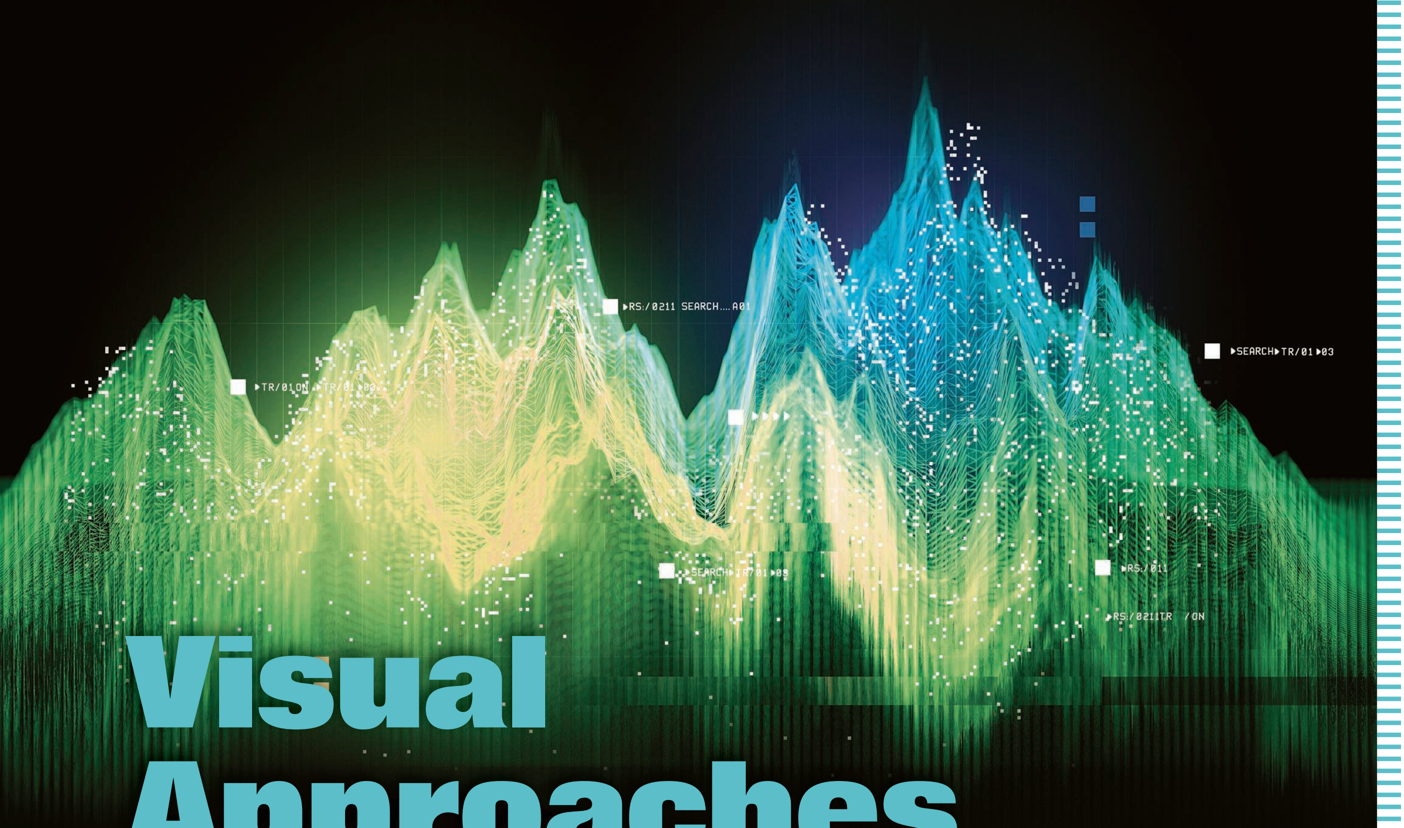


©ISTOCKPHOTO.COM/SOLARSEVEN

Visual Approaches for Exploratory Data Analysis

A Survey of the Visual Assessment of Clustering Tendency (VAT) Family of Algorithms

by Dheeraj Kumar and James C. Bezdek

Exploratory data analysis (EDA) using data clustering is extremely important for understanding the basic characteristics of a novel data set before developing complex statistical models and testing the various hypotheses. A preliminary step to clustering is deciding whether the data contain any clusters and, if so, how many clusters to seek. This is the clustering-tendency-assessment problem, which has not received much attention in the pattern-recognition literature. An important category of algorithms in this domain includes visual approaches, represented here by the visual assessment of tendency (VAT) algorithm, which reorders the pairwise dissimilarity matrix and then generates a reordered dissimilarity image (RDI) or cluster heat map that shows possible clusters in the data by dark blocks along the diagonal.

Since its introduction in 2002, the VAT algorithm has been modified by many researchers to improve the quality of the RDI, making it applicable to various types of data sets, such as high-volume, time-series, high-dimensional, and streaming data, among others (collectively called the

Digital Object Identifier 10.1109/MSMC.2019.2961163
Date of current version: 20 April 2020

2333-942X/20©2020IEEE

VAT family of algorithms). Various members of the VAT family have been applied to many applications, including image segmentation, urban mobility, transportation, speech processing, biomedical applications, social media, and Web data analytics, on a variety of real-life data sets with diverse characteristics and properties.

We hope that this detailed and systematic survey of the VAT family of algorithms and their applications will help researchers choose a useful member of the VAT family to help them understand structural details in their data. This article includes pseudocode for a suite of 25 algorithms in the VAT family of models, and the MATLAB implementation of selected algorithms are available on GitHub [1].

Clustering Tendency as a Tool for Exploratory Data Analysis

Unsupervised data-mining techniques, such as data clustering, are an important part of EDA, which aims to summarize and visualize the main characteristics of the data before developing complex statistical models and testing various hypotheses about structure in the data. Recent advances in sensing and storage technology and dramatic growth in applications on the Internet, digital imaging, video surveillance, and the Internet of Things (IoT) have accelerated the growth of data collection. With the ever-increasing availability of data across different disciplines, data clustering as a fundamental tool for EDA has gained more significance.

Data clustering aims to divide the data into several groups such that data points in each group are more similar to each other in some well-defined sense than to the points in other groups. Various clustering techniques have been developed over the years by researchers in many fields, including taxonomists, social scientists, psychologists, biologists, statisticians, mathematicians, engineers, computer scientists, and medical researchers [2]. Some of the most popular clustering approaches include hierarchical clustering (agglomerative and divisive), centroid-based approaches (k-means, fuzzy c-means, and so on), density-based algorithms [e.g., density-based spatial clustering of applications with noise (DBSCAN) and ordering points to identify the clustering structure (OPTICS)], and distribution-based clustering [expectation maximization, Gaussian mixture model (GMM), and so on] [3]–[5].

A natural question to ask before applying any clustering method to a data set is, "Does this data set contain any clusters and, if so, how many?" A major issue with unsupervised machine learning is that clustering methods will return clusters that satisfy the constraints of the algorithm that produces them, even if the data do not contain any clusters. Blindly applying a clustering analysis to a data set will divide the data into clusters, even if there are none, because that is what the algorithm is supposed to do. Therefore, before applying a clustering approach to a data set, the analyst must decide whether or not the data set contains meaningful clusters (i.e., nonrandom structures).

The issue of determining whether clusters are present as a step before actual clustering is called the *clustering-tendency-assessment problem*. Unfortunately, this has received very little attention in the pattern-recognition and exploratory-data-analysis literature. Some techniques for clustering-tendency assessment are discussed in [3] and [6], and they can be broadly split into two categories: statistical and visual. The statistical approaches to clustering tendency assessment, such as the dip test [7], Silverman test [8], and Hopkins statistics [9], apply the random-position hypothesis to check whether or not the data are generated from a continuous uniform distribution. A detailed description of such techniques can be found in [3] and [10]. Although these statistical approaches determine whether it is worth looking for clusters in the given data set by applying clustering algorithms, they provide little information about how many clusters to look for, an input required by many clustering algorithms. Moreover, statistical tests determine only whether the data fail to satisfy a distributional assumption, so they impose a strong constraint on the definition of clusters in the data.

Another class of clustering-tendency-assessment approaches uses visual techniques to indicate whether or not the data set contains possible clusters and, if so, how many clusters to seek. Bezdek and Hathaway [11] introduced a visual approach for assessing cluster tendency, VAT, that can be used in all cases involving numerical data. VAT uses a variant of Prim's algorithm [12] to perform matrix reordering (seriation) of the pairwise dissimilarity matrix to generate a reordered dissimilarity matrix, which, when viewed as a monochrome image (called an *RDI*, or *cluster heat map*), shows possible clusters in the data set by dark blocks along the diagonal.

The literature of the VAT family of algorithms and its applications is large and varied, with papers published in many different journals and conference proceedings. This diversity makes it difficult for researchers to follow recent developments and determine the applicability of these algorithms to their data. We believe that a survey of the VAT family of algorithms and their applications is timely and hope that it will be helpful for researchers choosing a useful visualization algorithm for EDA.

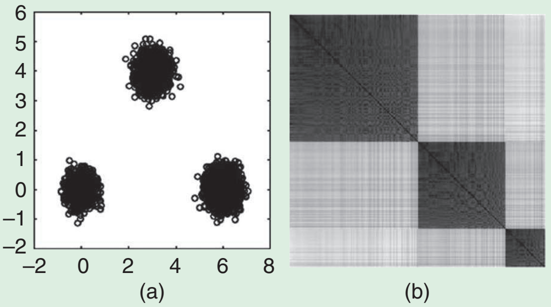


Figure 1. The data scatterplot VAT images for $N = 5{,}000$ Gaussian clusters: (a) data set of $N = 5{,}000$ and (b) VAT image of $I(D_E^*(X))$.

Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family

The MATLAB implementations for selected algorithms are available on Github [1].

