2008 Eighth IEEE International Conference on Data Mining

SpecVAT: Enhanced Visual Cluster Analysis

Liang Wang[1]*, Xin Geng[2,3], James Bezdek[1], Christopher Leckie[1], Ramamohanarao Kotagiri[1]
[1] School of Engineering, The University of Melbourne, Vic 3010, Australia
{lwwang,caleckie,rao}@csse.unimelb.edu.au, jbezdek@cs.uwf.edu
[2] School of Computer Science & Engineering, Southeast University, China
[3] State Key Lab. for Novel Software Technology, Nanjing University, China
xgeng@seu.edu.cn

Abstract

Given a pairwise dissimilarity matrix $\boldsymbol{D}$ of a set of objects, visual methods such as the VAT algorithm (for visual analysis of cluster tendency) represent $\boldsymbol{D}$ as an image $\mathrm{I}(\tilde{\boldsymbol{D}})$ where the objects are reordered to highlight cluster structure as dark blocks along the diagonal of the image. A major limitation of such visual methods is their inability to highlight cluster structure in $\mathrm{I}(\tilde{\boldsymbol{D}})$ when $\boldsymbol{D}$ contains clusters with highly complex structure. In this paper, we address this limitation by proposing a Spectral VAT (SpecVAT) algorithm, where $\boldsymbol{D}$ is mapped to $\boldsymbol{D'}$ in an embedding space by spectral decomposition of the Laplacian matrix, and then reordered to $\tilde{\boldsymbol{D}}'$ using the VAT algorithm. We also propose a strategy to automatically determine the number of clusters in $\mathrm{I}(\tilde{\boldsymbol{D}}')$, as well as a method for cluster formation from $\mathrm{I}(\tilde{\boldsymbol{D}}')$ based on the difference between diagonal blocks and off-diagonal blocks. We demonstrate the effectiveness of our algorithms on several synthetic and real-world data sets that are not amenable to analysis via traditional VAT.

1. Introduction

A general question in the data mining community is how to organize observed data into meaningful structures or taxonomies. As an exploratory data analysis tool, cluster analysis aims at grouping objects of a similar kind into their respective categories. Given a data set $\mathcal{O}$ comprising n objects $\{o_1, o_2, \cdots, o_n\}$ (*e.g.*, fish, flowers, beers, *etc*), (crisp) clustering partitions the data into c groups $C_1, C_2, \cdots, C_c$, so that $C_i \cap C_j = \emptyset$, if $i \neq j$ and $C_1 \cup C_2 \cup \cdots \cup C_c = \mathcal{O}$.

There have been a large number of clustering algorithms reported 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 $\boldsymbol{D} \in \mathcal{R}^{n \times n}$ representing a data set of n objects, this paper addresses the first two problems in cluster analysis, *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. A number of attempts have been made to estimate c [24, 13]. However, most methods are *post-clustering* measures of cluster validity. In contrast, tendency assessment attempts to estimate c before clustering occurs. Visual methods for cluster tendency assessment for various data analysis problems have been extensively studied [7]. For data that can be projected onto a 2D or 3D Euclidean space (which are commonly depicted with a scatter plot), direct observation can provide a good insight on the value of c. However, the complexity of most real-world high-dimensional data, or the availability of only pairwise relational data, restricts the effectiveness of this strategy.

The representation of data structures in an image format has a long and continuous history, *e.g.*, [11, 21, 2, 19]. Usually, pairwise dissimilarity information about a set of objects $\{o_1, o_2, \cdots, o_n\}$ is 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. The intensity of each pixel in the image corresponds to the dissimilarity between the pair of objects addressed by the row and column of the pixel. A "useful" reordered dissimilarity image highlights potential clusters as a set of "dark blocks" along the diagonal of the image, and can be viewed as a visual aid to tendency assessment.

This paper focuses on one method for generating reordered dissimilarity images (RDIs), namely VAT (Visual Assessment of cluster Tendency) of Bezdek and Hathaway

*This work was partially supported by Australian Research Council (ARC) Discovery Project under Grant No. DP0663196.

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.18

[2], although our approach can be applied to any method that generates RDIs. Several algorithms extend VAT for related assessment problems, *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 co-clustering. However, 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. Many practical applications involve data sets with highly complex structure, which invalidate the assumption of compact, well-separated clusters. 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 and their approximate size for data sets with highly irregular cluster structures. We also propose an effective strategy to measure the "goodness" of spectral VAT images for automated determination of the number of clusters c. In addition, we derive a visual clustering algorithm based on the spectral VAT image 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.

The remainder of the paper is organized as follows: Section 2 briefly reviews the VAT algorithm, and Section 3 illustrates our spectral VAT algorithm. Section 4 presents two applications of SpecVAT in cluster analysis, *i.e.*, determining the number of clusters c and finding the c clusters from the SpecVAT image. The results of several examples are discussed in Section 5, prior to a summary in Section 6.

