\tilde{\boldsymbol{D}}'_k)$ with $k = 1$ to k_{max}.
(2): Compute an optimal threshold T_k^* that can maximize σ_B^2 for the image $\mathrm{I}(\tilde{\boldsymbol{D}}'_k)$, *i.e.*, $T_k^* = \arg\max_{1\le T<L} \sigma_B^2(T)$.
(3): Obtain the corresponding "goodness" measure for each SpecVAT image $GM(k) = \sigma_B^2(T_k^*)$.
(4): Determine the number of clusters as $c = \arg\max_k GM(k)$.
Output: The number of clusters c

$$H^*_{hcn} = \{\boldsymbol{U} \in H_{hcn}\}, \tag{12}$$

with the properties of $u_{1k} = 1, 1 \le k \le n_1; u_{ik} = 1, n_{i-1} < k \le n_i, 2 \le i \le c$ where n_i is the size of the ith cluster in the data. We call $\boldsymbol{U}$ in H^*_{hcn} an aligned c-partition of $\mathcal{O}$ when its entries form c contiguous blocks of 1s in $\boldsymbol{U}$, ordered to begin from the upper left corner, and proceed down and to the right. Every member of H^*_{hcn} is isomorphic to the unique set of c distinct integers, i.e., the cardinalities of the c clusters in $\boldsymbol{U}$ that satisfy $\{n_i | 1 \le n_i; 1 \le i \le c; \sum_{i=1}^{c} n_i = n\}$, so aligned partitions can be alternatively specified by $\{n_1 : n_2 : \cdots : n_c\}$.

The important characteristics of $\mathrm{I}(\tilde{\boldsymbol{D}}')$ that can be exploited for finding a good candidate partition $\boldsymbol{U}$ are the contrast differences between the dark blocks along the main diagonal and the pixels adjacent to them. Our algorithm aims to generate candidate partitions in H^*_{hcn} by testing their fitness to the clusters suggested by the aligned dark blocks in $\mathrm{I}(\tilde{\boldsymbol{D}}')$. Toward this end, an objective function is defined to implicitly account for block structures. An intuitively appealing measure is the difference of the mean dissimilarity values between apparent clusters (i.e., dissimilarities in nondark blocks off-diagonal) and those within apparent clusters (i.e., dissimilarities in dark blocks along the diagonal) [48]. Let $\boldsymbol{U}$ be a candidate partition in H^*_{hcn}, $\{C_i, 1 \le i \le c\}$ be the crisp c-partition of $\mathcal{O}$ corresponding to $\boldsymbol{U}$, $|C_i| = n_i, \forall i$, and we abbreviate the membership $o_s \in C_i$ as $s \in i$. Mean dissimilarity between dark and nondark regions in $\mathrm{I}(\tilde{\boldsymbol{D}}')$ (i.e., between-cluster distances) E_b and mean dissimilarity within dark regions in $\mathrm{I}(\tilde{\boldsymbol{D}}')$ (i.e., within-cluster distances) E_w are, respectively, represented by

$$E_b = \sum_{i=1}^{c} \left(\sum_{s \in i, t \ni i} \tilde{d}^*_{st} \right) \Bigg/ \sum_{i=1}^{c} n_i (n - n_i), \tag{13}$$

$$E_w = \sum_{i=1}^{c} \left(\sum_{s,t \in i, s \ne t} \tilde{d}^*_{st} \right) \Bigg/ \sum_{i=1}^{c} n_i (n_i - 1). \tag{14}$$

The objective function is defined as

$$E(\boldsymbol{U}, \tilde{\boldsymbol{D}}') = E_b - E_w. \tag{15}$$

A good $\boldsymbol{U}$ should maximize this objective function, i.e.,

$$\boldsymbol{U}^* = \arg \max_{\boldsymbol{U} \in H^*_{hcn}} E(\boldsymbol{U}, \tilde{\boldsymbol{D}}'). \tag{16}$$

We use a Genetic Algorithm (GA) [47] for this optimization problem. As a particular class of evolutionary algorithms, a genetic algorithm is implemented as a computer simulation in which a population of abstract representations (called a genome) of candidate solutions (called individuals) to an optimization problem evolves toward better solutions. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated and multiple individuals are stochastically selected from the current population and modified to form a new population that is then used in the next iteration. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population, or there is no further improvement within a number of generations. Our visual data Partitioning procedure based on the SpecVAT image (called P-SpecVAT) is summarized in Table 3. Several points about this algorithm are noted as follows:

TABLE 3
Algorithm III: P-SpecVAT

Input: $\mathrm{I}(\tilde{\boldsymbol{D}}')$: the SpecVAT image generated from a set of n objects
$\pi()$: the permutation index obtained during VAT re-ordering
c: the number of clusters
b: the size of population

(1): Set the genome of each individual $\boldsymbol{x}_i (i = 1 \sim b)$ as a binary string of length $n - 1$, corresponding to the indices of the first $n - 1$ samples.

(2): Randomly set $c - 1$ elements in each $\boldsymbol{x}_i$ to '1' and others to '0' to create the initial population.

(3): Set the fitness function as taking the input $\boldsymbol{x}_i$, calculating the candidate partition $\boldsymbol{U}$ from $\boldsymbol{x}_i$, and returning the result of Eq. (15).

(4): Apply the GA algorithm until there is no improvement within $g = 10$ generations to find the optimum genome $\boldsymbol{x}^*$.

(5): Transform $\boldsymbol{x}^*$ into cluster partition $\boldsymbol{U}^*$ (which is equivalent to obtaining the sizes of each cluster $\{n_1, \cdots, n_c\}$). The position p_1 of the first '1' in $\boldsymbol{x}^*$ means the first cluster partition is from sample 1 to p_1. The position $p_j (j = 2, \ldots, c - 1)$ of the j-th '1' means the j-th cluster partition is from sample $(p_{j-1} + 1)$ to p_j. The c-th cluster partition is from sample $(p_{c-1} + 1)$ to n.

(6): Retrieve real object indices in each cluster C_i with the permutation index $\pi()$, *i.e.*, $C_1 = \{o_{\pi(1)}, \cdots, o_{\pi(n_1)}\}$, and $C_i = \{o_{\pi(n_{i-1}+1)}, \cdots, o_{\pi(n_{i-1}+n_i)}\}$ for $i = 2, \cdots, c$.

Output: The data partitioning results $\{C_1, \cdots, C_c\}$

- Although we use the GA algorithm in this algorithm, in principle, a number of optimization algorithms might also be adopted, e.g., particle swarm optimization.
- In addition to the SpecVAT image, this visual partitioning procedure can also operate on the VAT image or other reordered dissimilarity images.
- The time complexity of this algorithm mainly depends on the GA algorithm, whose complexity is $\mathrm{O}(bgf_o)$, where b is the population, g is the number of generations, and f_o is the computational complexity of the objective function. In our case, the complexity of the objective function is $O(cn^2)$, so the total algorithm complexity is $\mathrm{O}(bgcn^2)$.