Algorithm S1. Visual Assessment of Tendency (VAT) [11].

```
Input  : $D - n \times n$ dissimilarity matrix satisfying
             $-D_{ij} \geq 0$
             $-D_{ij} = D_{ji} \forall i, j$
             $-D_{ii} = 0 \forall i$
Output: $D^* - n \times n$ VAT reordered dissimilarity matrix
         $I(D^*) -$ VAT image of $D^*$
         $P -$ VAT reordering indices of D
         $d -$ ordering of MST cut magnitudes

Set $K = \{1, 2, \dots, n\}, I = J = \emptyset$
Select $(i, j) \in \underset{k \in K, q \in K}{\operatorname{argmax}} D_{kq}$
$P_1 = i$;  $I = \{i\}$ and $J = K - \{i\}$
for $t \leftarrow 2$ to $n$ do
    Select $(i, j) \in \underset{k \in I, q \in J}{\operatorname{argmin}} D_{kq}$
    $P_t = j; I = I \cup \{j\}; J = J - \{j\}$;  $d_{t-1} = D_{ij}$
end
for $p \leftarrow 1$ to $n$ do
    for $q \leftarrow 1$ to $n$ do
        $D^*_{p,q} = D_{P_p, P_q}$
    end
end
Create $I(D^*)$
```

Algorithm S2. Improved VAT (iVAT) [14].

```
Input  : $D^* - n \times n$ VAT-reordered dissimilarity matrix
Output: $D'^* - n \times n$ iVAT dissimilarity matrix

for $r \leftarrow 2$ to $n$ do
    $j = \underset{1 \leq k \leq r-1}{\operatorname{argmin}} \{D^*_{rk}\}$
    $D'^*_{rj} = D^*_{rj}$
    $c = \{1, 2, \dots, r-1\} - \{j\}$
    $D'^*_{rc} = \max\{D^*_{rj}, D'^*_{jc}\}$
end
$D'^*_{rc} = D'^*_{cr}$
```

Algorithm S3. Spectral VAT [18].

```
Input  : $D - n \times n$ dissimilarity matrix satisfying
             $-D_{ij} \geq 0$
             $-D_{ij} = D_{ji} \forall i, j$
             $-D_{ii} = 0 \forall i$
         $k -$ Dimension of the embedding subspace
Output: $\tilde{D}' -$ Spectrally-mapped and reordered
         dissimilarity matrix

Compute a local scale $\sigma_i$ for each object $o_i$
$\sigma_i = d(o_i, o_K) = d_{iK}$, where $o_K$ is the $k$th nearest
  neighbor of $o_i$
Construct weight matrix $W \in \mathbf{R}^{n \times n}$
Construct diagonal matrix $M$ with $m_{ii} = \sum_{j=1}^{n} w_{ij}$
Construct normalized Laplacian matrix: $L' = M^{-1/2} W M^{-1/2}$
Choose $v_1, v_2, \dots, v_k,$, the $k$ largest eigenvectors of $L$
$V = [v_1, \dots, v_k] \in \mathbf{R}^{n \times k}$
Normalize the rows of $V$ to generate $V' : v'_{ij} = v_{ij} / \|v_i\|$
for $i \leftarrow 1$ to $n$ do
    $i$th row of $V' : u_i \in \mathbf{R}^k$ Consider $u_i$ as $k$-dimensional
      embedding space corresponding to $o_i$
end
Construct new distance matrix $D' : d'_{ij} = \|u_i - u_j\|$
Apply VAT to $D'$ to obtain $\tilde{D}'$
```

Algorithm S4. Parallel VAT [29].

```
Input  : $D - n \times n$ dissimilarity matrix satisfying
             $-D_{ij} \geq 0$
             $-D_{ij} = D_{ji} \forall i, j$
             $-D_{ii} = 0 \forall i$
Output: $D^* - n \times n$ VAT reordered dissimilarity matrix
         $P -$ VAT reordering indices of D
         $d -$ Ordering of MST cut magnitudes

Compute weighted graph $G(V, E)$ with $n$ vertices
  in $V$ and edges in $E$, obtained from using $D$ as the
  adjacency matrix
$P = \{\}$
$S = V$
for each vertex $u$ in $V$ do
    for each vertex $v$ in $V$, such that $v \neq u$ do
        Find the minimum weighted edge from $u$ to $v$
    end
end
while no more vertices $u \in S$ can be merged do
    Merge vertices $(u \in U \subset S)$ to form connected
      components, called supervertices $(sv_U)$, using
      minimum weighted edges
    Treat supervertices as new vertices, $S = S \cup \{sv_U\} - U$
end
for each vertex $u$ in $S$ do
    Get the recursive ordering $O(u)$ of the subgraph in $u$
    $P \leftarrow P \cup O(u)$
end
for $p \leftarrow 1$ to $n$ do
    for $q \leftarrow 1$ to $n$ do
        $D^*_{p,q} = D_{P_p, P_q}$
    end
end
```

Algorithm S5. Dark Block Extraction [54].

```
Input  : $D - n \times n$ dissimilarity matrix
         $\alpha -$ minimum cluster size as a fraction of $n$
Output: $k -$ the number of dark blocks
Transform $D$ to $D'$ with $d_{ij} = 1 - e^{-d_{ij}/\sigma}$, where $\sigma$ is a
    scale parameter obtained by applying Otsu's
    algorithm [184] on $D$
Apply VAT to $D'$ to generate RDI $I^{(1)}$
Threshold $I^{(1)}$ using Otsu's algorithm [184] to generate
    binary image $I^{(2)}$
Filter $I^{(2)}$ using morphological operator with directional
    "line" structuring elements of size $l_1 = \alpha n$ to obtain
    filtered binary image $I^{(3)}$
Perform distance transformation on $I^{(3)}$ to obtain a
    new grayscale image $I^{(4)}$ and scale the pixel values
    to [0, 1]
Project pixel values of $I^{(4)}$ onto the main diagonal
    axis of the image to form a projection signal $H^{(1)}$
Smooth $H^{(1)}$ using simple average filter $h$ of length
    $l_2 = 2\alpha n$ to obtain the filtered signal $H^{(2)}$
Compute first-order derivative of $H^{(2)}$ to obtain
    signal $H^{(3)}$
Find peak position $p_i$ (positive-to-negative zero-crossing
    points) and valley position (negative-to-positive zero-
    crossing points) in $H^{(3)}$
Select major peaks by removing minor peaks and
    valleys using a filter size of length $l_3 = 2\alpha n$
$k =$ number of major peaks
```

Algorithm S6. Asymmetric iVAT [13].

```
Input  : $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} - N$ $p$-dimensional data points
         $k' -$ overestimate of actual number of clusters
         $n -$ approximating sample size
Output: $D'^*_n - n \times n$ iVAT reordered dissimilarity matrix
          of $D_n$
        $\tilde{S} -$ indices of samples in $D_n$
        $P -$ VAT reordering indices of $D_n$
        $d -$ ordering of MST cut magnitudes
Select the indices $m$ of $k'$ distinguished objects
$m_1 = 1$
$y = \{dist\{\mathbf{x}_1, \mathbf{x}_1\}, \ldots, dist\{\mathbf{x}_1, \mathbf{x}_N\}\}$
for $t \leftarrow 2$ to $k'$ do
    $y = (\min\{y_1, dist\{\mathbf{x}_{\mathbf{m}_{t-1}}, \mathbf{x}_1\}, \ldots,$
    $\min\{y_N, dist\{\mathbf{x}_{\mathbf{m}_{t-1}}, \mathbf{x}_N\}\})$
    $m_t = \underset{1 \le j \le N}{\operatorname{argmax}} \{y_j\}$
end
Group objects in $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ with their
  nearest distinguished objects
$S_1 = S_2 = \ldots = S_{k'} = \emptyset$
for $t \leftarrow 1$ to $N$ do
    $l = \underset{1 \le j \le k}{\operatorname{argmin}} \{dist\{\mathbf{x}_{\mathbf{m}_j}, \mathbf{x}_t\}\}$
    $S_l = S_l \cup \{t\}$
end
Randomly select data near each distinguished
    object to form $D_n$
for $t \leftarrow 1$ to $k'$ do
    $n_t = \left\lceil \frac{n \times |S_t|}{N} \right\rceil$
    Draw $n_t$ unique random indices $\tilde{S}_t$ from $S_t$
end
$\tilde{S} = \bigcup_{t=1}^{k'} \tilde{S}_t; D_n = dist\{\mathbf{x}_{\tilde{\mathbf{S}}}, \mathbf{x}_{\tilde{\mathbf{S}}}\}$
Apply VAT to $D_n$ returning $D^*_n$, $P$ and $d$
Apply iVAT to $D^*_n$ returning $D'^*_n$
```

Algorithm S7. Clustering Using Scalable iVAT (clusiVAT) [79]–[81].

```
Input  : $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} - N$ $p$-dimensional data points
         $k' -$ overestimate of actual number of clusters
         $n -$ approximating sample size
Output: $D'^*_n - n \times n$ iVAT reordered dissimilarity matrix of $D_n$
        $u : X \rightarrow \{1, 2, \ldots, k\} -$ cluster membership
Apply siVAT on $X$ returning $D'^*_n, \tilde{S}, P, d$
Choose the number of clusters $k$ using siVAT image
$t = \underset{1 \le i \le k}{\operatorname{argmax}}\, d_i$
Form the aligned partition:
$u^* = \{t_1 : t_2 - t_1 : \ldots : t_k - t_{k-1}\}$
$u_{\tilde{S}_{P_i}} = u^*_{P_i};\ 1 \le i \le k$
for $\hat{\mathbf{x}} \in \hat{X} = X - X_{\tilde{S}}$ do
    $j = \underset{i \in \tilde{S}}{\operatorname{argmin}} \{dist\{\mathbf{x}_{\hat{\mathbf{s}}}, \mathbf{x}_{\mathbf{i}}\}\}$
    $u_{\hat{s}} = u_j$
end (nearest prototype rule)
```

Algorithm S8. InsertPosition.

```
Input  : $P_n -$ VAT reordering indices of $D^*_n$
         $d_n -$ MST cut magnitude order of $D^*_n$
         $F_n -$ MST connection indices of $D^*_n$
         $V = \{v_1, v_2, \ldots, v_n\}$ Distance of $\mathbf{x}_{\mathbf{n+1}}$ from
           $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$
Output: $i -$ insertion position of $\mathbf{x}_{\mathbf{n+1}}$
        $P_{n+1} -$ initialization of VAT reordering indices of $D^*_{n+1}$
        $d_{n+1} -$ initialization of MST cut magnitude order
          of $D^*_{n+1}$
        $F_{n+1} -$ initialization of MST connection indices
          of $D^*_{n+1}$
$Y = V_{P_n} = \{v_{P_1}, v_{P_2}, \ldots, v_{P_n}\}$
$i = n + 1$
$j = \operatorname{argmin}(V)$
for $t \leftarrow 1$ to $n - 1$ do
    if $\min(\{Y_1, Y_2, \ldots, Y_t\}) < d_{n_t}$ then
        $i = t + 1$
        $j = \operatorname{argmin}(\{Y_1, Y_2, \ldots, Y_t\})$
        break
    end
end
$P_{n+1} = \{P_{n_1}, P_{n_2}, \ldots, P_{n_{i-1}}, n + 1\}$
$d_{n+1} = \{d_{n_1}, d_{n_2}, \ldots, d_{n_{i-2}}, \min(Y_1, Y_2, \ldots, Y_{i-1})\}$
$F_{n+1} = \{F_{n_1}, F_{n_2}, \ldots, F_{n_{i-1}}, j\}$
```