2. Brief review of VAT

The VAT algorithm [2] works on a pairwise dissimilarity matrix. Let $\mathcal{O} = \{o_1, o_2, \cdots, o_n\}$ denote n objects in the data and $\boldsymbol{D}$ 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 $1 \geq d_{ij} \geq 0; d_{ij} = d_{ji}; d_{ii} = 0$, for $1 \leq i, j \leq n$. [1]

The VAT algorithm displays a reordered dissimilarity matrix of $\boldsymbol{D}$ as a gray-scale image, each element of which is a scaled dissimilarity value d_{ij} between objects o_i and o_j. Let $\pi()$ be a permutation of $\{1, 2, \cdots, n\}$ such that $\pi(i)$ is the new index for o_i. The reordered list is thus $\{o_{\pi(1)}, \cdots, o_{\pi(n)}\}$. Let $\boldsymbol{P}$ be the permutation matrix with $p_{ij} = 1$ if $j = \pi(i)$ and 0 otherwise, then the matrix $\tilde{\boldsymbol{D}}$ for the reordered list is a similarity transform of $\boldsymbol{D}$ by $\boldsymbol{P}$,

$$\tilde{\boldsymbol{D}} = \boldsymbol{P}^T \boldsymbol{D} \boldsymbol{P} \tag{1}$$

[1] When the objects in $\mathcal{O}$ have a vectorial data representation, *i.e.*, $\mathcal{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_n\}$, $\boldsymbol{f}_i \in \mathcal{R}^r$, where each element of $\boldsymbol{f}_i$ corresponding to the object o_i provides one feature value of r attributes. It is easy to convert $\mathcal{F}$ into $\boldsymbol{D} = [d_{ij} = \|\boldsymbol{f}_i - \boldsymbol{f}_j\|]$ in any vector norm in $\mathcal{R}^r$

Table 1. Algorithm I: VAT

Input: An $n \times n$ scaled matrix of pairwise dissimilarities $\boldsymbol{D} = [d_{ij}]$, with $1 \geq d_{ij} \geq 0; d_{ij} = d_{ji}; d_{ii} = 0$, for $1 \leq i, j \leq n$

(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 $1 \leq i, j \leq n$.

Output: A scaled gray-scale image $I(\tilde{\boldsymbol{D}})$, in which $\max\{\tilde{d_{ij}}\}$ corresponds to *white* and $\min\{\tilde{d_{ij}}\}$ to *black*.

The reordering idea is to find $\boldsymbol{P}$ so that $\tilde{\boldsymbol{D}}$ is as close to a block diagonal form as possible. The VAT algorithm reorders the row and columns of $\boldsymbol{D}$ with a modified version of Prim's minimal spanning tree algorithm. If an object is a member of a cluster, then it should be part of a sub-matrix with low dissimilarity values (corresponding to within-cluster distances), which appears to be one of the dark blocks along the diagonal of the VAT image $\mathrm{I}(\tilde{\boldsymbol{D}})$, each of which corresponds to one cluster. To make the paper self-contained, we summarize the VAT algorithm in Table 1.

An example of VAT is shown in Figure 1. Figure 1(*left*) is a scatter plot of $n = 3000$ points in $\mathcal{R}^2$, which is generated from a mixture of $c = 5$ bi-variate normal distributions. These data points were converted to a 3000×3000 dissimilarity matrix $\boldsymbol{D}$ by computing the Euclidean distance between each pair of points. The 5 visually apparent clusters in Figure 1(*left*) are reflected by the 5 distinct dark blocks along the main diagonal in Figure 1(*right*), which is the VAT image of the data. Given the image of $\boldsymbol{D}$ in the original input order in Figure 1(*middle*), reordering is necessary to reveal the underlying cluster structure of the data.

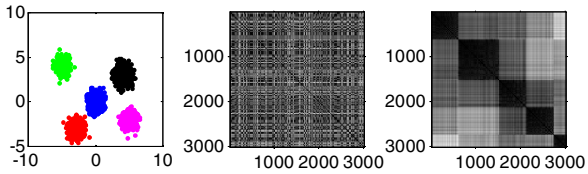


Figure 1. VAT: (*left*) scatter plot of a 2D data set, (*middle*) unordered image $\mathrm{I}(\boldsymbol{D})$, and (*right*) reordered VAT image $\mathrm{I}(\tilde{\boldsymbol{D}})$

Two points about VAT are noted here: 1) only a pairwise dissimilarity matrix $\boldsymbol{D}$ is required as the input. When vectorial forms of objects are available, it is easy to convert them into $\boldsymbol{D}$ using any dissimilarity measure. Even when vectorial data are unavailable, it is still feasible to use

certain flexible metrics to compute a pairwise dissimilarity matrix, *e.g.*, using DTW (Dynamic Time Warping) 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 can be detected from the reordered matrix by the presence of diagonal blocks within larger diagonal blocks.

3. Spectral VAT

At a 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. Note 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, *e.g.*, non-distinct boundaries and an unclear diagonal form. See Figures 5(a) and 6(a) for examples. Accordingly, viewers may deduce different numbers of clusters from such poor-quality images, or even cannot estimate c at all. This raises a problem of whether we can transform $\boldsymbol{D}$ into a new form $\boldsymbol{D'}$ so that the VAT image of $\boldsymbol{D'}$ can become clearer and more informative about the cluster structure. In this paper, we address this problem by combining VAT with spectral graph analysis.

Recently a number of researchers have used spectral analysis of a graph in applications such as graph embedding for dimensionality reduction [1, 26], 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] is 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 $\{o_1, o_2, \cdots, o_n\}$), $E = [e_{ij}]$ is the edge set with $e_{ij} = 1$ showing that there is a link between vertices i and j and 0 otherwise, and $\boldsymbol{W} = [w_{ij}]$, a $n \times n$ 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 the following aspects: 1) *Different graphs*, reflected in E, *e.g.*, the ε-neighborhood graph (connect all vertices whose pairwise distances are smaller than ε); the K-nearest neighbor graph (connect vertices i and j if o_j is among the K nearest neighbors of o_i and/or o_i

Table 2. Algorithm II: SpecVAT

Input: $\boldsymbol{D} = [d_{ij}]$: an $n \times n$ scaled matrix of pairwise dissimilarities, with $1 \geq d_{ij} \geq 0; d_{ij} = d_{ji}; d_{ii} = 0$, for $1 \leq i, j \leq n$

k: the number of eigenvectors used (or the dimension of the embedding subspace).