5 Dealing with Large Data Sets

The spectral analysis used in the SpecVAT algorithm relies on the eigendecomposition of an $n \times n$ similarity matrix, which generally takes $\mathrm{O}(n^3)$ time and $\mathrm{O}(n^2)$ space complexity. It is clear that spectral decomposition is intractable for large n, especially in the case of a dense matrix. This limitation makes the spectral VAT method impractical (or computationally infeasible) when handling large data sets. Even though eigendecomposition could be done (e.g., via distributed parallel computation), re-ordering such a large $n \times n$ matrix is also challenging for the VAT algorithm, as well as the inability of displaying such a large image due to the limitation of current screen resolutions. Additional strategies are thus required to scale the SpecVAT algorithm (and other relevant algorithms) to larger data sizes while maintaining "acceptable" cluster quality. Here, we propose a sampling-based extended scheme. To put this scheme in context, we use the following notation. Assume that an $n \times n$ symmetric positive semidefinite matrix, say $\boldsymbol{F}$, can be decomposed as $\boldsymbol{F} = \boldsymbol{U}_F \Sigma_F \boldsymbol{U}_F^T$, where Σ_F are the eigenvalues of $\boldsymbol{F}$ and $\boldsymbol{U}_F$ are the associated eigenvectors. Suppose m columns of $\boldsymbol{F}$ are randomly sampled without replacement. Let $\boldsymbol{A}$ be the $n \times m$ matrix of these sampled columns, and $\boldsymbol{S}$ be the $m \times m$ matrix consisting of the intersection of these m columns with the corresponding m rows. Without loss of generality, we can rearrange the columns and rows of $\boldsymbol{F}$ such that

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{S} & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{C} \end{pmatrix} \quad \text{with} \quad \boldsymbol{A} = \begin{pmatrix} \boldsymbol{S} \\ \boldsymbol{B}^T \end{pmatrix}, \tag{17}$$

where $\boldsymbol{B} \in \mathcal{R}^{m \times (n-m)}$ contains the elements from the sampled objects to the remaining ones, and $\boldsymbol{C} \in \mathcal{R}^{(n-m) \times (n-m)}$ contains the elements between all of the remaining objects. In the case of interest, $m \ll n$, $\boldsymbol{S}$ is very small but $\boldsymbol{C}$ is usually large.

As a technique for finding numerical approximations to eigenfunction problems, the Nyström method has recently been used for fast Gaussian process classification and regression [41], low-rank approximation to kernel matrices [42], large-scale manifold learning [43], and image segmentation [29]. Simply, it uses $\boldsymbol{S}$ and $\boldsymbol{A}$ to approximate $\boldsymbol{F}$ as

$$\boldsymbol{F} \approx \tilde{\boldsymbol{F}} = \boldsymbol{A}\boldsymbol{S}^{+}\boldsymbol{A}^T, \tag{18}$$

where "+" is the pseudoinverse. In fact, the Nyström method implicitly approximates $\boldsymbol{C}$ by $\boldsymbol{B}\boldsymbol{S}^{+}\boldsymbol{B}^T$, and the resulting approximate eigenvalues and eigenvectors of $\boldsymbol{F}$ are

$$\tilde{\Sigma}_F = \left(\frac{n}{m}\right)\Sigma_S \quad \text{and} \quad \tilde{\boldsymbol{U}}_F = \sqrt{\frac{m}{n}}\boldsymbol{A}\boldsymbol{U}_S\Sigma_S^{+}, \tag{19}$$

where $\boldsymbol{S} = \boldsymbol{U}_S \Sigma_S \boldsymbol{U}_S^{+}$ [41]. In a more explicit form, the approximated eigenvectors can be written as

$$\tilde{\boldsymbol{U}}_F = \begin{pmatrix} \boldsymbol{U}_S \\ \boldsymbol{B}^T\boldsymbol{U}_S\Sigma_S^{+} \end{pmatrix}. \tag{20}$$

It can be seen that only $\boldsymbol{A}$ (or $\boldsymbol{S}$ and $\boldsymbol{B}$) is needed to compute the approximated eigenvectors of $\boldsymbol{F}$. In this procedure, eigendecomposition on the small sample matrix $\boldsymbol{S} \in \mathcal{R}^{m \times m}$ is practical, and multiplication with the matrix $\boldsymbol{B}$ (i.e., $\boldsymbol{B}^T\boldsymbol{U}_S\Sigma_S^{+}$) is also feasible.

To perform visual cluster analysis on a large data set, we propose to apply the A-SpecVAT and P-SpecVAT algorithms to a representative sample set, and then extend the sample clustering result to obtain (approximate) clusters for the remaining objects, like [32], [28], and [30]. That is, after obtaining the approximate eigenvectors of $\boldsymbol{F}$ using the Nyström method, we treat each row of $\tilde{\boldsymbol{U}}_F$ as a new "instance" in the spectral space (implicitly corresponding to an original object in $\mathcal{O}$). We choose the new instances corresponding to the chosen samples to obtain the sample SpecVAT images, which are used to determine the number of clusters c via A-SpecVAT, and then apply P-SpecVAT on the best sample SpecVAT image to obtain the partitioning of the sample data, i.e., obtaining the class labels of the sampled objects $\{o_1^s, o_2^s, \ldots, o_m^s\}$, say $\{l_1, l_2, \ldots, l_m\}$, where $l_i \in \{1, \ldots, c\}$.

TABLE 4
Algorithm IV: E-SpecVAT

```
Input: $\boldsymbol{D} = [d_{ij}]$: an $n \times n$ scaled pairwise dissimilarity matrix
       $k_{max}$: the user-specified maximum number of eigenvectors used
(1): Compute a local scale $\sigma_i$ for each object $o_i$ as $\sigma_i = d(o_i, o_K) = d_{iK}$, where $o_K$ is the $K$-th nearest neighbor of $o_i$.
(2): Choose the $m$ indices from $\{1, 2, \cdots, n\}$ randomly to form the sample index set $I_s$, and the set of the remaining object indices $I_r$, which are respectively used to get sub-matrices $\boldsymbol{D}_S$ and $\boldsymbol{D}_B$ from $\boldsymbol{D}$.
(3): Construct the matrices $\boldsymbol{S} \in \mathcal{R}^{m \times m}$ from $\boldsymbol{D_S}$ and $\boldsymbol{B} \in \mathcal{R}^{m \times (n-m)}$ from $\boldsymbol{D}_B$ using the weighted Gaussian function $\exp(-d_{ij}d_{ji}/(\sigma_i\sigma_j))$.
(4): Perform eigendecomposition of $\boldsymbol{S}$, and compute the approximate eigenvectors $\tilde{\boldsymbol{U}}_F$ using Eq (20).
(5): For $k = 1 \sim k_{max}$, choose the first $k$ columns of $\tilde{\boldsymbol{U}}_F$ to form $\boldsymbol{V}_k \in \mathcal{R}^{n \times k}$ and normalize the rows of $\boldsymbol{V}_k$ to unit Euclidean norm to generate $\boldsymbol{V'_k}$. Treat each row of $\boldsymbol{V'_k}$ as a new instance to compute a new pairwise dissimilarity matrix $\boldsymbol{D'_s} \in \mathcal{R}^{m \times m}$ between the sample instances to obtain sample SpecVAT images $\mathrm{I}(\tilde{\boldsymbol{D}}'_s)$, based on which the A-SpecVAT method is used to determine $c$ by finding a best $k^*$.
(6): For the sample SpecVAT image $\mathrm{I}(\tilde{\boldsymbol{D}}'_{k^*})$, apply the P-SpecVAT algorithm to partition the sample objects indexed by $I_s$ into $c$ groups.
(7): Perform out-of-sample extension using $\boldsymbol{V'_{k^*}}$ to obtain the cluster labels of the remaining objects indexed by $I_r$.
Output: The number of clusters $c$ and the data partitioning results
```