Algorithm S9. Incremental VAT (inc-VAT).

```
Input  : $D'^{*}_{n} - n \times n$ VAT reordered dissimilarity matrix for
           $X_n,\ n \geq 2$
         $P_n$ – VAT reordering indices of $D^{*}_{n}$
         $d_n$ – MST cut magnitude order of $D^{*}_{n}$
         $F_n$ – MST connection indices of $D^{*}_{n}$
         $V = \{v_1, v_2, \ldots, v_n\}$ – distance of $\mathbf{x_{n+1}}$ from
           $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\}$
Output: $D^{*}_{n+1} - (n+1) \times (n+1)$ VAT reordered dissimilarity
           matrix for $X_{n+1}$
         $P_{n+1}$ – VAT reordering indices of $D^{*}_{n+1}$
         $d_{n+1}$ – MST cut magnitude order of $D^{*}_{n+1}$
         $F_{n+1}$ – MST connection indices of $D^{*}_{n+1}$

Find the insertion position $i$ for $\mathbf{x_{n+1}}$
$(i, P_{n+1}, d_{n+1}, F_{n+1}) = \text{InsertPosition}(P_n, d_n, F_n, V)$
                                                    (algorithm S8)
$A = \{P_{n_i}, P_{n_{i+1}}, \ldots, P_{n_n}\}$
$C = \{P_{n_1}, P_{n_2}, \ldots, P_{n_{i-1}}\}$
$B = P_n \setminus C$
$E = \emptyset$
$G = \{1, 2, \ldots, i-1\}$
$H = \{F_{n_i}, F_{n_{i+1}}, \ldots, F_{n_n}\}$
Reorder the remaining points after insertion
  position
while $A \neq \emptyset$ do
    $(A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H) =$
      $\text{IncInsert}(A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H, D^{*}_{n}, P_n, d_n, F_n)$
                                                    (algorithm S10)
end
$D^{*}_{n} = D^{*}_{n_{G,G}}$
$Y^{*} = Y_G$
$Y^{*} = [Y^{*}_{1}, Y^{*}_{2}, \ldots, Y^{*}_{i-1}, 0, Y^{*}_{i}, Y^{*}_{i+1}, \ldots, Y^{*}_{n}]$
$D^{*}_{n+1} =$ Insert $Y^{*}$ after $i-1$th row and $i-1$th column
  of $D^{*}_{n}$
```

Algorithm S10. IncInsert.

```
Input  : $D^{*}_{n} - n \times n$ VAT reordered dissimilarity matrix for $X_n$
         $P_n$ – VAT reordering indices of $D^{*}_{n}$
         $d_n$ – MST cut magnitude order of $D^{*}_{n}$
         $F_n$ – MST connection indices of $D^{*}_{n}$
Input—Output: $A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H$
$\left.\begin{array}{l} z_1 = d_{n_{Pos(B_1)-1}} \\ w_1 = B_1 \\ v_1 = \arg(P_{n+1} = P_{n_{H_1}}) \end{array}\right\}$ Minimum distance, closest point index, and MST connection index from G1
$\left.\begin{array}{l} z_2 = \min(Y_A) \\ w_2 = A_{\text{argmin}(Y_A)} \\ v_2 = i \end{array}\right\}$ Minimum distance, closest point index, and MST connection index from G2
$\left.\begin{array}{l} z_3 = \min(D^{*}_{n_{Pos(A),Pos(E)}}) \\ (j, k) = \text{argmin}(D^{*}_{n_{Pos(A),Pos(E)}}) \\ w_3 = A_j \\ v_3 = \arg(P_{n+1} = E_k) \end{array}\right\}$ Minimum distance, closest point index, and MST connection index from G3
$z = \min(z_1, z_2, z_3)$
switch $z$ do
    case $z_1$ do
        $(A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H) =$
          $\text{M1}(A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H, z_1, w_1, v_1)$
                                                    (algorithm S12)
    end
    case $z_2$ do
        $(A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H) =$
          $\text{M2}(A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H, z_2, w_2, v_2)$
                                                    (algorithm S13)
    end
    case $z_3$ do
        $(A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H) =$
          $\text{M3}(A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H, z_3, w_3, v_3)$
                                                    (algorithm S14)
    end
end
```

Algorithm S11. LongestSubsequence

```
Input  : $P_n$ – VAT reordering indices of $D^{*}_{n}$
         $P_{n+1}$ – VAT reordering indices of $D^{*}_{n+1}$
Output: LongestSubsequence $(P_n, P_{n+1})$

for $j \leftarrow 1$ to $n$ do
    if $P_{n_j} \notin P_{n+1}$ then
        break
    end
end
LongestSubsequence $(P_n, P_{n+1}) = \{P_{n_1}, P_{n_2}, \ldots, P_{n_{j-1}}\}$
```

Algorithm S12. Procedure M1.

```
Input  : $z_1$ – minimum distance from G1
         $w_1$ – closest point index from G1
         $v_1$ – MST connection index from G1
Input—Output: $A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H$
$P_{n+1} = \{P_{n+1}, w_1\};\ d_{n+1} = \{d_{n+1}, z_1\};\ F_{n+1} = \{F_{n+1}, v_1\}$
$G = \{G, Pos(w_1)\}$
$C =$ LongestSubsequence $(P_n, P_{n+1})$        (algorithm S11)
$B = P_n \setminus C$
while $length(H) > length(B)$ do
    delete $H_1$
end
delete $w_1$ from $A$
```

Algorithm S13. Procedure M2.

```
Input  : $z_2$ – minimum distance from G2
         $w_2$ – closest point index from G2
         $v_2$ – MST connection index from G2
```

```
Input-Output: $A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H$
$P_{n+1} = \{P_{n+1}, w_2\}$;  $d_{n+1} = \{d_{n+1}, z_2\}$;  $F_{n+1} = \{F_{n+1}, v_2\}$
$G = \{G, Pos(w_2)\}$

if $w_2 = A_1$ then
    $C =$ LongestSubsequence $(P_n, P_{n+1})$ (algorithm S11)
    $B = P_n \setminus C$
    while $length(H) > length(B)$ do
        delete $H_1$
    end
    $E = P_{n+1} \setminus \{n+1\} \setminus C$
else
    $E = \{E, w_2\}$
end

delete $w_2$ from $A$
```

Algorithm S14. Procedure M3.

```
Input  : $z_3$ – minimum distance from G3
         $w_3$ – closest point index from G3
         $v_3$ – MST connection index from G3

Input—Output: $A, B, E, P_{n+1}, d_{n+1}, F_{n+1}, G, H$
$P_{n+1} = \{P_{n+1}, w_3\}$;  $d_{n+1} = \{d_{n+1}, z_3\}$;  $F_{n+1} = \{F_{n+1}, v_3\}$
$G = \{G, Pos(w_3)\}$

if $w_3 = A_1$ then
    $C =$ LongestSubsequence $(P_n, P_{n+1})$ (algorithm S11)
    $B = P_n \setminus C$
    while $length(H) > length(B)$ do
        delete $H_1$
    end
    $E = P_{n+1} \setminus \{n+1\} \setminus C$
else
    $E = \{E, w_3\}$
end

delete $w_3$ from $A$
```

Algorithm S15. Incremental iVAT (inc-iVAT).

```
Input  : $D^*_{n+1} - (n+1) \times (n+1)$ inc-VAT reordered
           dissimilarity matrix for $X_{n+1}$
         $D'^*_n - n \times n$ iVAT dissimilarity matrix for $X_n$
         $i$ – insertion index of the new data point $\mathbf{x_{n+1}}$
           in $P_{n+1}$
Output: $D'^*_{n+1} - (n+1) \times (n+1)$ inc-iVAT dissimilarity
           matrix for $X_{n+1}$

$c = \{1, 2, \ldots, i-1\}$
$D'^*_{n+1_{cc}} = D'^*_{n_{cc}}$

for $r \leftarrow i$ to $n+1$ do
    $j = \underset{1 \le m \le r-1}{\text{argmin}} \{D^*_{n+1_{rm}}\}$
    $D'^*_{n+1_{rj}} = D^*_{n+1_{rj}}$
    $c = \{1, 2, \ldots, r-1\} \setminus \{j\}$
    $D'^*_{n+1_{rc}} = \max\{D^*_{n+1_{rj}}, D'^*_{n+1_{jc}}\}$
    $D'^*_{n+1_{rc}} = D'^*_{n+1_{cr}}$
end
```

Algorithm S16. Decremental VAT (dec-VAT).

```
Input  : $D^*_n - n \times n$ VAT reordered dissimilarity matrix for $X_n$
         $P_n$ – VAT reordering indices of $D^*_n$
         $d_n$ – MST cut magnitude order of $D^*_n$
         $F_n$ – MST connection indices of $D^*_n$
         $\mathbf{x_p}$ – point to remove
Output: $D^*_{n-1} - (n-1) \times (n-1)$ VAT reordered dissimilarity
           matrix for $X_{n-1}$
         $P_{n-1}$ – VAT reordering indices of $D^*_{n-1}$
         $d_{n-1}$ – MST cut magnitude order of $D^*_{n-1}$
         $F_{n-1}$ – MST connection indices of $D^*_{n-1}$

Find the position $i$ of $\mathbf{x_p}$ in $P_n$
$i = \arg(P_n = p)$

Find the data points $(J)$ and their indices $(I)$ in
  $P_n$, that are connected to $\mathbf{x_p}$ in the MST of $D^*_n$
$I = \arg(F_n = i)$;  $J = P_{n_I}$

if $J = \emptyset$ then
    $(D^*_{n-1}, P_{n-1}, d_{n-1}, F_{n-1}) =$ LeafNodeRemove$(D^*_n, P_n,$
      $d_n, F_n, i)$                                  (algorithm S17)
else
    if $i = P_{n_1}$ then
        $P_{n-1} = \{P_{n_2}\}$; $d_{n-1} = \emptyset$; $F_{n-1} = \{1\}$
        $A = \{P_{n_3}, P_{n_4}, \ldots, P_{n_n}\}$
        $C = \{P_{n_1}, P_{n_2}\}$
        $B = P_n \setminus C$
        $E = \emptyset$
        $G = \{2\}$
        $H = \{F_{n_3}, F_{n_4}, \ldots, F_{n_n}\}$
        $k = \arg(J = P_{n_2})$; delete $J_k$
    else
        $P_{n-1} = \{P_{n_1}, P_{n_2}, \ldots, P_{n_{i-1}}\}$;  $d_{n-1} = \{d_{n_1}, d_{n_2}, \ldots, d_{n_{i-2}}\}$;
        $F_{n-1} = \{F_{n_1}, F_{n_2}, \ldots, F_{n_{i-1}}\}$
        $A = \{P_{n_{i+1}}, P_{n_{i+2}}, \ldots, P_{n_n}\}$
        $C = \{P_{n_1}, P_{n_2}, \ldots, P_{n_i}\}$
        $B = P_n \setminus C$
        $E = \emptyset$
        $G = \{1, 2, \ldots, i-1\}$
        $H = \{F_{n_{i+1}}, F_{n_{i+2}}, \ldots, F_{n_n}\}$
    end

    while $A \neq \emptyset$ do
        if $B_1 = J_1$ then
            $(A, B, E, P_{n-1}, d_{n-1}, F_{n-1}, G, H, J) =$
              SpecialInsert$(A, B, E, P_{n-1}, d_{n-1}, F_{n-1}, G, H, J,$
                          $D^*_n, P_n, d_n, F_n)$ (algorithm S18)
        else
            $(A, B, E, P_{n-1}, d_{n-1}, F_{n-1}, G, H, ) =$
              DecInsert$(A, B, E, P_{n-1}, d_{n-1}, F_{n-1}, G, H, D^*_n, P_n,$
                      $d_n, F_n)$                 (algorithm S19)
        end
    end
    $D^*_{n-1} = D^*_{n_{G,G}}$
end
```