(1): Compute a local scale σ_i for each object o_i using

$$\sigma_i = d(o_i, o_K) = d_{iK} \tag{2}$$

where o_K is the K-th nearest neighbor of o_i. [a]

(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): Let $\boldsymbol{M}$ to be a diagonal matrix with $m_{ii} = \sum_{j=1}^{n} w_{ij}$ (*i.e.*, the (i, i) element of $\boldsymbol{M}$ is the sum of $\boldsymbol{W}$'s i-th row), and construct the matrix

$$\boldsymbol{L'} = \boldsymbol{M}^{-1/2} \boldsymbol{W} \boldsymbol{M}^{-1/2} \tag{3}$$

which is a normalized version of the Laplacian matrix. [b]

(4): Choose $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_k$, 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 $\boldsymbol{V'} \in \mathcal{R}^{n \times k}$, *i.e.*, $v'_{ij} = v_{ij}/\|v_{i:}\|$.

(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 in the k-dimensional embedding space (corresponding to original o_i). [c] Then construct a new pairwise dissimilarity matrix $\boldsymbol{D'}$ between objects by defining $d'_{ij} = \|\boldsymbol{u}_i - \boldsymbol{u}_j\|$.

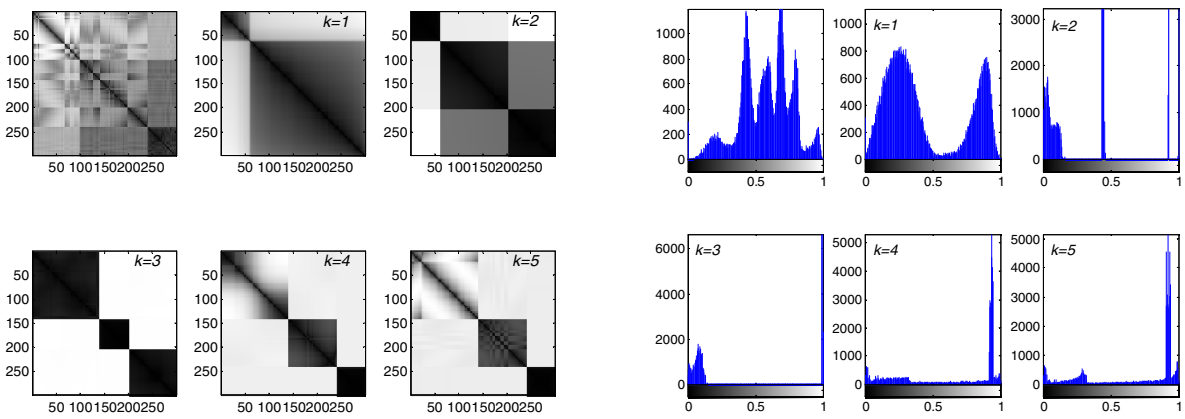
(7): Apply the VAT algorithm to $\boldsymbol{D'}$ to obtain $\mathrm{I}(\tilde{\boldsymbol{D}}')$.

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

[a] Using a specific scaling parameter allows self-tuning of the object-to-object distance according to the local statistics of the neighborhoods surrounding objects i and j, resulting in high affinities within clusters and low affinities across clusters [27].

[b] Note that $\boldsymbol{M}^{(-1/2)}(\boldsymbol{M} - \boldsymbol{W})\boldsymbol{M}^{(-1/2)} = \boldsymbol{I} - \boldsymbol{L'}$. Replacing $\boldsymbol{I} - \boldsymbol{L'}$ with $\boldsymbol{L'}$ 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 non-constant blocks in [15, 23].

[c] This is actually graph embedding [1]. 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 original objects are transformed into new representations.



(a) (b)

Figure 2. (a) Original VAT (*top-left*) and SpecVAT images with different k of synthetic data S-1, and (b) their corresponding gray-scale histograms

is among the K nearest neighbors of o_j), and the fully connected graph (simply connect all vertices with each other); 2) *Different weighting functions*, reflected in $\boldsymbol{W}$, *e.g.*, simple 0-1 weighting or the commonly used Gaussian similarity function $s(\boldsymbol{f}_i, \boldsymbol{f}_j) = \exp(-\frac{\|\boldsymbol{f}_i - \boldsymbol{f}_j\|^2}{2\sigma^2}) = \exp(-\frac{d_{ij}^2}{2\sigma^2})$ with a global scaling parameter σ; and 3) *Different 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 matrix whose elements are the degrees of the nodes of $\mathcal{G}$, *i.e.*, $m_{ii} = \sum_{j=1}^{n} w_{ij}$.

The spectral decomposition of the Laplacian matrix provides useful information about the properties of the graph [6]. 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]. 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. Such an embedding for the data comes from approximations to a natural map that is defined on the entire data manifold [1]. After a comprehensive study of recent spectral methods, we adopt a combination of adjacency graph, weighting function and graph Laplacian for obtaining better graph embedding (and thus better SpecVAT images, see Figures 5(b) and 6(b)). We summarize our spectral VAT algorithm in Table 2.