Next, we address the problem of out-of-sample extension, i.e., how to assign each of the remaining $n - m$ objects to one of the c previously determined groups. Out-of-sample extension can be treated as a prediction problem in the embedding space. For each o_j^e $(j = m+1, m+2, \ldots, n)$ in $\boldsymbol{B}$, we compute the distance between its corresponding instance and the instances of the sample objects in the embedding space, and then assign the object o_j^e to the class label with the maximum votes from its k nearest neighbors (kNN) w.r.t labeled samples, i.e., obtaining the cluster labels of the remaining objects $\{o_{m+1}^e, o_{m+2}^e, \ldots, o_n^e\}$, say $\{l_{m+1}, l_{m+2}, \ldots, l_n\}$. Table 4 summarizes our Extended SpecVAT algorithm for dealing with large data sets (called E-SpecVAT). Several points about E-SpecVAT are noted as follows:

- This method does not require the entire data matrix D to be loaded into memory at once, since this may not be feasible for large data sets. When the matrix D is very large, we may incrementally load it to perform Step (1).
- In addition to random sampling, other active sampling techniques may be used, e.g., selective sampling [32] and K-means sampling [42] (when object data is available).
- The eigenvectors generated from the Nyström approximation are not exactly orthogonal because they are extrapolated from the eigenvectors of S. We could adopt the methods described in [29] to compute the approximated orthogonalized eigenvectors in Step 4.
- The time complexity of this algorithm mainly depends on the eigendecomposition approximation using the Nyström method, sample partitioning, and out-of-sample extension. To calculate approximations to the top k^* eigenvectors and eigenvalues of F, the runtime of this process is $\mathrm{O}(m^3 + k^*mn)$, in

TABLE 5
Summary of the Data Sets Used and Results of Estimating c

Data	c_p	# attri.	n	c_{ov}^m	c_{sv}^m	c_{sv}^a
S-1	3	2	299	≥ 1	3	3
S-2	3	2	303	≥ 2	3	3
S-3	3	2	266	≥ 2	3	3
S-4	5	2	622	≥ 4	5	5
S-5	4	2	512	≥ 2	4	4
S-6	3	2	238	≥ 2	3	3
S-7	3	2	1000	3	3	3
S-8	2	2	2000	≥ 1	2	2
S-9	3	2	2000	≥ 2	3	3
R-1	2	9	683	≥ 3	2	2
R-2	3	1200	1755	3 or 4	3	3
R-3	3	-	194	≥ 3	3	3
R-4	3	4	150	≥ 2	$\underline{2}$	$\underline{2}$
R-5	2	16	435	≥ 2	2	2
R-6	3	13	178	≥ 3	3	3
R-7	10	-	100	≥ 9	10	10
R-8	10	649	2000	≥ 7	$\underline{9}$	$\underline{9}$
R-9	6	9	214	≥ 4	6	6

which $\mathrm{O}(m^3)$ is required for eigendecomposition on S and $\mathrm{O}(k^*mn)$ for multiplication with B. As analyzed before, visual data partitioning on the sample SpecVAT image has the complexity of $\mathrm{O}(bgk^*m^2)$. The time complexity for the extension of the $n-m$ remaining objects using kNN is $\mathrm{O}((n-m)kk^*m)$. Together with $\mathrm{O}(Kn^2)$ for computing local-scale parameters, the total algorithm complexity is

$$\mathrm{O}(m^3 + (bg-k)k^*m^2 + (k+1)k^*mn + Kn^2).$$

To make the algorithm complexity nearly linear to the number of objects n for truly large data sets, it may be necessary to use a global scale σ instead of the local-scale parameters σ_i (and thus, removing $\mathrm{O}(Kn^2)$ time complexity).

6 Experimental Results

In order to evaluate our algorithms, we have carried out a number of experiments on artificially generated data sets, as well as real-world data sets (summarized in Table 5). Unless otherwise mentioned, in the following experiments, the (euclidean) distance matrix D is computed in the original attribute space (if the object vectorial representation is available). All experiments were implemented in a Matlab 7.2 environment on a PC with an Intel 2.4 GHz CPU and 2 GB memory running Windows XP.

6.1 Evaluation Data Sets

Nine synthetic data sets with different data structures (i.e., different numbers of clusters and different data distributions) are used in our experiments. The scatter plots of these synthetic data sets are shown in Fig. 3, in which each color represents a visually meaningful group. The first 6 data sets are taken from [26], and the last 3 are generated by ourselves. Except for S-7, which is a mixture of 3 Gaussian shapes, all other data sets involve more irregular data structures, in which an obvious cluster centroid for each group is not necessarily available. As can be seen, some of these data sets include different scales between clusters, or some clusters hidden in a cluttered background.

Nine real-world data sets were also considered to evaluate our algorithms, six of which are from the UCI Machine Learning Repository, i.e., R-1, R-4, R-5, R-8 and R-9. In short, R-1 (*breast-cancer*) database includes 683 instances, each of which has 9 attributes and belongs to one of 2 classes. R-2 (*face*) data set was first used in [4], which is a subset of the Yale-B data set [31], including 1,755 images of 3 different individuals. R-3 (*genetic*) data set [18] is a 194×194 matrix consisting of pairwise dissimilarities from a set of 194 human gene products that were clustered into 3 protein families. R-4 (*iris*) data set contains 3 physical classes, 50 instances each, where each class refers to a type of iris plant with 4 attributes. R-5 (*voting*) data set consists of 435 US House of Representatives members' votes on 16 key votes. R-6 (*wine*) data set contains 178 wine instances derived from 3 different cultivars. R-7 (*action*) data set [50] is a 100×100 matrix consisting of pairwise dissimilarities derived from 100 human action clips. R-8 (*multiple-features*) data set consists of binary image features of 10 handwritten numerals ('0' ~ '9'), 200 patterns per class. These digits are represented as a 649-dimensional vector in terms of 6 feature sets. R-9 (*glass*) data set includes 214 samples from 6 types of glass defined in terms of their oxide content.

6.2 Determining the Number of Clusters

For each data set (except for R-3 and R-7, which are already in dissimilarity matrix form), we computed a pairwise dissimilarity matrix D in the original attribute space. The VAT images are shown in Fig. 4a for the synthetic data sets and Fig. 5a for the real data sets. It can be seen that the cluster structure of the data in these VAT images is not necessarily clearly highlighted. Accordingly, viewers have difficulties in giving a sound result about the number of clusters in these data sets, and different viewers may deduce different estimates of c. Next, we applied the SpecVAT algorithm to each of the data sets, and showed the SpecVAT images in Fig. 4b for the synthetic data sets and Fig. 5b for the real data sets. In contrast to the original VAT images, the SpecVAT images generally have a clearer display of the block structure, and thus better highlight the hidden cluster structure. Table 5 summarizes the number of clusters determined by manual inspection from the original VAT image (c_{ov}^m), manual inspection from a series of SpecVAT images (c_{sv}^m) and automatic estimation from a series of SpecVAT images (c_{sv}^a). For the iris data set, our method gives $c = 2$. This is probably due to the well-recognized fact that in this data set, one class is linearly separable from the other two classes, while the latter two are not linearly separable from each other. For the multiple-features data set, our method gives $c = 9$, another underestimate. This is probably due to the same fact as the iris data set that two classes are highly similar to each other. The results of cluster number estimation from the SpecVAT images for the other 16 data sets are correct in terms of the number of real physical classes c_p, whether it was estimated automatically by our A-SpecVAT algorithm or by manual inspection. The results again highlight the benefits of converting D to D' by graph embedding to obtain a more accurate estimate of c.