Algorithm S17. LeafNodeRemove.

```
Input  : $D^*_n - n \times n$ VAT reordered dissimilarity matrix for $X_n$
         $P_n -$ VAT reordering indices of $D^*_n$
         $d_n -$ MST cut magnitude order of $D^*_n$
         $F_n -$ MST connection indices of $D^*_n$
         $i -$ Position of leaf node $\mathbf{x_p}$ in $P_n$
Output: $D^*_{n-1} - (n-1) \times (n-1)$ VAT reordered dissimilarity
          matrix for $X_{n-1}$
         $P_{n-1} -$ VAT reordering indices of $D^*_{n-1}$
         $d_{n-1} -$ MST cut magnitude order of $D^*_{n-1}$
         $F_{n-1} -$ MST connection indices of $D^*_{n-1}$

Initialize $P_{n-1}, F_{n-1}, d_{n-1}$, and $D^*_{n-1}$
$P_{n-1} = P_n$
$F_{n-1} = F_n$
$d_{n-1} = d_n$
$D^*_{n-1} = D^*_n$

Delete the elements corresponding to $\mathbf{x_p}$
delete $P_{{n-1}_i}$
delete $F_{{n-1}_i}$
$K = \arg(F_{n-1} > i)$
$F_{{n-1}_K} = F_{{n-1}_K} - 1$
delete $d_{{n-1}_{i-1}}$
delete $i$th row and $i$th column of $D^*_{n-1}$
```

Algorithm S18. SpecialInsert.

```
Input  : $D^*_n - n \times n$ VAT reordered dissimilarity matrix for $X_n$
         $P_n -$ VAT reordering indices of $D^*_n$
         $d_n -$ MST cut magnitude order of $D^*_n$
         $F_n -$ MST connection indices of $D^*_n$

Input—Output: $A, B, E, P_{n-1}, d_{n-1}, F_{n-1}, G, H, J$

$z = \min(D^*_{n_{G,Pos(A)}})$
$(j, k) = \operatorname{argmin}(D^*_{n_{G,Pos(A)}})$
$w = A_j$
$v = \arg(P_{n-1} = G_k)$
$P_{n-1} = \{P_{n-1}, w\}$
$d_{n-1} = \{d_{n-1}, z\}$
$F_{n-1} = \{F_{n-1}, v\}$
$G = \{G, Pos(w)\}$

if $w \in J$ then
    delete $w$ from $A$
    if $w = J_1$ then
        $C = \text{LongestSubsequence}(P_n, \{P_{n-1}, i\})$
                                        (algorithm S11)
        $B = P_n \setminus C$
        while $length(H) > length(B)$ do
            delete $H_1$
        end
    end
    delete $w$ from $J$
else
    $E = \{E, w\}$
end
```

Algorithm S19. DecInsert.

```
Input  : $D'^*_n - n \times n$ VAT reordered dissimilarity matrix for $X_n$
         $P_n -$ VAT reordering indices of $D^*_n$
         $d_n -$ MST cut magnitude order of $D^*_n$
         $F_n -$ MST connection indices of $D^*_n$
Input—Output: $A, B, E, P_{n-1}, d_{n-1}, F_{n-1}, G, H$

$z_1 = d_{n_{Pos(B_1)-1}}$                      } Minimum distance, closest
$w_1 = B_1$                                     }   point index, and MST
$v_1 = \arg(P_{n-1} = P_{n_{H_1}})$             } connection index from G1

$z_3 = \min(D^*_{n_{Pos(A),Pos(E)}})$            } Minimum distance,
$(j, k) = \operatorname{argmin}(D^*_{n_{Pos(A),Pos(E)}})$ } closest point index,
$w_3 = A_j$                                     } and MST connection
$v_3 = \arg(P_{n-1} = E_k)$                     }   index from G3

$z = \min(z_1, z_3)$

switch $z$ do
    case $z_1$ do
        $(A, B, E, P_{n-1}, d_{n-1}, F_{n-1}, G, H) =$
         $\text{M1}(A, B, E, P_{n-1}, d_{n+1}, F_{n-1}, G, H, z_1, w_1, v_1)$
                                        (algorithm S12)
    end
    case $z_2$ do
        $(A, B, E, P_{n-1}, d_{n-1}, F_{n-1}, G, H) =$
         $\text{M3}(A, B, E, P_{n-1}, d_{n-1}, F_{n-1}, G, H, z_3, w_3, v_3)$
                                        (algorithm S14)
    end
end
```

Algorithm S20. Decremental iVAT (dec-iVAT).

```
Input  : $D^*_{n-1} - (n-1) \times (n-1)$ dec-VAT reordered
          dissimilarity matrix for $X_{n-1}$
         $D'^*_n - n \times n$ iVAT dissimilarity matrix for $X_n$
         $i -$ insertion index of the new data point $\mathbf{x_p}$
          in $P_n$
Output: $D'^*_{n-1} - (n-1) \times (n-1)$ inc-iVAT dissimilarity
          matrix for $X_{n-1}$

$c = \{1, 2, \ldots, i-1\}$
$D'^*_{{n-1}_{cc}} = D'^*_{n_{cc}}$

for $r \leftarrow i$ to $n-1$ do
    $j = \operatorname{argmin}_{1 \le m \le r-1} \{D^*_{{n-1}_{rm}}\}$
    $D'^*_{{n-1}_{rj}} = D^*_{{n-1}_{rj}}$
    $c = \{1, 2, \ldots, r-1\} \setminus \{j\}$
    $D'^*_{{n-1}_{rc}} = \max\{D^*_{{n-1}_{rj}}, D'^*_{{n-1}_{jc}}\}$
    $D'^*_{{n-1}_{rc}} = D'^*_{{n-1}_{cr}}$
end
```

Algorithm S21. siVAT+ [78].

```
Input  : $X=\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}\subset\mathbb{R}^p$ – High dimensional big data
         $k'$ – Overestimate of true number of clusters
         $n$ – Number of approximated samples
         $\alpha$ – A fraction of $N$, $0<\alpha<1$
Output: $D'^{*}_{n\times n}$ - siVAT + reordered dissimilarity matrix

% % MMRS+ Sampling % %
// Select the indices $m$ of $k'$ distinguished MM objects
$M=\emptyset$
Randomly select the initial object, $\mathbf{x}_{\mathbf{m_0}}\in\mathbf{X}$
$M\leftarrow M\cup\{x_{m0}\}$
for $i=2$ to $k'$ do
    Randomly generate $n_s=\lceil\alpha N\rceil$ sized sample $Y$ from $X$
        $Y^{(i)}=\{\mathbf{y}_1^{(i)},\ldots,\mathbf{y}_{n_s}^{(i)}\}$
    Compute distance of each data point of sample $Y^{(i)}$ to the closest element of existing $M$, $z^{(i)}=(\min\{r_{m_11}^{(i)},r_{m_21}^{(i)},\ldots,r_{m_i1}^{(i)}\},$
     $\min\{r_{m_12}^{(i)},r_{m_22}^{(i)},\ldots,r_{m_i2}^{(i)}\},\ldots,\ \min\{r_{m_1n_s}^{(i)},r_{m_2n_s}^{(i)},\ldots,r_{m_in_s}^{(i)}\})=(z_1^{(i)},\ldots,z_{n_s}^{(i)})$
     $m_i={}^{*}\mathrm{argmax}_{1\le j\le n_s} z_j^{(i)}$  // Choose the farthest (distinguished) object in $Y^{(i)}$ from existing $M$
     $M\leftarrow M\cup\{x_{(mi)}\}$  // Update $M$ with new distinguished object $x_{(mi)}$, where $(m_i)=$ index of $y_{m_i}^{(i)}$ in $X$
end
// Group $n_s$ random objects in $O$ with its nearest distinguished object
Randomly generate $n_s$ sized (intermediate) sample $S_{int}$
Compute $w_{it}$ – distance of each distinguished object $m_i\in M$ to each data point $\in S_{int}$
$Z_1=Z_2\ldots.=Z_{k'}=\emptyset$
for $t=1$ to $n_s$ do
     $l={}^{*}\mathrm{argmin}_{1\le i\le k'}\{w_{it}\}$
     $Z_l=Z_l\cup\{t\}$
end
// Randomly select data from $S_{int}$ near each distinguished point to obtain the $n$ number of samples
$n_i=\lceil n\times|Z_i|/n_s\rceil$  $i=1,2,\ldots,k'$
Draw $n_i$ unique random indices from $Z_i$ to build sample $Z_i'$
$S=\bigcup_{i=1}^{k'} Z_i'$
%% Cluster Assessment %%
Compute $D_n=dist(x_S,x_S)$
Apply VAT/iVAT on $D_n$, returning $D_n'^{*},P,c$ (MST edge cuts)
Choose the number of clusters $k$ using image of $D_n'^{*}$
```

Algorithm S22. Fast Ensemble siVAT (FensiVAT) [91].

```
Input  : $X=\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}\subset\mathbb{R}^p$ – High dimensional big data
         $q$ – down-space dimension
         $k'$ – Overestimate of true number of clusters
         $n$ – Number of approximated samples
         $Q$ – number of random projections (RPs)
Output: $D'^{*}_{n\times n}$ - FensiVAT reordered dissimilarity matrix

% % data set generation in down space % %
Generate down-space data sets $Y\subset\mathbb{R}^{N\times q}$ using $Y=(1/\sqrt{q})XT$, where $T\in\mathbb{R}^{p\times q}$ is a random matrix
% % Near-MMRS Sampling: MMRS on $Y$ % %
Randomly select the first distinguished object $x_{m_0}$
Distance of $x_{m_0}$ from $N$ points, $z=(z_1,\ldots z_N)=\{dist(x_{m_0},x_1),\ldots,dist(x_{m_0},x_N)\}=(r_{m_01},\ldots.r_{m_0N})$
Calculate the distance of $x_{m_j}$ from $N$ points, $(r_{m_j1},\ldots.r_{m_jN})$
for $i=1$ to $k'$ do
     $z\leftarrow(\min\{z_1,r_{m_{i-1}1}\},\ldots,\min\{z_N,r_{m_{i-1}N}\})$
     $m_i={}^{*}\mathrm{argmax}_{1\le j\le N}\, z_j$
end
Group each object in $O$ with its nearest distinguished object
$Z_1=Z_2\ldots.=Z_{k'}=\emptyset$
```

```
for $t=1$ $N$ do
    $l={}^{*}\text{argmin}_{1\le i\le k'}\{r_{m_i t}\}$
    $Z_l=Z_l\cup\{t\}$
end
*Randomly select data near each distinguished point to obtain the $n$ number of samples*
$n_i=\lceil n\times|Z_i|/N\rceil \quad i=1,2,\ldots,k'$
Draw $n_i$ unique random indices from $Z_i$ to build sample $Z_i'$
$S_d=\bigcup_{i=1}^{k'} Z_i'$
**% % Ensemble method to obtain a reliable iVAT image % %**
Generate $Q$, down-space data sets $\{S_{d,i}\}_{i=1}^{Q}\subset\mathbb{R}^q$ from $S_u\subset\mathbb{R}^p$ $(S_d\rightarrow S_u)$, using random matrices $\{T_i\}_{i=1}^{Q}\in\mathbb{R}^{q\times q}$,
    $|S_u|=|S_d|=n$
Compute distance matrices $\{D_{d,i}\}_{i=1}^{Q}$ from $\{S_{d,i}\}_{i=1}^{Q}$
$D_{n,d}\leftarrow 0$ (Initialize a $n\times n$ distance matrix)
for $t=1$ $Q$ do
    $W_i=NormalizeRows(D_{d,i})$
    $V_i=1/2(W_i+W_i^T)$
    $D_{n,d}=D_{n,d}+V_i$
end
Apply VAT/iVAT on $D_{n,d}$ returning $D'^{*}_{n,d}, P, c$
Choose the number of clusters $k$ using image of $D'^{*}_{n,d}$
**% % Clustering % %**
Find indices $u$ of $k$ largest values in MST cut magnitudes $c$
Form the aligned partition, $U^{*}=\{u_1:u_2-u_1:\ldots:u_k-u_{k-1}\}$, $U_{S_u}=U^{*}_{P_i}$, $1\le i\le k$
Generate down-space data sets $\{Y_i\}_{i=1}^{Q}\subset\mathbb{R}^q$ using RP, $|Y_i|=N$
for each $Y_i$ do
    Consider sample $Y_{S_d}^{(i)}\subset\mathbb{R}^q$ and $Y_i-Y_{S_d}^{(i)}\subset\mathbb{R}^q$, where $|Y_{S_d}^{(i)}|=n$, and $|Y_i-Y_{S_d}^{(i)}|=N-n$ for *each data point*, $\hat{y}\in Y_i-Y_{S_d}^{(i)}$ do
        $l=\text{argmin}_{i\in S_d}\{dist\{\hat{y},y_i\}\}$
        $U_{\hat{y}}^{(i)}=U_l$
    end
end
$U=$ Mode of labels for each data points $U_{\hat{y}}^{(i)}$
```

Algorithm S23. Coclustering VAT (coVAT) [94].