4. Applications of Spectral VAT

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. We explore the use of spectral VAT for these two problems. That is, we wish to answer the following two questions:

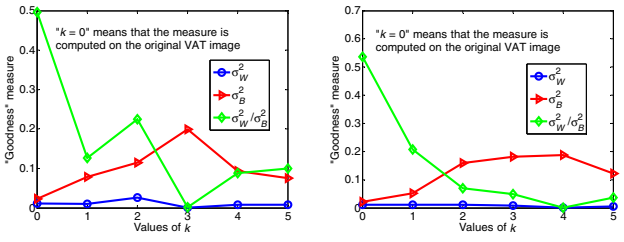
(Q1) Can we automatically determine the number of clusters c, as suggested by $\mathrm{I}(\tilde{\boldsymbol{D}}')$, in an objective manner, without viewing the visual display? This enables us to capitalize on the information possessed by the SpecVAT image.

(Q2) Can we automatically extract a crisp c-partition of $\mathcal{O}$, which is suggested by the visual evidence in $\mathrm{I}(\tilde{\boldsymbol{D}}')$? If so, how well does it perform?

4.1. Determining the number of clusters

Figure 2(a) shows an example of the original VAT image and SpecVAT images with different numbers of eigenvectors (corresponding to synthetic data S-1 in Figure 4). We can see that the SpecVAT images are generally clearer than the original VAT image in revealing real data structure. See Figures 5 and 6 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". The corresponding gray-scale histograms in Figure 2(b) suggest that a good SpecVAT image should, ideally, include two explicit modalities in the distributions, ideally 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



(a) S-1 (b) S-5

Figure 3. "Goodness" measures of original VAT and SpecVAT images with different k

Table 3. Algorithm III: ADNC

Input: $\boldsymbol{D} = [d_{ij}]$: an $n \times n$ scaled matrix of pairwise dissimilarities, with $1 \geq d_{ij} \geq 0; d_{ij} = d_{ji}; d_{ii} = 0$, for $1 \leq i, j \leq n$

k_{max}: the user-specified maximum of the number of the eigenvectors used in spectral VAT

For $k = 1$ to k_{max}

(1): Perform spectral VAT to obtain $\mathrm{I}(\tilde{\boldsymbol{D}}'_k)$;

(2): Compute an optimal threshold T_k^* that can maximize σ_B^2 for the image $\mathrm{I}(\tilde{\boldsymbol{D}}'_k)$, *i.e.*, $T_k^* = \arg\max_{1 \leq T < L} \sigma_B^2(T)$;

(3): Compute the corresponding "goodness" measure $GM(k) = \sigma_B^2(T_k^*)$.

End

Output: The number of clusters $c = \arg\max_k GM(k)$.

appropriate criterion for evaluating the "goodness" of the SpecVAT images from a more general standpoint. 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.*, $p_l = n_l/N, p_l > 0, \sum_{l=1}^{L} p_l = 1$.

Suppose that the image pixels can be divided into two classes C_1 and C_2 (*e.g.*, implicitly corresponding to "within-cluster blocks" and "between-cluster blocks" in the VAT or SpecVAT image) by a threshold at level T. C_1 denotes pixels with levels $[1, \cdots, T]$ and C_2 denotes pixels with levels $[T+1, \cdots, L]$. 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

$$\mu_1 = \sum_{l=1}^{T} lP(l|C_1) = \sum_{l=1}^{T} lp_l/\omega_1 = \mu(T)/\omega(T) \quad (4)$$

$$\mu_2 = \sum_{l=T+1}^{L} lP(l|C_2) = \sum_{l=T+1}^{L} lp_l/\omega_2 = \frac{\mu_L - \mu(T)}{1 - \omega(T)} \quad (5)$$

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 T-th 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

$$\sigma_1^2 = \sum_{l=1}^{T} (l - \mu_1)^2 P(l|C_1) = \sum_{l=1}^{T} (l - \mu_1)^2 p_l/\omega_1 \quad (6)$$

$$\sigma_2^2 = \sum_{l=T+1}^{L} (l - \mu_2)^2 P(l|C_2) = \sum_{l=T+1}^{L} (l - \mu_2)^2 p_l/\omega_2 \quad (7)$$

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

$$\eta = \sigma_B^2/\sigma_W^2, \quad \gamma = \sigma_T^2/\sigma_W^2, \quad \xi = \sigma_B^2/\sigma_T^2 \quad (8)$$

where

$$\sigma_W^2 = \omega_1\sigma_1^2 + \omega_2\sigma_2^2 \quad (9)$$

$$\sigma_B^2 = \omega_1\omega_2(\mu_2 - \mu_1)^2 \quad (10)$$

$$\sigma_T^2 = \sum_{l=1}^{L} (l - \mu_L)^2 p_l = \sigma_W^2 + \sigma_B^2 \quad (11)$$

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 (these discriminant criteria are equivalent to one another). Note that σ_W^2 and σ_B^2 are functions of T, but 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 (or ease of thresholding) for the image or the bi-modality of the histogram [17]. 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.*, $k = 1 \sim k_{max}$), 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. Accordingly, we select the best SpecVAT image as the one with the maximum goodness value, and determine the number of clusters as

$$c = \arg \max_{k \in \{1, \cdots, k_{max}\}} \sigma_B^2(T^*, k) \quad (12)$$

We give two examples of "goodness" values in Figure 3, showing the effectiveness of such a measure in determining a relatively good SpecVAT image (*e.g.*, $k = 3$ for S-1 and $k = 4$ for S-5). Table 3 summarizes the algorithm for Automatically Determining the Number of Clusters (ADNC) from $\boldsymbol{D}$.

Table 4. Algorithm IV: VC (Visual Clustering)

Input: $\mathrm{I}(\tilde{\boldsymbol{D}}')$: image generated from n object samples;
c: the number of clusters to seek; and
b: size of the population.

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

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

(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. (18).