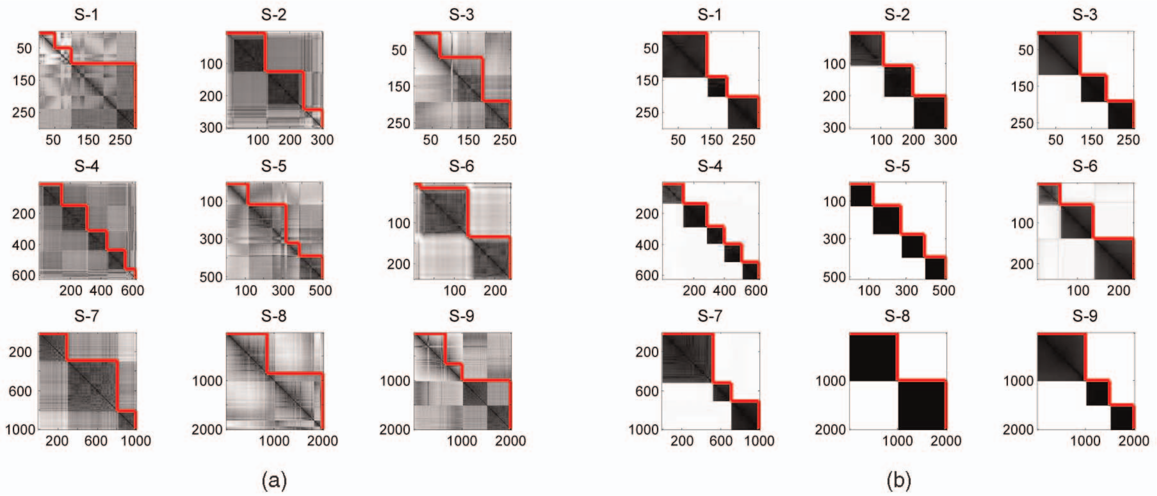


Fig. 4. (a) Original VAT images of nine synthetic data sets with visual clustering results imposed by dark gray lines (see electronic version for color figures), and (b) the corresponding best SpecVAT images with visual clustering results.

6.3 Visual Data Partitioning and Algorithm Comparison

We evaluate our visual partitioning algorithm's performance by comparing the cluster labels of the objects given by our algorithm with the ground truth labels (available for these 18 data sets). An accuracy (AC) metric has been widely used for clustering performance evaluation [5], [16], [25]. Suppose that z_i^c is the clustering label of an object o_i and z_i^g is the corresponding ground truth label, then AC is defined as $\max_{map} \sum_{i=1}^{n} \delta(z_i^g, map(z_i^c))/n$, where n is the total number of objects in the data, $\delta(z_1, z_2)$ is the delta function that equals 1 if and only if $z_1 = z_2$ and 0 otherwise, and map is the mapping function that permutes clustering labels to match equivalent labels given by the ground truth. The Kuhn-Munkres algorithm is usually used to obtain the best mapping [12]. See Figs. 4b and Fig. 5b for visualization of our data partitioning results on synthetic and real-world data sets. The clustering accuracy of our visual partitioning algorithm on the original VAT image (V_{ov}) and the SpecVAT image (V_{sv}) is summarized in Table 6, from which we can see that our algorithm obtains satisfactory partitioning results. In addition, it is clear that V_{sv} performs better than V_{ov}.

We also implemented several typical clustering algorithms for comparison. These algorithms are, respectively, K-means (K_m), Ward's hierarchal clustering (L_w) [49], standard spectral clustering with a global-scale parameter [15] (S_σ), and spectral clustering with local-scale parameters [26] (S_{σ_i}). The clustering accuracies of these algorithms on these 18 synthetic and real data sets are listed in Table 6, from which it can be seen that the overall accuracy of our partitioning clustering algorithm on the SpecVAT image is better than that of K-means, Ward's algorithm, and standard spectral clustering with a global-scale parameter, and is comparable to that of spectral clustering with local scaling. Moreover, our visual methods give intuitive observations on the number of clusters, cluster structure, and partition results from the images, as well as eliminating the randomly initialized K-means clustering stage (as usually used in spectral clustering).

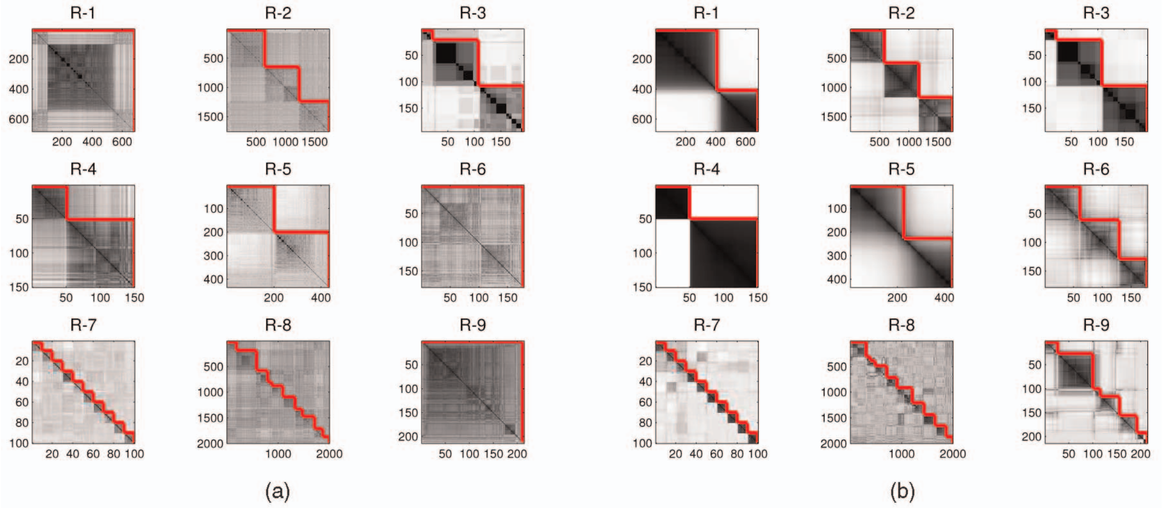


Fig. 5. (a) Original VAT images of 9 real-world data sets with visual clustering results imposed by dark gray lines (see electronic version for color figures), and (b) the corresponding best SpecVAT images with visual clustering results.