Input : $D-m\times n$ rectangular dissimilarity matrix
Output: $D^{*},\hat{D}_r,\hat{D}_c$, and $\hat{D}_{r\cup c}-$ reordered dissimilarity matrices

```
Build estimates of $D_r$ and $D_c$ by interpreting the $m$ rows and $n$ columns of $D$ as the feature vector representing $m$ row
    objects and $n$ column objects, respectively
$D_{r\cup c}=\begin{bmatrix} D_r & D \\ D^T & D_c \end{bmatrix}$
Apply VAT to $D_{r\cup c}$ generating permutation array $P_{r\cup c}=\{P(1),P(2)\ldots,P(m+n)\}$
Initialize $rc=cc=0$; $RP=CP=0$
for $t\leftarrow 1$ to $m+n$ do
    if $P(t)\le m$ then
        $rc=rc+1, rc$ is row component
        $RP(rc)=P(t), RP$ are row indices
    else
        $cc=cc+1, cc$ is column component
        $CP(cc)=P(t)-m, CP$ are column indices
    end
end
for $p\leftarrow 1$ to $m$ do
    for $q\leftarrow 1$ to $n$ do
        $D^{*}_{p,q}=D_{RP_p,CP_q}$
    end
end
Create $I(D^{*})$
```

Algorithm S24. coVAT2 [95], [96].

```
Input  : $D - m \times n$ rectangular dissimilarity matrix
Output: $D^{**}, \hat{D}_r$, and $\hat{D}_c$ – Reordered dissimilarity
         matrices

Build estimates of $D_r$ and $D_c$ by interpreting the
  $m$ rows and $n$ columns of $D$ as the feature vector
  representing $m$ row objects and $n$ column objects
  respectively
Apply VAT to $D_r$ generating permutation array
  $RP = \{RP(1), P(2) \ldots, P(m)\}$
Apply VAT to $D_c$ generating permutation array
  $CP = \{CP(1), C(2) \ldots, C(n)\}$
for $p \leftarrow 1$ to $m$ do
    for $q \leftarrow 1$ to $n$ do
        $D^{**}_{p,q} = D_{RP_p, CP_q}$
    end
end
Create $I(D^{**})$
```

Algorithm S25. Scalable coVAT (scoVAT) [98].

```
Input  : $D - m \times n$ rectangular dissimilarity matrix, $M$ and
         $N$ are large
         $c'$ – An overestimate of the true but unknown
         number of clusters c
         $m$ – Row sample size
         $n$ – column sample size
Output: $D^{*}_{m \times n}$ – scoVAT reordered dissimilarity matrix

Build estimates of $[D_r]_{M \times M}$ and $[D_c]_{N \times N}$ by interpreting
  the $M$ rows and $N$ columns of $D$ as the feature vector
  representing $M$ row objects and $N$ column objects
  respectively
Apply siVAT on $[D_r]_{M \times M}$ returning $m$ sampled rows
Apply siVAT on $[D_c]_{N \times N}$ returning $n$ sampled rows
Build $D_{m \times n}$ by extracting $m$ sampled rows and $n$ sampled
  columns from $D_{M \times N}$
Apply coVAT on $D_{m \times n}$
```

Visual Assessment of Clustering Tendency

The VAT algorithm, introduced in [11], is a method for visually assessing the clustering tendency in a set of objects $O = \{o_1, o_2, \ldots, o_n\}$, whether they are represented by object feature vectors or numerical pairwise dissimilarity values. The input matrix D, generally any dissimilarity matrix, but usually a pairwise distance matrix built from vector data inputs, is reordered to obtain D^* using a modified Prim's algorithm. The image $I(D^*)$, when displayed as a grayscale image, shows possible clusters as dark blocks along the diagonal. The pseudocode for VAT is given in algorithm S1 in "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family."

Our first example replicates part of [13, Fig. 2]. Figure 1(a) is a scatterplot of three subsets, X_1, X_2, and X_3, drawn from Gaussian distributions centered at (0, 0), (3, 4), and (6, 0), with cardinalities of $|X_1| = 750$, $|X_2| = 1{,}750$, $|X_3| = 2{,}500$, respectively. All three distributions had the same covariance matrix:

$$\Sigma_i = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

Figure 1(b) is the VAT image of the Euclidean distance matrix D_E, whose ijth entry is the Euclidean distance $d_{E,ij} = \|x_i - x_j\|$. The structure of the data in Figure 1(a) is represented in Figure 1(b) by the three dark diagonal blocks. For the three compact, well-separated clusters in X, the sizes of the three blocks in the VAT image correspond exactly to the cardinalities of the three subsets, beginning with the largest block for X_3 at the top, X_2 in the middle, and X_1 at the bottom. This shows how the VAT image can suggest both the number and sizes of clusters in the data. In addition, the VAT image in Figure 1(b) depends on the choice of distance used to build the input dissimilarity matrix. A change in the metric might result in a different VAT image for the same data set, which explains the explicit notation $I(D^*_E(X))$ for the image in Figure 1(b). However, we will usually use Euclidean distance, so we suppress the set X and subscript E unless clarity demands it.

Improvements in the VAT RDI for Complex Cluster Structure and Noisy Data

Although VAT can often provide a useful estimate of the number of clusters in a data set that have globular compact separated clusters, the VAT image can often be inconclusive, especially if the cluster structure in the data set is complex. To illustrate this point, Figure 2(a) and (b) shows two data

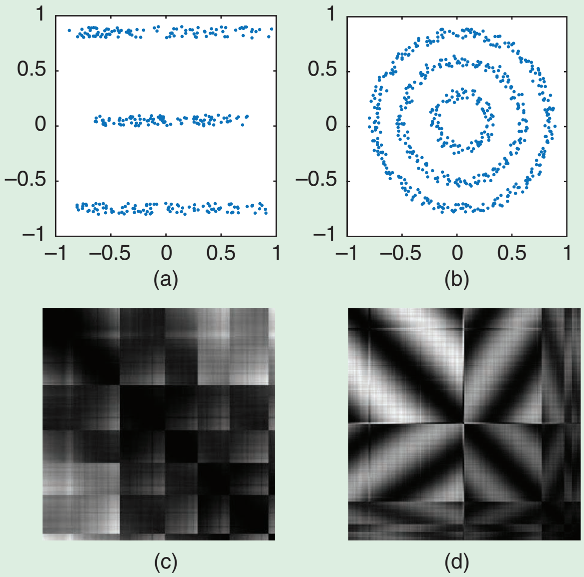


Figure 2. The data scatterplots and VAT images for the two data sets: (a) a three-line data set, (b) a three-ring data set, (c) $I(D^*)$ for the three-line data set, and (d) $I(D^*)$ for the three-ring data set.

sets consisting of data points along three parallel lines and three concentric circles. Most human observers would agree that the three lines and rings in Figure 2(a) and (b) constitute three clusters. Their VAT images are shown in Figure 2(c) and (d), respectively, and neither image gives any indication about the presence of three clusters in these two data sets. The problem for VAT images of these two data sets is that the clusters are not “clouds” of points; rather, they are “stringy.” To alleviate this problem, a variety of improvements have been proposed in the literature. These fall into two major categories: one uses graph-based distances, and the other involves spectral graph theory. These two approaches are discussed next.

Wang et al. [14] proposed an improved VAT (iVAT) method to enhance the RDI generated by the VAT algorithm by transforming the input dissimilarity matrix with a path-based distance measure. The path-based dissimilarity measure is based on the idea that, if two objects o_i and o_j are very far from each other (reflected by a large distance value d_{ij}), but there is a path connecting them through a sequence of other objects, such that the distances between any two successive objects are small, then d_{ij} should be adjusted to a smaller value to reflect this connection. This adjustment reflects the idea that, no matter how far the distance between two objects may be, they should be considered as coming from one cluster if they are connected by a set of successive objects forming dense regions (reflecting the characteristic of elongated clusters).

An efficient formulation of the iVAT algorithm (based on recursion), which significantly reduces its computational complexity from $O(n^3)$ (for the iVAT implementation presented in [14]) to $O(n^2)$ was proposed by Havens and Bezdek [15]. Their iVAT implementation begins by first finding the VAT reordered dissimilarity matrix D^* and then transforming the input distances in the distance matrix $D^* = [d^*_{ij}]$ by path-based minimax distances $D'^* = [d'^*_{ij}]$, given by

$$d'^{*}_{ij} = \min_{p \in P_{ij}} \max_{1 \le h \le |p|} D^{*}_{p[h]p[h+1]}, \tag{1}$$

where P_{ij} is the set of all paths from object $i(o_i)$ to object $j(o_j)$ in the VAT-generated MST of O. This formulation is not only computationally efficient, but it also retains a direct relationship between the VAT and iVAT images, thus making it feasible to directly extract single-linkage (SL) clusters from the iVAT image. This improved version of VAT now appears in almost all of the literature based on algorithms in the VAT family. The pseudocode for iVAT is given in algorithm S2 of “Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family.”

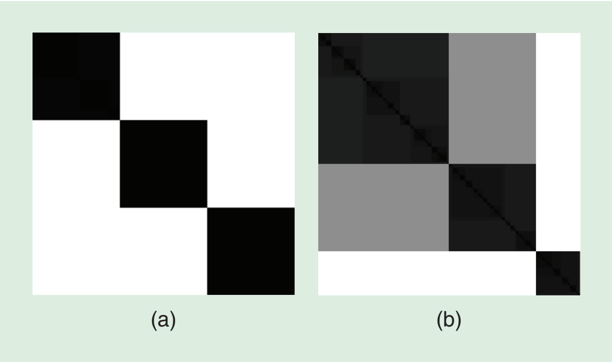


Figure 3. The iVAT images for the two data sets in Figure 2(a) and (b): (a) $I(D'^*)$ for the three-line data set and (b) $I(D'^*)$ for the three-ring data set.