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

(5): Transform $\boldsymbol{x}^*$ into cluster partition $\boldsymbol{U^*}$. 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 a the j-th cluster partition is from sample $(p_{j-1}+1)$ to p_j. The last (the c-th) cluster partition is from sample $(p_{c-1}+1)$ to n.

Output: The sizes of each cluster $\{n_1, \cdots, n_c\}$. Together with the permutation index $\pi()$ obtained during re-ordering, real object indices in each cluster C_i can be retrieved, *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 $i = 2, \cdots, c$.

4.2. Visual clustering

We now consider how to find data clusters from a given RDI, in which the proximity matrix has been reordered as close as possible to a block diagonal form. The c-partitions of $\mathcal{O}$ are generally 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 non-degenerate c-partition matrices for $\mathcal{O}$ is

$$H_{hcn} = \{\boldsymbol{U} \in \mathcal{R}^{c \times n} | 0 \le u_{ik} \le 1, \forall i, k\} \tag{13}$$

$$\text{with} \sum_{i=1}^{c} u_{i,k} = 1, \forall k \text{ and } \sum_{k=1}^{n} u_{ik} > 0, \forall i \tag{14}$$

Element u_{ik} of $\boldsymbol{U}$ is the membership of object k in cluster i. In the case of "crisp" 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} for a given c under consideration to those partitions that mimic the block structure in $\mathrm{I}(\tilde{\boldsymbol{D}}')$ [9]. $\boldsymbol{U}$ in H_{hcn} is called an aligned c-partition of $\mathcal{O}$ when its entries form c contiguous blocks of 1's in $\boldsymbol{U}$, ordered to begin from the upper left corner, and proceeding down and to the right, *i.e.*,

$$H^*_{hcn} = \{\boldsymbol{U} \in H_{hcn}\} \tag{15}$$

with the properties of $u_{1k} = 1, 1 \le k \le n_1; u_{ik} = 1, n_{i-1} < k \le n_i, 2 \le i \le c$. The special nature of aligned partitions enables us to specify them in an alternative form. 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 specified by $\{n_1 : \cdots : n_c\}$, *e.g.*, $\boldsymbol{U} = [1\ 1\ 0\ 0\ 0; 0\ 0\ 1\ 1\ 0; 0\ 0\ 0\ 0\ 1] = \{2 : 2 : 1\}$.

The important characteristics of $\mathrm{I}(\tilde{\boldsymbol{D}}')$ that can be exploited for finding a good $\boldsymbol{U}$ are the contrast difference between the dark blocks along the main diagonal and the pixels adjacent to them. The proposed algorithm aims to generate candidate partitions in H^*_{hcn} by testing their fitness to the clusters suggested by the aligned dark blocks in $\mathrm{I}(\tilde{\boldsymbol{D}}')$. Towards this end, an objective function is defined to implicitly account for two components of block structures: "squareness" and "edginess". An intuitively appealing measure is the difference of the mean dissimilarity values between apparent clusters (*i.e.*, dissimilarities in non-dark blocks off-diagonal) and those within apparent clusters (*i.e.*, dissimilarities in dark blocks along the diagonal).

Let $\boldsymbol{U}$ be a candidate partition 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 non-dark 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

$$E_b = \sum_{i=1}^{c} \Big(\sum_{s \in i, t \ni i} \tilde{d'^*_{st}} \Big) / \sum_{i=1}^{c} n_i (n - n_i) \tag{16}$$

$$E_w = \sum_{i=1}^{c} \Big(\sum_{s,t \in i, s \ne t} \tilde{d'^*_{st}} \Big) / \sum_{i=1}^{c} n_i (n_i - 1) \tag{17}$$

The objective function is defined as

$$E(\boldsymbol{U}, \tilde{\boldsymbol{D}}') = E_b - E_w \tag{18}$$

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

$$\boldsymbol{U^*} = \arg \max_{\boldsymbol{U} \in M^*_{hcn}} E(\boldsymbol{U}, \tilde{\boldsymbol{D}}') \tag{19}$$

In principle, a number of optimization algorithms might be used. We use a Genetic Algorithm (GA) [8]. 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, 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 clustering procedure is summarized in Table 4, leading to a new clustering algorithm in which the final randomly initialized K-means stage (as used in spectral clustering) is eliminated.

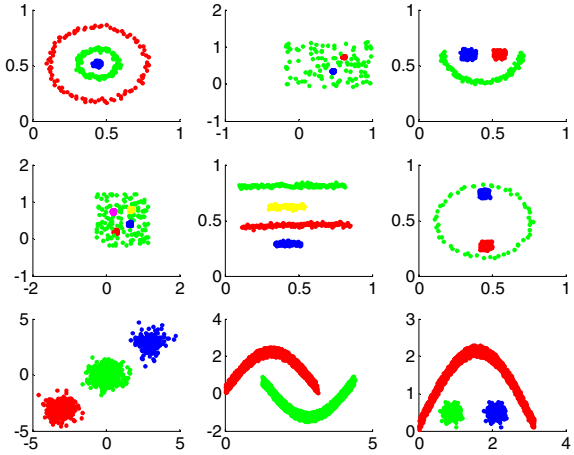


Figure 4. Scatter plots of 9 synthetic data sets. From left to right and from top to bottom: S-1 $\sim$ S-9

5. Experimental results

In order to test our algorithms, we carried out a number of experiments on 9 artificially generated data sets, as well as 6 real-world data sets. Unless otherwise mentioned, in the following experiments the (Euclidean) distance matrix $\boldsymbol{D}$ is computed in the original attribute space (if the object vectorial representation is available).