TABLE 6
Comparison of Clustering Algorithm Accuracy (Percent)

Data	c	K_m	L_w	S_σ	S_{σ_i}	V_{sv}	V_{ov}
S-1	3	45.64	48.83	100.0	100.0	100.0	63.21
S-2	3	73.70	71.62	84.49	100.0	100.0	84.49
S-3	3	74.06	75.56	100.0	100.0	100.0	78.95
S-4	5	79.45	82.80	88.91	100.0	100.0	88.59
S-5	4	69.36	70.70	100.0	100.0	100.0	84.18
S-6	3	82.59	83.19	96.64	100.0	100.0	82.77
S-7	3	97.17	100.0	100.0	100.0	100.0	100.0
S-8	2	88.30	71.70	100.0	100.0	100.0	92.15
S-9	3	76.05	77.95	100.0	100.0	100.0	57.50
R-1	2	96.05	96.63	96.78	96.78	94.88	65.15
R-2	3	94.80	100.0	99.83	99.83	99.66	96.18
R-3	3	-	-	100.0	100.0	100.0	100.0
R-4	2	98.00	100.0	100.0	100.0	100.0	99.33
R-4	3	81.45	89.33	90.67	93.33	92.67	67.33
R-5	2	88.14	91.72	87.13	88.05	90.80	83.45
R-6	3	96.30	92.70	97.75	97.75	98.31	39.33
R-7	10	-	-	100.0	100.0	100.0	100.0
R-8	10	76.19	95.55	82.15	81.60	82.25	68.95
R-9	6	41.16	42.06	46.00	46.26	46.26	37.85

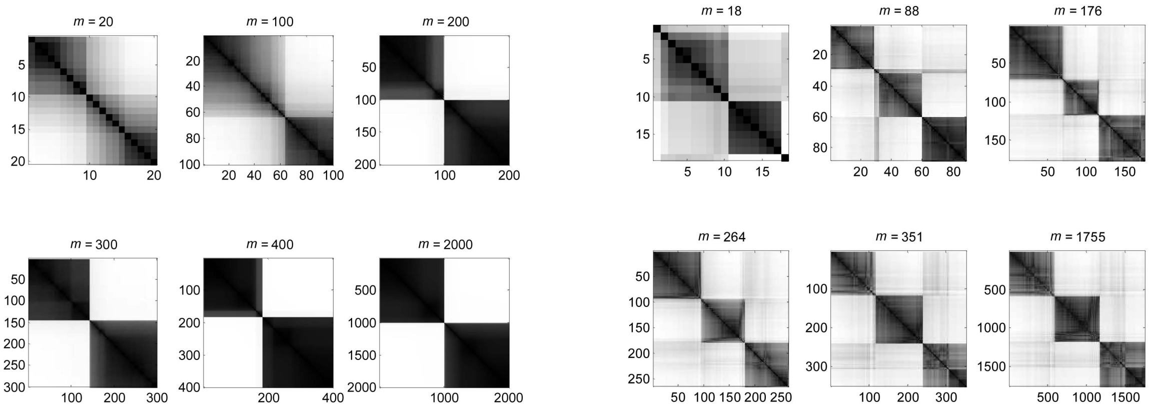
TABLE 7
Summary of Clustering Accuracy Using E-SpecVAT (Percent)

SR	0.01	0.05	0.10	0.15	0.20	1
S-8	97.42	100.0	100.0	100.0	100.0	100.0
R-2	97.12	99.78	99.79	99.82	99.82	99.66
R-8	67.82	82.06	81.63	81.86	80.36	82.25

6.4 Approximate SpecVAT and Clustering

In order to test the performance of the E-SpecVAT algorithm, we first selected three data sets from those data sets used in the previous experiments, i.e., S-8, R-2, and R-7. These three data sets are relatively large among synthetic and real-world data sets. In addition, the sizes of these three data sets are such that we can apply both E-SpecVAT with sampling and SpecVAT without sampling to examine the effect of sampling on the resulting SpecVAT images as well as the clustering accuracy. We tried multiple sampling rates (SR, i.e., m/n) on the three data sets. For each sampling rate, 50 trials were made, and then we computed the average clustering accuracy. Fig. 6 shows examples of the sample SpecVAT images on the data sets S-8 and R-2 with different sample sizes (only one run), from which it can be seen that when the sample size is sufficient, the sample SpecVAT images can produce a good approximation to the SpecVAT image using the full data. This also naturally leads to satisfactory (approximate) partitioning results, as summarized in Table 7. More interestingly, several results using the sampling-based method are (slightly) better than that of the method using all of the data. Such a phenomenon also appears in [28]. This is probably because a small portion of representative samples may be enough to effectively reveal the complete structure of the whole data, while the introduction of more (possibly noisy) data will likely bring negative influences on the algorithms. However, we also noticed that such "interesting" results often appear when the sample size is sufficient (refer to the sample SpecVAT images shown in Fig. 6), and such sampling-based accuracies are somewhat better but very close to those using the full data (see the results on R-2 and R-8 in Table 7).

Next, we applied our E-SpecVAT algorithm to the problem of high-resolution image segmentation (where it is generally infeasible to use the full data directly due to computational and storage complexities). Different features (such as intensity, color, texture, and proximity) can be used to compute the similarities between image pixels, e.g., locally-windowed color and texture histograms used in [29]. We used only the "intensity" feature to compute the dissimilarities between image pixels since our main concern is to demonstrate the feasibility of our approximate algorithm in the context of image segmentation, rather than focus on image segmentation itself. In our experiments, we used five 481×321 images (Fig. 7 (*left column*)) taken from the Berkeley image segmentation database. Running A-SpecVAT and/or P-SpecVAT algorithms directly on the whole image (which contains $n = 481 \times 321 = 154{,}401$ pixels, representing relatively larger data sets) would be simply impossible in the Matlab environment. For these images, the number of sampled pixels was empirically chosen to be 300 (around 0.2 percent of the number of total pixels), considering that there are far fewer coherent groups in a scene than pixels (i.e., $c \ll n$). We cannot measure the clustering error in this case because of the lack of ground truth. So, the best



(a) (b)

Fig. 6. Examples of sample SpecVAT images with different sample sizes for data sets S-8 and R-2 (a) Synthetic data S-8 (b) Real data R-2.

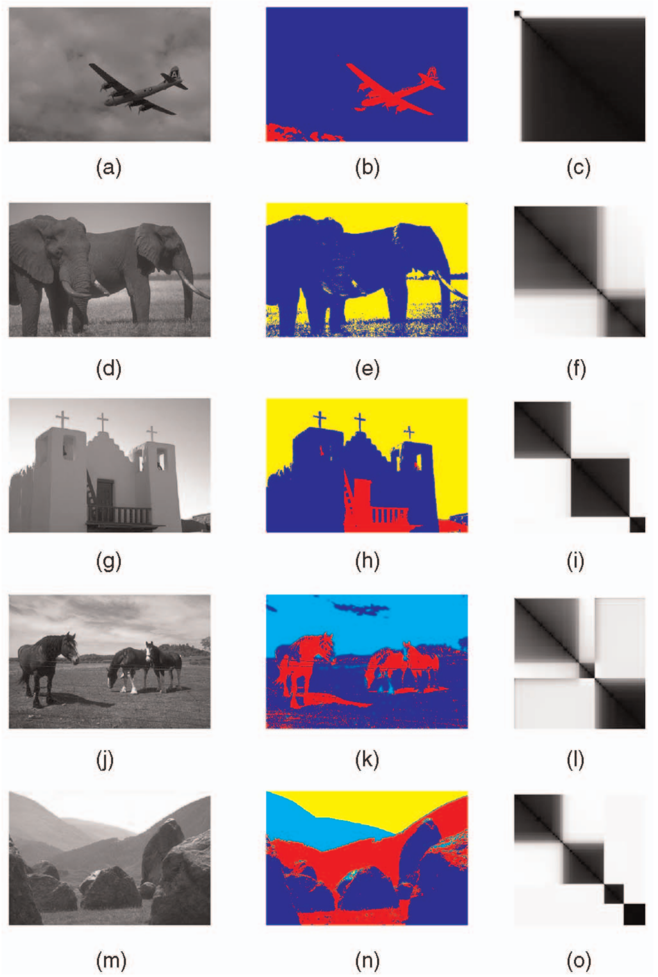


Fig. 7. Original images (*left*), image segmentation results using E-SpecVAT (*middle*, see color figures in electronic version for better visualization), and the corresponding sample SpecVAT images (*right*).

we can do for evaluation here is to resort to visual inspection of the image segmentation results. Fig. 7 (*middle column*) shows the segmentation results on these five images, in which pixels with the same color represent one group. We automatically set $c = 2, 3$, or 4 for these five images according to the number of visually meaningful components suggested in the sample SpecVAT images (Fig. 7 (*right column*)). In these cases, our E-SpecVAT algorithm partitioned the images into meaningful components. This demonstrates again that our algorithm is effective on these larger image data sets.