Figure 3 shows the iVAT images of the two data sets shown in Figure 2(a) and (b). Both iVAT images give a clear and accurate portrayal of the structure of the input data and suggest the presence of three clusters by three dark blocks along the diagonal. Again, the sizes of the three dark blocks indicate the relative sizes of the three clusters in the data.

As opposed to iVAT, which first uses the VAT algorithm on the raw distance matrix and then uses a graph-based distance to improve the RDI quality, an alternate approach was employed in [16] and [17] that first transforms the raw distance matrix before feeding it to the VAT algorithm. Markov random-field VAT (MrfVAT), proposed in [16], modifies the input-distance matrix using Markov random fields, which updates each object with its local information dynamically and maximizes a global probability measure.

Similarly, the approach presented in [17] takes a refined co-association matrix, which was originally used in ensemble clustering, as an initial similarity matrix and transforms it by a path-based measure before applying it to VAT. These methods can deal with data sets that have complex cluster structures (where VAT is likely to fail) and can reveal the relationship of clusters hierarchically. The MrfVAT images of the two complex data sets shown in Figure 2(a) and (b) are similar to the iVAT images shown in Figure 3. Since the differences between the iVAT and MrfVAT images for the three-line and three-ring data sets are indistinguishable to the human eye, they are not included in this article.

Another set of algorithms that improve VAT-generated RDIs are based on spectral graph theory. The spectral VAT (SpecVAT) algorithm [18] addresses the limitation of VAT in highlighting the complex cluster structure present in a data set by first mapping the raw distance matrix D to a graph embedding space D' before reordering it by using the VAT algorithm. SpecVAT first converts D to a weighted affinity matrix and then performs spectral decomposition of the normalized Laplacian of the weighted affinity matrix. It then transforms the original feature vector representation of data points using the k largest eigenvectors. The VAT algorithm applied to the transformed representation of the data points produces RDIs that have much sharper contrast between dark blocks along the diagonal and the remaining pixels in the image. The output of the SpecVAT algorithm is a set of images (corresponding to different values of k) from which we can visually choose a “best” SpecVAT image in terms of clarity and block structure.

One way to choose the best from among SpecVAT images is to consider their grayscale histograms, which should include two explicit modalities in the distributions, corresponding to within-cluster distances (diagonal dark-block regions) and between-cluster distances (off-diagonal non-dark-block regions). The ideal scenario would have a narrow distribution for each modality and a large distance between the two modalities. Figure 4 shows the SpecVAT images and their grayscale histograms for the three-ring data set [Figure 2(b)] for $k = \{1, 2, \ldots, 5\}$. The SpecVAT

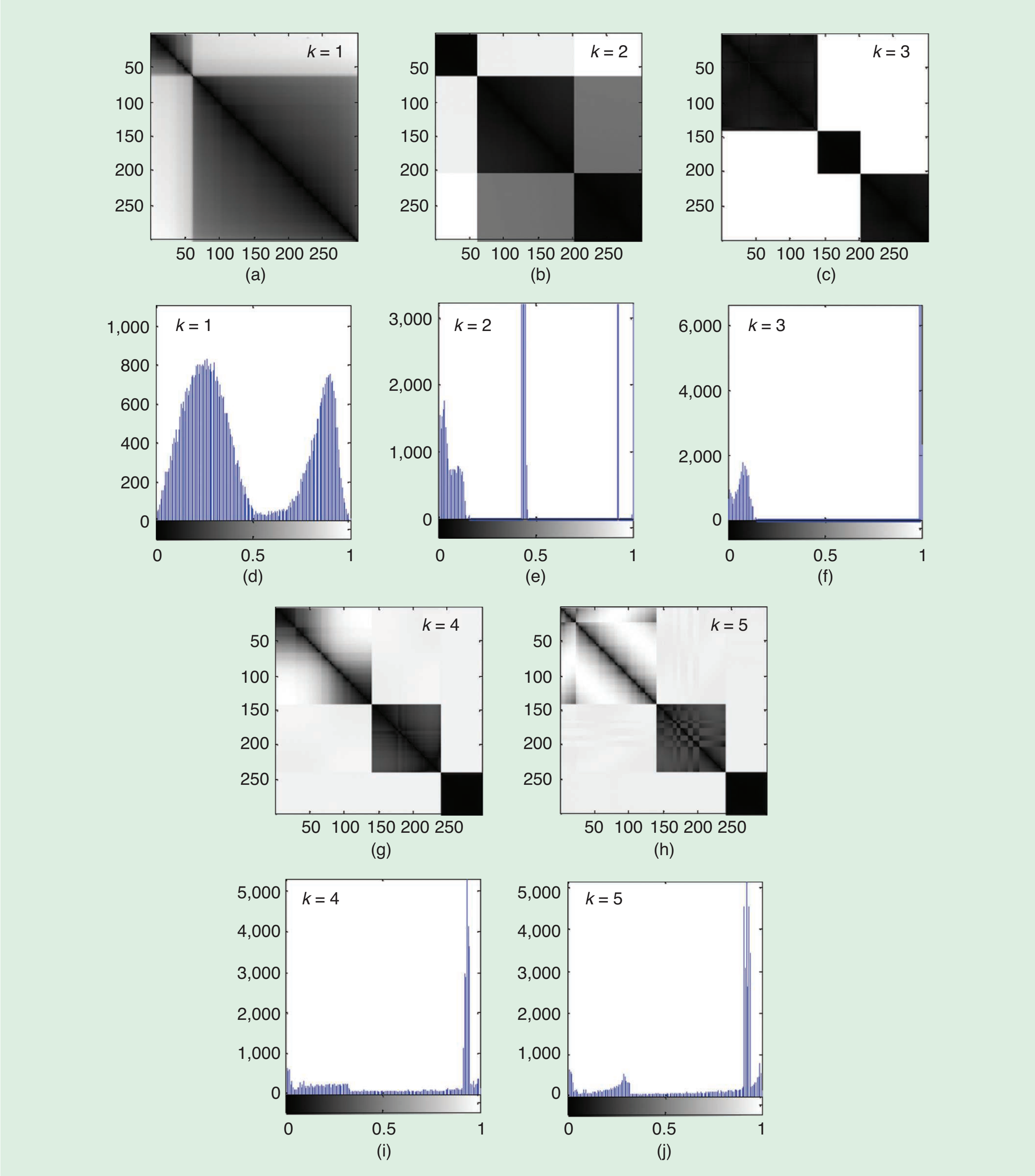


Figure 4. The SpecVAT images and their grayscale histograms for different values of k for the three-ring data set [Figure 2(b)]: (a) specVAT image ($k = 1$), (b) specVAT image ($k = 2$), (c) specVAT image ($k = 3$), (d) specVAT image histogram ($k = 1$), (e) specVAT image histogram ($k = 2$), (f) specVAT image histogram ($k = 3$), (g) specVAT image ($k = 4$), (h) specVAT image ($k = 5$), (i) specVAT image histogram ($k = 4$), and (j) specVAT image histogram ($k = 5$).

image corresponding to $k = 3$ is the clearest, with maximum contrast between the diagonal dark blocks and the nondiagonal white region. The visual suggestion is that the most likely number of clusters in the data set is $k = 3$. The pseudocode for SpecVAT is given in algorithm S3 of "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family." We will return to SpecVAT in "Applications of the VAT Family to Different Domains," where we discuss the extended SpecVAT (E-SpecVAT) algorithm [19].

Another factor that significantly deteriorates the quality of the VAT image is the presence of noise (especially inliers: bridge points between clusters). This shortcoming of VAT is inherited from the SL algorithm, which is the backbone of VAT reordering. SL has a well-known defect called the *chaining effect* [20], which happens when a few points form a bridge between two clusters, causing an SL to (mistakenly) join the two clusters into one. This chaining effect makes SL and, therefore, VAT sensitive to noise and bridge points (inliers).

Excerpted from [21], Figure 5(a) shows a data set W consisting of two well-separated clusters (X shown by green and brown points) and three inliers (the bridging points between the clusters) $\{w_1, w_2, w_3\}$ added to it (shown in black). The iVAT image of W is displayed in Figure 5(b), which weakly suggests the presence of three clusters in the data set. A solution to this problem was proposed by Kumar et al. in [21], who used three new approaches to detect and remove the inliers. The first method, distance modification with local outlier factor (LOF) (DM$^{\mathrm{LOF}}$) adjusts the distance values using LOF scores so that the influence of inliers on subsequent processing is reduced or eliminated. The other two methods are data-removal approaches based on LOF and maximin sampling anomaly scores, DR_{LOF} and DR_{MM}. These two schemes identify and remove the inliers (data cleansing) before subsequent processing begins. Figure 5(c)–(e) shows the modified iVAT images after applying these three methods; all three images now correctly indicate the presence of two clusters in the data set.

Extension of VAT to Asymmetric/Incomplete Dissimilarity Input Data

VAT and the subsequent RDI enhancement techniques discussed in the previous section require complete information about the distance matrix between the data points and require it to be symmetric for their respective algorithms to work. The symmetric requirement for the distance matrix assumes that a distance measure between two data points (say, o_i and o_j) is commutative, that is, $d(o_i, o_j) = d(o_j, o_i)$. However, for many applications, there are dissimilarity measures, which are not symmetric, such as $d(o_i, o_j) \neq d(o_j, o_i)$.

This asymmetry of D is common in social network data, where relationships are not reciprocal, and in other domains, such as bioinformatics, where the popular basic-local-alignment search tool similarity [22] between biological sequences, such as the amino acid sequences of proteins or the nucleotides of DNA or RNA sequences, is not symmetric. Sampson's monastery data [23] is an example of this type. For these data, Breiger et al. [24] give the relationship from Bonhaven to Ambrose the value 2, but the value from Ambrose to Bonhaven in the opposite direction is 1. An extension of the VAT/iVAT algorithms to asymmetric matrices [called *asymmetric iVAT* (*asiVAT*)] was proposed by Havens et al. [25]. The extension is based on replacing the asymmetric input data with its unique least-squared error approximation by a symmetric matrix. Given an asymmetric distance matrix D, the asiVAT algorithm first generates a symmetric approximation of D as $D \leftarrow (D + D^T)/2$ before applying VAT/iVAT to the (symmetric) transformed matrix.

Another challenge when determining the clustering tendency of many social network data is that, usually, the relational matrix is incomplete; that is, the dissimilarity between many pairs of objects cannot be determined, and these missing values prohibit the direct application of VAT/iVAT to such data. For example, for Sampson's monastery data, some relationships are "missing," giving rise to an incomplete input-distance matrix. According to Wasserman and Faust [26], this is the most common form of social network data.