5.1. Evaluation data

Synthetic data: Nine synthetic data sets with different data structures are used in our experiments. The scatter plots of these synthetic data sets are shown in Figure 4, in which each color represents a visually meaningful group. The first 6 data sets are from [27] [2], 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. In addition, some of them include different scales between clusters, or some clusters are hidden in a cluttered background.

Table 5. Summary of the data sets used

Data	c	# attri.	n	$[n_1, \cdots, n_{c_p}]$
S-1	3	2	299	[61 99 139]
S-2	3	2	303	[95 106 102]
S-3	3	2	266	[73 118 75]
S-4	5	2	622	[150 111 136 109 116]
S-5	4	2	512	[150 122 123 117]
S-6	3	2	238	[100 56 82]
S-7	3	2	1000	[291 519 190]
S-8	2	2	2000	[1000 1000]
S-9	3	2	2000	[500 1000 500]
R-1	2	9	683	[444 239]
R-2	3	1200	1755	[585 585 585]
R-3	3	-	194	[21 87 86]
R-4	3	4	150	[50 50 50]
R-5	2	16	435	[267 168]
R-6	3	13	178	[59 71 48]

Real data: Six real-world data sets were also considered to evaluate our algorithms, four of which are from the UCI Machine Learning Repository [3]. These data sets are summarized as follows: a) R-1: *Breast-cancer* database includes 699 instances, each of which has 9 attributes and belongs to one of 2 classes. Since there are 16 instances that contain a single missing attribute value, we removed them and used the remaining 683 instances for our experiment. b) R-2: This data set was used in [4]. It contains single-light-source *Face* images of 3 different individuals, each seen under 585 viewing conditions. Each original image was down-sampled to 30×40 pixels, thus providing in total 1755 images with 1200 dimensions (*i.e.*, 30×40). c) R-3: *Genetic* data set is originally from the work in [18], which is a 194×194 matrix consisting of pairwise dissimilarities from a set of 194 human gene products that were clustered into three protein families. d) R-4: *Iris* data set contains 3 physical classes, 50 instances each, where each class refers to a type of iris plant and the attributes of each instance include 4 numeric values. e) R-5: *Voting* data set consists of 435 US House of Representatives members' votes on 16 key votes (267 democrats and 168 republicans). Votes were numerically encoded as 0.5 for "yea", -0.5 for "nay" and 0 for "unknown disposition", so that the voting record of each congressman is represented as a ternary-valued vector in $\mathcal{R}^{16}$. f) R-6: *Wine* data set contains the results of a chemical analysis of wines grown in the same region, but derived from 3 different cultivars. The analysis determines the quantities of 13 constituents found in each of three types of wines. The total number of instances is 178.

[2] http://www.vision.caltech.edu/lihi/Demos/SelfTuningClustering.html

[3] http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html

Table 6. The results of c estimated

Data	c_p [a]	c_{ov}^m [b]	c_{sv}^m [c]	c_{sv}^a [d]
S-1	3	≥ 1	3	3
S-2	3	≥ 2	3	3
S-3	3	≥ 2	3	3
S-4	5	≥ 4	5	5
S-5	4	≥ 2	4	4
S-6	3	≥ 2	3	3
S-7	3	3	3	3
S-8	2	≥ 1	2	2
S-9	3	≥ 2	3	3
R-1	2	≥ 3	2	2
R-2	3	3 or 4	3	3
R-3	3	≥ 3	3	3
R-4	3	≥ 2	2	2
R-5	2	≥ 2	2	2
R-6	3	≥ 3	3	3

[a] c_p: The number of real physical classes in the data
[b] c_{ov}^m: c determined by manual inspection from the original VAT image
[c] c_{sv}^m: c determined by manual inspection of a series of SpecVAT images
[d] c_{sv}^a: c determined automatically from a series of SpecVAT images

5.2. Determining c

The characteristics of the synthetic and real data sets are clearly summarized in Table 5. For each of them (except for R-5), we computed a pairwise dissimilarity matrix $\boldsymbol{D}$ in the original attribute space. The VAT images are shown in Figure 5(a) for synthetic data and Figure 6(a) for real data. 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 estimations of c. Further, we performed the SpecVAT algorithm, and showed SpecVAT images in Figure 5(b) for synthetic data and Figure 6(b) for real data. In contrast to the original VAT images, the SpecVAT images have generally clearer displays in terms of block structure, thus better highlighting the hidden cluster structure. Table 6 summarizes the number of clusters determined from SpecVAT images automatically, along with the results estimated from the VAT/SpecVAT images using manual inspection by the authors for comparison. For iris, our method gives $c = 2$. This is due to 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. The results of cluster number estimation from the SpecVAT images for other 14 data sets are accurate in terms of the number of real physical classes, whether it was estimated automatically by our ADNC algorithm or by manual inspection. The results again highlight the benefits of converting $\boldsymbol{D}$ to $\boldsymbol{D'}$ by graph embedding for obtaining more accurate estimation of c.