Finally, we consider the application of our E-SpecVAT method to a very large data set used in [32], which is a set of $n = 3{,}000{,}000$ 2D data points drawn from a mixture of $c = 5$ normal distributions. Five clusters in the data set are visually apparent, but there is a high level of mixing between outliers from components in the mixture (see Fig. 8a for its scatter plot). Computing squared euclidean distances between pairs of vectors yields a matrix D with $n \times (n-1)/2 = \mathrm{O}(10^{12})$ dissimilarities for the data set. We were not able to calculate, load, and process a full-distance data matrix of this size. Here, we used an approximate "lookup table" mode by just storing the object data set, accessing only the vectors needed to make a particular distance computation, and releasing the memory used by these vectors immediately after to avoid exhausting memory. To avoid exhausting memory, the processing was broken up by calling the extension routine multiple times (depending on the chunk size), and each chunk was used to extend the partition for loaded objects. We set the number of clusters $c = 5$ for the data set like [32] and tried two different sample sizes, i.e., $m = 600$ and $m = 2{,}500$. The sample SpecVAT image with respect to $m = 2{,}500$ is shown in Fig. 8b, which also suggested five clusters in the data. Since running once on this data set took quite a long time, we just reported the clustering error rates with respect to only one run per case, i.e., 0.0061 for $m = 600$ and 0.0052 for $m = 2500$. It can be seen that both measures of errors are very small. This was certainly not an easy clustering problem, in terms of how well separated the clusters actually are. The point here was to demonstrate the feasibility property of our E-SpecVAT method on truly large data sets, and this example demonstrated this.

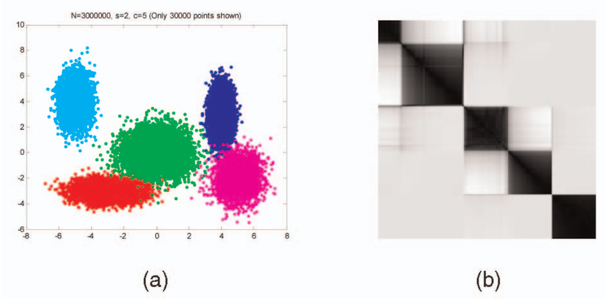


Fig. 8. (a) The scatter plot of the data set taken from [32] and (b) its sample SpecVAT image with $m = 2{,}500$.

7 Discussion and Conclusion

This paper has presented an enhanced visual approach toward automatically determining the number of clusters and partitioning data in either object or pairwise relational form. In order to better reveal the hidden cluster structure, especially for complex-shaped data sets, the VAT algorithm has been improved by using spectral analysis of the proximity matrix of the data. Based on spectral VAT, a "goodness" measure of SpecVAT images has been proposed for automatically determining the number of clusters. Also, we have derived a visual clustering algorithm based on SpecVAT images and its unique block-structured property. We have also proposed an extended strategy to scale the SpecVAT algorithm to larger data sets. A series of primary and comparative experiments on synthetic and real-world data sets have demonstrated that our algorithms perform well in terms of both visual cluster tendency assessment and data partitioning.

There are strong relations between the SpecVAT algorithm (and thus, P-SpecVAT) and other works: both the SpecVAT algorithm and the spectral clustering algorithm described in [15] use spectral decomposition of the normalized Laplacian matrix that is essentially the graph embedding procedure of [1]. A prominent property of the graph embedding framework is the complete preservation of the cluster structure in the embedding space. For new representations in the embedding space, spectral clustering in [15]

TABLE 8
The VAT Algorithm

```
Input: An $n \times n$ scaled matrix of pairwise dissimilarities $\boldsymbol{D} = [d_{ij}]$, with $1 \ge d_{ij} \ge 0; d_{ij} = d_{ji}; d_{ii} = 0$, for $1 \le i,j \le n$
(1): Set $I = \emptyset$, $J = \{1,2,\cdots,n\}$ and $\pi = (0,0,\cdots,0)$.
    Select $(i,j) \in \arg_{p\in J,q\in J} \max\{d_{pq}\}$.
    Set $\pi(1) = i$, $I \leftarrow \{i\}$ and $J \leftarrow J - \{i\}$.
(2): Repeat for $t = 2,3,\cdots,n$
    Select $(i,j) \in \arg_{p\in I,q\in J} \min\{d_{pq}\}$.
    Set $\pi(t) = j$, update $I \leftarrow I \cup \{j\}$ and $J \leftarrow J - \{j\}$.
(3): Form the reordered matrix $\tilde{\boldsymbol{D}} = [\tilde{d_{ij}}] = [d_{\pi(i)\pi(j)}]$, for $1 \le i,j \le n$.
Output: A scaled gray-scale image $\mathrm{I}(\tilde{\boldsymbol{D}})$, in which $\max\{\tilde{d_{ij}}\}$ corresponds to white and $\min\{\tilde{d_{ij}}\}$ to black
```

uses K-means to cluster them; while for our visual clustering algorithm, we first convert them to a new reordered dissimilarity image, and then use the GA to partition its block structures. A local scaling scheme is suggested in [26] to replace the global scale σ in [15], leading to better clustering, especially when the data includes multiple scales or when the clusters are placed within a cluttered background. These connections naturally suggest that our P-SpecVAT algorithm can perform competitively with spectral clustering [15], [26]. The slight difference in accuracy between P-SpecVAT and the algorithm of [26] could be due to different objective functions and optimization strategies in the partitioning stage.

Our algorithms will probably reach their useful limit when the image formed by any reordering of D is not from a well-structured dissimilarity matrix. While our A-SpecVAT algorithm may return a slightly overestimated or underestimated value of c, it provides an initial estimate, thus avoiding the need to run a clustering algorithm multiple times over a wide range of c values in an attempt to find valid clusters. In this way, our method compares favorably to postclustering validation methods in computational efficiency. Note that our method does not eliminate the need for cluster validity (i.e., the third problem in cluster analysis). Cluster validation depends greatly on the choice of the validity criteria and the clustering algorithms. There have been countless studies of direct and indirect validity indices for crisp, fuzzy, and probabilistic clustering methods. Unfortunately, validity functionals have little chance of being generally useful for identifying the best clustering solution, even within a restricted class of models. Studies on visual data partitioning and validation remain limited. The existing index-based validation methods are seemingly unsuitable to validate our visual algorithms. How to find a direct visual validation method will be one of important issues in our future work.