The karate-club scenario is another popular real-life example of social network data with missing values [27]. This famous data set is an asymmetric social network that links 34 members of a university karate club. There are 156 links and 1,000 missing values. VAT/iVAT RDIs formed from incomplete data do not offer a very rich interpretation of cluster structure. To address this problem, Park et al. [28]

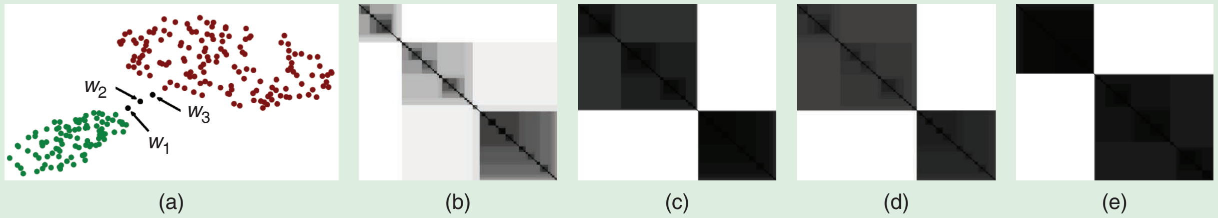


Figure 5. The application of iVAT, DM^{LOF}, DR_{LOF}, and DR_{MM} to a data set with inliers: (a) $W = X \cup \{w_1, w_2, w_3\}$, (b) the iVAT image $I(D'^*)$ of W, (c) DM^{LOF} of W, (d) DR_{LOF} of W, and (e) DR_{MM} of W. (Source: Kumar et al. [21].)

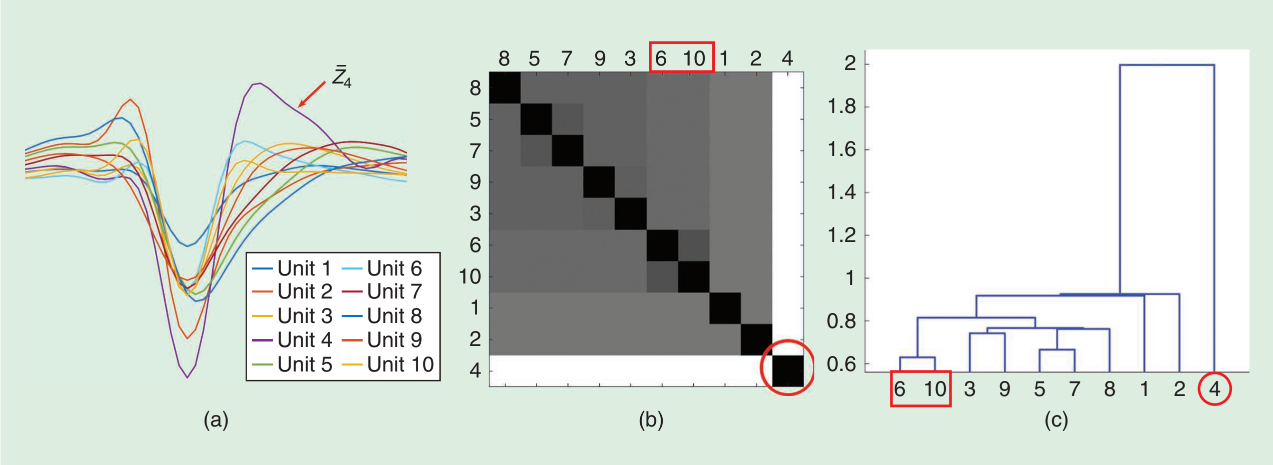


Figure 6. The iVAT as a visualization of SL clusters and dendrogram: (a) $X_{10} \sim 10$ waveforms, (b) the iVAT image of $D_E(X_{10})$, and (c) the SL dendrogram of X_{10}. (Source: Mahallati et al. [35].)

proposed that the input-distance matrix should be completed using various matrix imputation techniques, such as sampling from a given distribution, prediction from regressing over the known values (kernel regression, bootstrapped regression, and so on), and a combination of sampling and regression, before applying VAT/iVAT.

Parallelized VAT Algorithm

The VAT algorithm, although helpful in determining the cluster structure in the data visually, has an $O(n^2)$ computational complexity and becomes computationally prohibitive for even a moderately sized data set (e.g., a data set with 20,000 points). In recent years, the graphics processing unit (GPU) has emerged as an inexpensive, energy-conserving, and highly efficient single-instruction, multiple-threads parallel-computing device, and it can be found in many mainstream desktop computers and workstations.

Parveen and Sreevalsan-Nair [29] were the first to propose a parallel implementation of VAT (pVAT). The original implementation of VAT used Prim's algorithm for constructing the minimum spanning tree (MST) to find the permutation order of elements in the distance matrix. However, for the parallel implementation in [29], the authors used Boruvka's algorithm [30] to find the MST, as implemented in [31], on a GPU. The pVAT algorithm obtained the same reordered image as VAT while providing a speed of up to six orders of magnitude faster. The pseudocode for pVAT is given in algorithm S4 of "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family."

The massively parallel computing capability of CUDA-enabled GPUs was exploited by Meng and Yuan [32] to develop a GPU-accelerated VAT, which improved the computational efficiency of VAT using a parallel implementation. Along similar lines, edge-based VAT (eVAT), an edge-based algorithm that can replicate the output of iVAT [14] but is more efficient and more suitable for parallelism, was proposed by Meng and Yuan in [33]. They also proposed a parallel scheme to accelerate eVAT using the NVIDIA GPU and CUDA architecture.

Clustering Algorithms Based on VAT

The VAT algorithm and its modifications help determine whether there is a cluster structure in the data and, if so, how many clusters to look for. Based on visual information contained in the VAT RDI, any clustering algorithm can be used to find the suggested clusters (say k) in the data. However, since generating a VAT RDI requires finding the MST of the data points, a natural choice for the clustering algorithm is SL hierarchical clustering [5], which simply requires cutting the $k-1$ longest edges of the MST to generate k clusters. A paper by Havens et al. [34] explored the relationship between the VAT algorithm and SL hierarchical clustering and showed that the VAT reordering of dissimilarity data is directly related to the clusters produced by the SL hierarchical clustering.

To illustrate the relationship between iVAT and SL clustering, the example shown in Figure 6 is excerpted from [35]. Figure 6(a) is a set of 10 waveforms, X_{10}, each represented by a sample vector of $p=48$ equally spaced values. The waveform labeled x_4 is visually anomalous to the other nine graphs. Figure 6(b) is the iVAT image of X_{10} made from the 10×10 matrix of Euclidean distances between pairs of waveform vectors. The integers along the borders of the iVAT image of X_{10} show the identity of each pixel after iVAT reordering. Waveform x_4 is isolated in a single 1×1 dark block in the lower right corner of Figure 6(b). This illustrates the potential of an iVAT image to suggest anomalies in the input data.

This iVAT image also depicts the other nine waveforms as members of a second large cluster. Moreover, the image suggests a hierarchical substructure within the 9×9 block. The intensities in the highlighted (6, 10) pair suggests that these two waveforms are most closely related, followed by the (5, 7) and then (3, 9) pairs. These internal pairings are a bit hard to see in Figure 6(b), but if you look closely, they are there. The SL hierarchy is easily extracted by applying a back pass that cuts edges in the iVAT MST that reordered X_{10}. Figure 6(c) includes a dendrogram of the clusters produced by extracting the SL hierarchy of clusters this way. We see that the anomalous waveform (x_4) is reluctant to join the SL hierarchy, coming in as the last merger on the dendrogram. The evolution of clusters in Figure 6(c) is clearly visible in the iVAT image. Figure 6(b) and (c) makes the relationship between iVAT and SL quite transparent: an iVAT image can be interpreted as a visual front end to SL clustering.

Although SL is a natural clustering algorithm to use after VAT reordering because it does not require any additional computation, other choices have also been applied in the literature. Fuzzy clustering algorithms, such as fuzzy *c*-means, were used in [36]–[39] after clustering tendency was visually assessed with the VAT algorithm. Hathaway et al. [36] extend the popular kernelized clustering algorithms to relational data by proposing a kernelized form of the non-Euclidean relational fuzzy *c*-means algorithm using a VAT image as a preliminary step for clustering. Sledge et al. [37] used VAT images to determine the number of clusters in the data set before applying their reformulated fuzzy possibilistic *c*-means (PCM) algorithm, which can be applied to A-norm relational data.

Some VAT-based clustering algorithms have also been proposed in the literature. Prasad and Reddy [40] extended VAT as a complete clustering method called the *visualized clustering approach* (*VCA*) by using a Khun–Munkres function (a combinatorial optimization algorithm that solves the assignment problem in polynomial time). VCA can effectively access the number of clusters and discover the clustering results. An extension of the VCA approach, called the *context-aware graph-based VAC*, was also proposed by the same authors. This scheme computes the context-aware dissimilarity matrix (CAD) of the data set using pairwise and k-nearest neighbor hypergraphs for a set of objects. The CAD is then used as an input for VAT and the VCA clustering approach. The VAT reordering of the points in a data set was used as a preprocessing step in [41] to mitigate the ordering effects for the clustering structures formed by the fuzzy adaptive resonance theory. This approach is especially useful when performing offline incremental learning to improve clustering performance, reduce the number of categories, and decrease variability in the clustering outcome.

Application to Cluster Validity

The third and final step in the cluster-analysis task (after assessing the clustering tendency before clustering and applying the clustering algorithm to the data to generate clusters) is to verify the "correctness" of a particular set of clusters in a given data set, commonly known as the *cluster-validity problem*. Cluster validity is a widely studied problem. The vast majority of validation methods attempt to assess the quality of generated clusters by a scalar measure of partition quality. One problem inherent in this approach is that representing the correctness of particular cluster analysis by a single real number invariably loses much information.

Bezdek and Hathaway [42], [43] took the opposite approach to the cluster-validity problem and proposed a VAT-based visual display of fit approach, which they called *visual cluster validity* (*VCV*), using all of the information produced by the clustering method. Their method was inspired by the SHADE approach introduced in [44], and it applies to all prototype generator clustering methods. The VCV approach retains and organizes the information that is lost through the massive aggregation of information by scalar validity measures. VCV, although it provides good visual tools for validating clustering results for "object data," that is, data points being represented by feature vectors, cannot be applied to relational data because it needs object prototype parameters—specifically, the mean vectors—from prototype generator clustering methods, which are unavailable from relational algorithms.

To solve this problem, Ding and Harrison [45] presented a relational VCV (RVCV) method based on VCV. RVCV uses relational prototype parameters, distances, and membership values and follows the steps of VCV; however, it permits the reordering of clusters at the crucial stage (corresponding to the first stage in VCV), thus permitting generalization to relational data. RVCV presents relational cluster validity results in a natural, visual form and fills a gap in the body of visual cluster-validity theory initiated by Bezdek and Hathaway.

Gunnersen et al. [46] noticed that both VAT and VCV result in a visualization of Euclidean distance, either between the data points themselves or between the data point and the cluster prototype. They proposed a new visual cluster membership validity (VCMV) algorithm, which extends the VCV algorithm by visualizing class memberships produced by an external fuzzy clustering algorithm rather than Euclidean distance. They also combined the VCMV algorithm with self-tuning spectral clustering [47] to create SpecVCMV, which simultaneously utilizes the advantages of spectral clustering, addresses the chaining phenomenon [20] found in VAT and the underlying assumptions found in VCV to create a robust algorithm that behaves consistently across data sets and yields more useful results in complex data sets.