Table 7. Clustering algorithm comparison (%)

Data	c	K_m [a]	L_w [b]	S_σ [c]	S_{σ_i} [d]	V_{sv}	V_{ov}
S-1	3	45.6	48.8	**100**	**100**	**100**	63.2
S-2	3	73.7	71.6	84.5	**100**	**100**	84.5
S-3	3	74.1	75.6	**100**	**100**	**100**	79.0
S-4	5	79.5	82.8	88.9	**100**	**100**	88.6
S-5	4	69.4	70.7	**100**	**100**	**100**	84.2
S-6	3	82.6	83.2	96.6	**100**	**100**	82.8
S-7	3	97.2	**100**	**100**	**100**	**100**	**100**
S-8	2	88.3	71.7	**100**	**100**	**100**	92.2
S-9	3	76.1	78.0	**100**	**100**	**100**	57.5
R-1	2	96.1	96.6	**96.8**	**96.8**	94.9	65.2
R-2	3	94.8	**100**	99.8	99.8	99.7	96.2
R-3	3	-	-	**100**	**100**	**100**	**100**
R-4	2	98.0	**100**	**100**	**100**	**100**	99.3
R-4	3	81.5	89.3	90.7	**93.3**	92.7	67.3
R-5	2	88.1	**91.7**	87.1	88.1	90.8	83.5
R-6	3	96.3	92.7	97.8	97.8	**98.3**	39.3

[a] K-means algorithm
[b] Ward's hierarchy clustering
[c] Spectral clustering using a global scale σ [15]
[d] Spectral clustering using local scales σ_i [27]

5.3. Visual clustering and comparison

We evaluate our visual clustering algorithm's performance by comparing the cluster labels of the objects given by our algorithm with the ground-truth labels. 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 ground truth label, *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]. The clustering accuracy of our VC algorithm on the original VAT image (V_{ov}) and the SpecVAT image (V_{sv}) is summarized in Table 7, from which we can see that no doubt V_{sv} obtains better results than V_{ov}.

We also implemented several typical clustering algorithms for comparison. These algorithms are K-means, and Ward's hierarchal clustering algorithm [14], spectral clustering with a global scale σ [15], and spectral clustering with local scale σ_i [27]. For the K-means algorithm, we reported the average accuracy of the result over 100 trials. The clustering accuracies of these algorithms are listed in Table 7, in which the best results for each data set are bolded. From Table 7, we can see that the overall accuracy

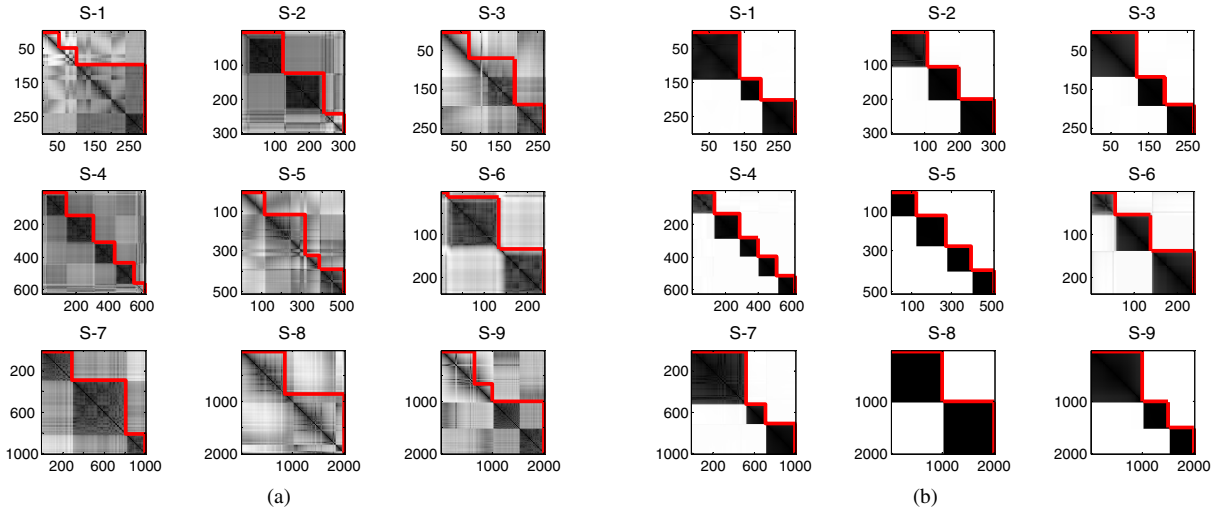


Figure 5. (a) Original VAT images of 9 synthetic data sets with visual clustering results imposed by red lines, and (b) the corresponding best SpecVAT images with visual clustering results

of our clustering algorithm on the SpecVAT image is better than that of K-means, Ward's algorithm and standard spectral clustering with a global scale, 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, as well as eliminating the randomly initialized K-means stage (as usually used in spectral clustering).

5.4. Discussion

There are strong relations between the SpecVAT algorithm (and thus VC) 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] uses K-means to cluster them; while for our visual clustering algorithm, we first convert them to a new reordered image (corresponding to a new pairwise dissimilarity matrix), and then use the GA to partition its block structures. A local scaling scheme is suggested in [27] 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 VC algorithm can perform competitively with spectral clustering [27, 15]. The slight difference in accuracy between VC and the algorithm of [27] could be due to the difference between 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 $\boldsymbol{D}$ is not from a well-structured dissimilarity matrix. While our ADNC algorithm may return a slightly over-estimated or underestimated value of c, it provides an initial estimate of the cluster number, thus avoiding running a clustering algorithm multiple times over a wide range of c in an attempt to find valid clusters. In this way, our method compares favorably to post-clustering 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), but it improves the probability of success.

There are other methods besides VAT that produce RDIs, *e.g.*, [11, 21, 19]. We are thus not restricted to using VAT, and can apply our algorithms to the output of any method for finding reordered dissimilarity images.