Appendix

The VAT Algorithm

The VAT algorithm [2] works on a pairwise dissimilarity matrix. Let $\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$ denote n objects in the data and D a pairwise matrix of dissimilarities between objects, each element of which $d_{ij} = d(o_i, o_j)$ is the dissimilarity between objects o_i and o_j, and generally, satisfies $1 \ge d_{ij} \ge 0; d_{ij} = d_{ji}; d_{ii} = 0$, for $1 \le i,j \le n$. The VAT algorithm displays a reordered dissimilarity matrix of D as a gray-scale image. Let $\pi()$ be a permutation of $\{1, 2, \ldots, n\}$ such that $\pi(i)$ is the new index for o_i. The reordered list is thus $\{o_{\pi(1)}, \ldots, o_{\pi(n)}\}$. Let P be the permutation matrix with $p_{ij} = 1$ if $j = \pi(i)$ and 0 otherwise, then the matrix $\tilde{D}$ for the reordered list is a similarity transform of D by P, i.e., $\tilde{D} = P^T DP$. The reordering idea is to find P so that $\tilde{D}$ is as close to a block diagonal form as possible. The VAT algorithm reorders the row and columns of D with a modified version of Prim's minimal spanning tree algorithm [46]. If an object is a member of a cluster, then it should be part of a submatrix with low dissimilarity values (corresponding to within-cluster distances), which appears as one of the dark blocks along the diagonal of the VAT image $\mathrm{I}(\tilde{D})$, each of which corresponds to one cluster. We summarize the VAT algorithm in Table 8.

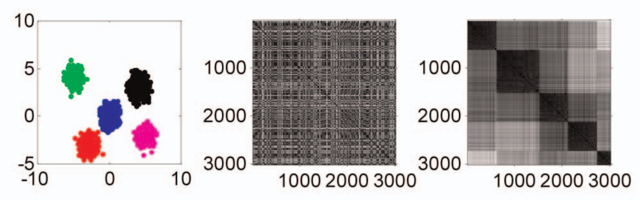


Fig. 9. VAT: (*left*) scatter plot of a 2D data set, (*middle*) unordered image $\mathrm{I}(D)$, and (*right*) reordered VAT image $\mathrm{I}(\tilde{D})$

An example of VAT is shown in Fig. 9. Fig. 9a is a scatter plot of $n = 3000$ points in $\mathcal{R}^2$, which is generated from a mixture of $c = 5$ bivariate normal distributions. These data points were converted to a 3000×3000 dissimilarity matrix D by computing the euclidean distance between each pair of points. The five visually apparent clusters in Fig. 9a are reflected by the five distinct dark blocks along the main diagonal in Fig. 9b, which is the VAT image of the data. Given the image of D in the original input order in Fig. 9b, reordering is necessary to reveal the underlying cluster structure of the data.

Acknowledgments

The authors would like to thank Associate Editor Dr. Talia and the anonymous reviewers for their valuable comments and suggestions. This work was supported by the Australian Research Council (ARC) discovery projects (DP0663196 and DP0987421) when L. Wang worked at the University of Melbourne.

References

[1] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Advances in Neural Information Processing Systems,* MIT Press, 2002.

[2] J.C. Bezdek and R.J. Hathaway, "VAT: A Tool for Visual Assessment of (Cluster) Tendency," *Proc. Int'l Joint Conf. Neural Networks,* pp. 2225-2230, 2002.

[3] J.C. Bezdek, R.J. Hathaway, and J. Huband, "Visual Assessment of Clustering Tendency for Rectangular Dissimilarity Matrices," *IEEE Trans. Fuzzy Systems,* vol. 15, no. 5, pp. 890-903, Oct. 2007.

[4] M. Breitenbach and G. Grudic, "Clustering through Ranking on Manifolds," *Proc. Int'l Conf. Machine Learning,* 2005.

[5] D. Cai, X. He, and J. Han, "Document Clustering Using Locality Preserving Indexing," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 12, pp. 1624-1637, Dec. 2005.

[6] F. Chung, *Spectral Graph Theory,* vol. 92. Am. Math. Soc., 1997.

[7] W.S. Cleveland, *Visualizing Data.* Hobart Press, 1993.

[8] R. Hathaway, J.C. Bezdek, and J. Huband, "Scalable Visual Assessment of Cluster Tendency," *Pattern Recognition,* vol. 39, pp. 1315-1324, 2006.

[9] X. Hu and L. Xu, "A Comparative Study of Several Cluster Number Selection Criteria," *Intelligent Data Engineering and Automated Learning,* pp. 195-202, Springer, 2003.

[10] J. Huband, J.C. Bezdek, and R. Hathaway, "Bigvat: Visual Assessment of Cluster Tendency for Large Data Sets," *Pattern Recognition,* vol. 38, no. 11, pp. 1875-1886, 2005.

[11] R. Ling, "A Computer Generated Aid for Cluster Analysis," *Comm. ACM,* vol. 16, pp. 355-361, 1973.

[12] L. Lovasz and M. Plummer, *Matching Theory.* Elsevier Science publishers B.V. and Akadémiai Kiadó, Budarest, 1986.

[13] U. Maulik and S. Bandyopadhyay, "Performance Evaluation of Some Clustering Algorithms and Validity Indices," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 12, pp. 1650-1654, Dec. 2002.

[14] G.J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition.* John Wiley & Sons, 2005.

[15] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems.* MIT Press, 2002.

[16] H.Z. Ning, W. Xu, Y. Chi, and T.S. Huang, "Incremental Spectral Clustering with Application to Monitoring of Evolving Blog Communities," *Proc. SIAM Int'l Conf. Data Mining,* 2007.

[17] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 9, no. 1, pp. 62-66, Jan. 1979.

[18] N. Pal, J. Keller, M. Popescu, J. Bezdek, J. Mitchell, and J. Huband, "Gene Ontology-Based Knowledge Discovery through Fuzzy Cluster Analysis," *J. Neural, Parallel and Scientific Computing,* vol. 13, pp. 337-361, 2005.

[19] P.J. Rousseeuw, "A Graphical Aid to the Interpretations and Validation of Cluster Analysis," *J. Computational and Applied Math.,* vol. 20, pp. 53-65, 1987.

[20] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.

[21] T. Tran-Luu, "Mathematical Concepts and Novel Heuristic Methods for Data Clustering and Visualization," PhD thesis, Univ. of Maryland, 1996.

[22] U. von Luxburg, "A Tutorial on Spectral Clustering," technical report, Max Planck Inst. for Biological Cybernetics, 2006.

[23] Y. Weiss, "Segmentation Using Eigenvectors: A Unifying View," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 975-982, 1999.

[24] R. Xu and D. Wunsch, II, "Survey of Clustering Algorithms," *IEEE Trans. Neural Networks,* vol. 16, no. 3, pp. 645-678, May 2005.

[25] W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on Non-Negative Matrix Factorization," *Proc. ACM SIGIR,* 2003.

[26] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering," *Advances in Neural Information Processing Systems,* MIT Press, 2004.

[27] J.C. Dunn, "Indices of Partition Fuzziness and the Detection of Clusters in Large Sets," *Fuzzy Automata and Decision Processes,* Elsevier, 1976.

[28] M. Pavan and M. Pelillo, "Efficient Out-of-Sample Extension of Dominant-Set Clusters," *Advances in Neural Information Processing Systems,* MIT Press, 2004.

[29] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral Grouping Using the Nyström Method," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 2, pp. 214-225, Jan. 2004.

[30] Y. Bengio, J. Paiement, P. Vincent, O. Delallean, N. Roux, and M. Ouimet, "Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering," *Advances in Neural Information Processing Systems,* MIT Press, 2004.

[31] A. Georghiades, P. Belhumeur, and D. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 643-660, June 2001.

[32] L. Wang, J. Bezdek, C. Leckie, and R. Kotagiri, "Selective Sampling for Approximate Clustering of Very Large Data Sets," *Int'l J. Intelligent Systems,* vol. 23, no. 3, pp. 313-331, 2008.

[33] S. Guattery and G.L. Miller, "Graph Embeddings and Laplacian Eigenvalues," *SIAM J. Matrix Analysis and Applications,* vol. 21, no. 3, pp. 703-723, 2000.

[34] I. Dhillon, D. Modha, and W. Spangler, "Visualizing Class Structure of Multidimensional Data," *Proc. 30th Symp. Interface: Computing Science and Statistics,* 1998.

[35] R.B. Calinski and J. Harabasz, "A Dendrite Method for Cluster Analysis," *Comm. Statistics,* vol. 3, pp. 1-27, 1974.

[36] J.W. Tukey, *Exploratory Data Analysis.* Addison-Wesley, 1977.

[37] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the Number of Clusters in a Data-Set via the Gap Statistics," *J. Royal Statistical Soc. B,* vol. 63, pp. 411-423, 2001.

[38] I. Sledge, J. Huband, and J.C. Bezdek, "(Automatic) Cluster Count Extraction from Unlabeled Data-Sets," *Proc. Joint Fourth Int'l Conf. Natural Computation (ICNC) and Fifth Int'l Conf. Fuzzy Systems and Knowledge Discovery (FSKD),* 2008.

[39] J.C. Bezdek and N.R. Pal, "Some New Indices of Cluster Validity," *IEEE Trans. System, Man and Cybernetics,* vol. 28, no. 3, pp. 301-315, June 1998.

[40] *Decomposition Methodology for Knowledge Discovery and Data Mining,* O. Maimon and L. Rokach, eds., pp. 90-94.World Scientific, 2005.

[41] C. Williams and M. Seeger, "Using the Nyström Method to Speed up Kernel Machines," *Advances in Neural Information Processing Systems,* pp. 682-688, MIT Press, 2000.

[42] K. Zhang, I. Tsang, and J. Kwok, "Improved Nyström Low-Rank Approximation and Error Analysis," *Proc. Int'l Conf. Machine Learning,* 2008.

[43] A. Talwalkar, S. Kumar, and H. Rowley, "Large-Scale Manifold Learning," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* 2008.

[44] L. Wang, X. Geng, J. Bezdek, C. Leckie, and R. Kotagiri, "SpecVAT: Enhanced Visual Cluster Analysis," *Proc. Int'l Conf. Data Mining,* 2008.

[45] L. Wang, C. Leckie, J. Bezdek, and R. Kotagiri, "Automatically Determining the Number of Clusters in Unlabeled Data Sets," *IEEE Trans. Knowledge and Data Eng.,* vol. 21, no. 3, pp. 335-350, Mar. 2009.

[46] K.H. Rosen, *Discrete Mathematics and Its Applications.* McGraw-Hill, 1999.

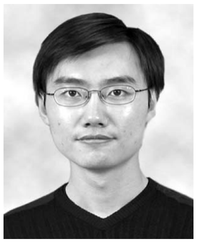
[47] E. Falkenauer, *Genetic Algorithms and Grouping Problems,* John Wiley & Sons, 1997.

[48] T. Havens, J. Bezdek, J. Keller, and M. Popescu, "Clustering in Ordered Dissimilarity Data," technical report, Univ. of Missouri, 2007.

[49] B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach.* Chapman & Hall/CRC, 2005.

[50] L. Wang, C. Leckie, X. Wang, R. Kotagiri, and J. Bezdek, "Tensor Space Learning for Analyzing Activity Patterns from Video Sequences," *Proc. IEEE Int'l Conf. Data Mining (ICDM) Workshop Knowledge Discovery and Data Mining from Multimedia Data and Multimedia Applications,* pp. 63-68, 2007.

Liang Wang received the PhD degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, in 2004. He is currently a professor at the Institute of Automation, Chinese Academy of Sciences, China. He serves as a PC member of leading international conferences and a referee of major international journals. He is an associate editor of the *IEEE Transactions on Systems, Man, and Cybernetics, Part B (TSMC-B)*, the *International Journal of Image and Graphics*, and *Signal Processing*, and a guest editor for three upcoming special issues in the *Pattern Recognition Letters*, the *International Journal of Pattern Recognition and Artificial Intelligence*, and the *TSMC-B*. He has widely published in top international journals and conferences such as the *IEEE Transactions on Pattern Aanalysis and Machine Intelligence*, the *IEEE Transactions on Image Processing*, the *IEEE Transactions on Knowledge and Data Engineering*, the ICDM, the ICCV, and the CVPR. His research interests include pattern recognition, data mining, computer vision, and machine learning. He is a senior member of the IEEE.



Xin Geng received the BSc and MSc degrees in computer science from Nanjing University, China, and the PhD degree in computer science from Deakin University, Melbourne, Australia. His research interests include computer vision, pattern recognition, and machine learning. He has published more than 20 refereed papers in these areas, including those published in prestigious journals and top international conferences. He has been a guest editor of several international journals, such as the *Pattern Recognition Letters*, the *International Journal of Pattern Recognition and Artificial Intelligence*. He has served as a program committee member for several international conferences, such as the PRICAI08, AI08, MMSP08, IEEE IRI09, ICNC09, and CIKM09. He is also a frequent reviewer for various international journals and conferences.

James Bezdek received the PhD degree in applied mathematics from Cornell University in 1973. He is past president of three professional societies: North American Fuzzy Information Processing Society (NAFIPS), International Fuzzy Systems Association (IFSA), and the IEEE Computational Intelligence Society. He is the founding editor of two journals: the *International Journal of Approximate Reasoning* and the *IEEE Transactions on Fuzzy Systems*. He is a fellow of the IEEE and the IFSA, and a recipient of the IEEE 3rd Millennium, IEEE CIS Fuzzy Systems Pioneer, and IEEE CIS Rosenblatt medals. His research interests include woodworking, optimization, motorcycles, pattern recognition, cigars, fishing, image processing, blues music, and cluster analysis.

Christopher Leckie received the BSc degree in 1985, the BE degree in electrical and computer systems engineering in 1987, and the PhD degree in computer science in 1992, all from Monash University, Australia. He joined Telstra Research Laboratories in 1988, where he conducted research and development into artificial intelligence techniques for various telecommunication applications. In 2000, he joined the University of Melbourne, Australia, where he is currently an associate professor with the Department of Computer Science and Software Engineering. His research interests include using artificial intelligence for network management and intrusion detection, and data mining techniques such as clustering.

Kotagiri Ramamohanarao received the PhD degree from Monash University. He was awarded the Alexander von Humboldt Fellowship in 1983. He has been at the University of Melbourne since 1980 and was appointed a professor in computer science in 1989. He held several senior positions including the head of Computer Science and Software Engineering, head of the School of Electrical Engineering and Computer Science, deputy director of the Centre for Ultra Broadband Information Networks, etc. He served as a member of the Australian Research Council Information Technology Panel. He served on the Prime Minister's Science, Engineering, and Innovation Council Working Party on Data for Scientists. At present, he is on the editorial boards for the *Knowledge and Information Systems* and the *IEEE Transactions on Knowledge and Data Engineering*, the *Journal of Statistical Analysis and Data Mining*, and the *VLDB Journal*. He served as a program committee member of several international conferences including the SIGMOD, the IEEE ICDM, the VLDB, and the ICDE. He was the program co-chair for the VLDB, the PAKDD, the DASFAA, and the DOOD conferences. He is a steering committee member of the ICDM, the PAKDD, and the DASFAA. He has research interests in the areas of database systems, agent-oriented systems, information retrieval, data mining, intrusion detection, and machine learning.