6. Conclusion

This paper has presented a new visual technique for automatically determining the number of clusters and partitioning either object or pairwise relational data. Our contributions are summarized as follows: 1) The VAT algorithm is enhanced by using spectral analysis of the proximity matrix of the data. The new SpecVAT algorithm can better reveal the hidden cluster structure, especially for complex-shaped data sets. 2) Based on spectral VAT, the cluster structure in the data can be reliably estimated by visual inspection. As well, we propose a "goodness" measure of SpecVAT images for automatically determining the number of clusters. 3) We derive a visual clustering algorithm based on SpecVAT images and its unique block-structured property. 4) We perform a series of primary and comparative experiments on 9 synthetic data sets and 6 real-world data sets, and obtain encouraging results in terms of the first two major problems in cluster analysis.

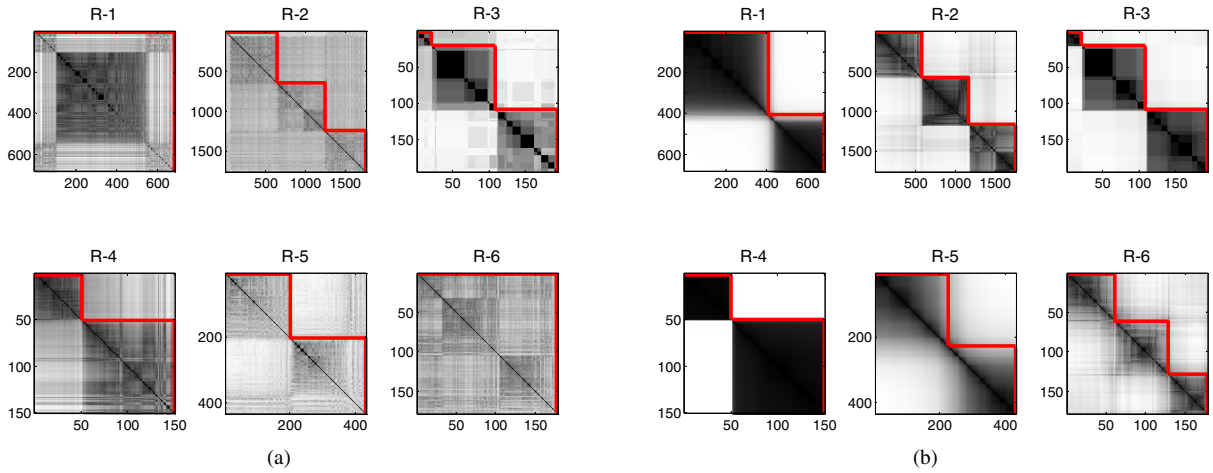


Figure 6. (a) Original VAT images of 6 real-world data sets with visual clustering results imposed by red lines, and (b) the corresponding best SpecVAT images with visual clustering results

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Proc. Advances in Neural Information Processing Systems*, 2002.

[2] J. C. Bezdek and R. J. Hathaway. VAT: A tool for visual assement of (cluster) tendency. In *Proc. International Joint Conference on Neural Networks*, pages 2225–2230, 2002.

[3] J. C. Bezdek, R. J. Hathaway, and J. Huband. Visual assessment of clustering tendency for rectangular dissimilarity matrices. *IEEE Transactions on Fuzzy Systems*, 15(5):890–903, 2007.

[4] M. Breitenbach and G. Grudic. Clustering through ranking on manifolds. In *Proc. International Conference on Machine Learning*, 2005.

[5] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):1624–1637, 2005.

[6] F. Chung. Spectral graph theory. In *CBMS Regional Conference Series in Mathematics, American Mathematical Society*, volume 92, 1997.

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

[8] E. Falkenauer. *Genetic Algorithms and Grouping Problems*. John Wiley & Sons Ltd, Chichester, England, 1997.

[9] T. Havens, J. Bezdek, J. Keller, and M. Popescu. Clustering in ordered dissimilarity data. Technical report, University of Missouri, Columbia, MO, 2007.

[10] J. Huband, J. C. Bezdek, and R. Hathaway. bigvat: Visual assessment of cluster tendency for large data sets. *Pattern Recognition*, 38(11):1875–1886, 2005.

[11] R. Ling. A computer generated aid for cluster analysis. *Communications of the ACM*, 16:355–361, 1973.

[12] L. Lovasz and M. Plummer. *Matching Theory*. Budapest, Akadémiai Kiadó, North Holland, 1986.

[13] U. Maulik and S. Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654, 2002.

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

[15] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. Advances in Neural Information Processing Systems*, 2002.

[16] H. Z. Ning, W. Xu, Y. Chi, and T. S. Huang. Incremental spectral clustering with application to monitoring of evolving blog communities. In *Proc. SIAM International Conference on Data Mining*, 2007.

[17] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[18] N. Pal, J. Keller, M. Popescu, J. Bezdek, J. Mitchell, and J. Huband. Gene ontology-based knowledge discovery through fuzzy cluster analysis. *Journal of Neural, Parallel and Scientific Computing*, 13:337–361, 2005.

[19] P. J. Rousseeuw. A graphical aid to the interpretations and validation of cluster analysis. *J. Compu. Appl. Math.*, 20:53–65, 1987.

[20] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[21] T. Tran-Luu. *Mathematical Concepts and Novel Heuristic Methods for Data Clustering and Visualization*. PhD Thesis, University of Maryland, College Park, MD, 1996.

[22] U. von Luxburg. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics, Germany, 2006.

[23] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *Proc. IEEE International Conference on Computer Vision*, pages 975–982, 1999.

[24] R. Xu and D. W. II. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[25] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proc. ACM SIGIR Conference on Information Retrieval*, 2003.

[26] S. Yan, D. Xu, B. Zhang, and H. Zhang. Graph embedding: A general framework for dimensionality reduction. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2005.

[27] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Proc. Advances in Neural Information Processing Systems*, 2004.