Huband and Bezdek proposed another VCV algorithm they called *VCV2* in [48], which compares the partitions found using any clustering algorithm with the VAT image of the unlabeled input data. The VCV2 method matches the VAT RDI image with the transformed VAT-like image of the (reordered) partition matrix generated by the clustering algorithm. The stronger the visual match, the more confident we are that the candidate partition is a useful representation of substructure in the data.

Figure 7, taken from [48], shows the VAT image $I(R^*)$ and VCV2 images $I(U^*)$ for different values of *c* for a data set consisting of five distinct Gaussian clusters. For the VCV2 images for $c=2$ and $c=4$ in Figure 7(b) and (c), the dark blocks differ visibly from those in the VAT image, so these two candidates are rejected. The VCV2 images in Figure 7(e) and (f) have roughly the same visual structure as the VAT image $I(R^*)$, the principal difference being the second block in the image for $c=6$, which is somewhat lighter in color, indicating that a very small subset of points (possibly a singleton) was chosen as the sixth cluster. Based on visual image similarity, the VCV2 algorithm clearly suggests that $c=5$ is the preferred solution.

A few papers have tried to understand the relationship between the VAT-based VCV indices described previously and the traditional numeric cluster validity indices. Havens et al. [49] addressed the relationship between the VAT algorithm and Dunn's cluster validity index [50]. Their experiments on a variety of data sets demonstrate that the effectiveness of VAT in showing cluster tendency is strongly related to Dunn's index, which provides a measure of contrast between the dark diagonal blocks in the VAT RDI and the bright background regions. Similarly, a framework that couples the possibilistic Rand index and the VAT algorithm to estimate the number of clusters and identify coincident clusters found by the PCM algorithm was discussed in [38].

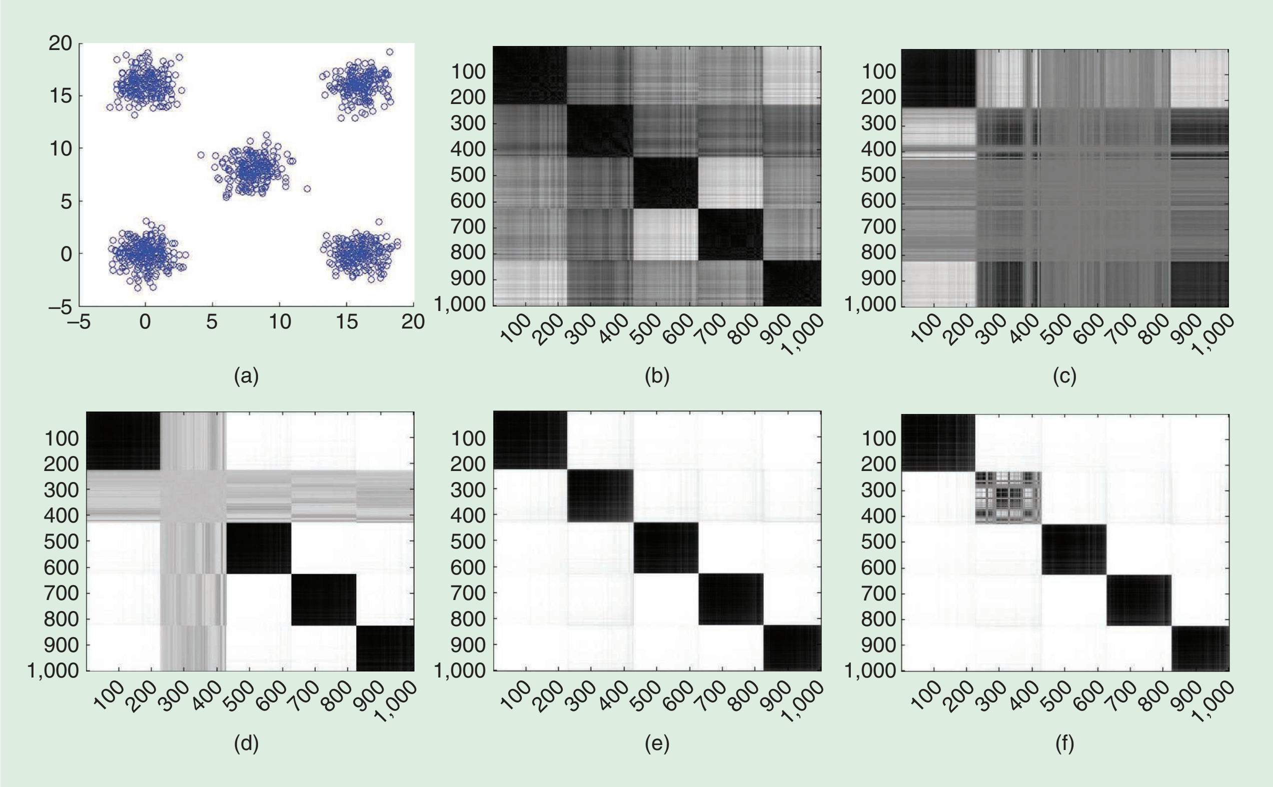
Automating VAT/iVAT for Clustering Tendency Assessment

The RDI generated by the VAT algorithm provides an excellent tool to (visually) interpret possible cluster structure in a data set. A human observer can (sometimes) count the number of dark blocks along the diagonal of a VAT image to get an estimate for k, the number of clusters for which to look. For data sets with compact, well-separated clusters, the dark blocks along the diagonal of the VAT image are clear and easily countable, but as the data become more and more mixed, the VAT image will degrade considerably. Although humans can usually deduce the suggested number of clusters from a VAT image in all but the most incorrigible data sets, different humans may see different values, especially when the clusters have significant overlap or strange geometries, both of which lead to VAT images with nondistinct diagonal block boundaries.

Due to these shortcomings, visual methods, such as VAT, have been criticized for being subjective and requiring human input, which becomes impractical and seems somewhat archaic in the current climate of automation created by advances in artificial intelligence. Several groups have tackled this problem and designed methods to automatically detect the number of clusters without requiring human input to interpret a VAT/iVAT RDI. These techniques, classified by the technique used in them to automatically determine k from the VAT RDI, are discussed next.

Automatic Assessment Based on Image Processing of the VAT RDI

Keller and Sledge [51] were the first to develop an algorithm that takes the VAT RDI as an input to determine the degree



(a) (b) (c)

(d) (e) (f)

Figure 7. The VAT $I(R^*)$ and VCV2 images $I(U^*)$ for different values of c for a five-cluster data set: (a) input data with five clusters; (b) a VAT-ordered image of $I(R^*)$; and VCV2 images of (c) $I(U^*)$ for $c=2$, (d) $I(U^*)$ for $c=4$, (e) $I(U^*)$ for $c=5$, and (f) $I(U^*)$ for $c=6$.

of clustering (not exactly the number of clusters) automatically. They generated a series of threshold plots by summing all of the black pixels of the VAT RDI for different threshold values (representing the probability of occurrence of black pixels in the VAT image). These threshold plots were then combined to generate a clustering score for the particular data set. Later works by Sledge et al. [52], [53] automatically (algorithmically) determined k, the number of clusters from the VAT image. They experimented on a variety of ineffective techniques before proposing a cluster count extraction (CCE) algorithm, which implemented frequency domain correlation and feature recognition to count the number of clusters automatically.

Inspired by the CCE technique, Wang and Leckie [54] proposed a new method called *dark block extraction* (*DBE*) for automatically estimating the number of clusters from the VAT RDI using several image- and signal-processing techniques. The main steps of the DBE algorithm are 1) dissimilarity transformation and image segmentation, 2) directional morphological filtering of the binary image, 3) distance transform and diagonal projection of the filtered image, and 4) detection of major peaks and valleys in the projection signal. The pseudocode for the DBE algorithm is given in algorithm S5 of "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family."

Figure 8 shows the output of different steps of DBE when applied to a data set consisting of five Gaussian clusters in a 2D space. The views of Figure 8 are self-descriptive; more information about them can be found in [54]. Prabhu and Duraiswamy [55] extended DBE to a new method called *enhanced DBE*, which relies on E-VAT [56], DBE, distance measures for diverse type of attributes, and basic image- and signal-processing techniques.

The iVAT algorithm proposed in [14] and [15] generates much sharper RDIs than VAT; therefore, it is easier to infer the number of clusters from the iVAT image manually or automatically. To this end, Wang et al. [14] proposed an automated VAT (aVAT) algorithm to automatically determine the number of clusters suggested by the iVAT RDI using common image-processing techniques (binarization, edge map, dissimilarity transform, and so on). A comparison of aVAT with CCE and DBE showed that aVAT was somewhat better then CCE and DBE since it explicitly showed the number of clusters, positions, and ranges of each block (or clusters) within the image itself.

Automatic Assessment Based on Spectral Graph Theory

Another category of techniques for automatically extracting the number of clusters in a data set using VAT without any human intervention are based on spectral graph theory. The SpecVAT algorithm discussed in the "Improvements in the VAT RDI for Complex Cluster Structure and Noisy Data" section belong to this category. The detailed implementation of SpecVAT, including an example showing how to obtain the best value of k without human intervention, is given in the "Improvements in the VAT RDI for Complex Cluster Structure and Noisy Data" section (see Figure 4 and the corresponding discussion).

Automatic Assessment Based on Off-Diagonal Dissimilarity

The off-diagonal values of the reordered dissimilarity matrix have been used by some researchers to automatically determine the cluster structure of a data set. Hu and Hathaway [57], [58] introduced the concept of tendency curves to identify possible diagonal blocks in the RDI by using various averages of values of the w subdiagonal band (excluding the diagonal) of the RDI, which are stored as vectors and displayed as curves. The possible cluster borders are then seen as the high–low patterns on the tendency curves, which can be caught not only by human eyes but also by the computer using a suitable threshold. An enhanced technique called *visual assessment of cluster tendency using diagonal tracing* (*VATdt*) was proposed in [59], which extensively experimented on a particular type of tendency curve called a *d-curve* and concluded that a d-curve is effective in determining the number of clusters in a data set, even in those cases where the human eyes see no structure from the visual outputs of VAT.

Automatic Assessment Based on Optimization Techniques

Some optimization-based techniques have also been proposed to automatically determine k from the VAT RDI. Havens et al. [60] defined an objective function by combining the measures of contrast and edginess of the VAT RDI to recognize the blocky structure in reordered data. The objective function is optimized when the boundaries in the VAT RDI are matched by those in an aligned partition of the objects. The authors proved that the set of aligned partitions is exponentially smaller than the set of all partitions that must be searched if clusters are sought in the raw data, thus making the optimization problem tractable. They propose an extraction of clusters from the ordered dissimilarity data (CLODD) algorithm, which uses particle-swarm optimization to find optimal clusters from the aligned partitions.

Another technique, suggested by Pakhira and Dutta [61], proposes using genetic algorithms to automatically determine the number of clusters from VAT images. Similar to Havens et al. [60], Pakhira and Dutta generated a set of aligned partitions as candidate cluster memberships and used a variable-string-length genetic algorithm, where chromosomes are real-coded with cluster centers, and Dunn's cluster validity index is used as the objective function. A related article by the same authors [62] proposed optimizing another cluster validity measure, the Pakhira, Bandyopadhyay, and Maulik (PBM) index [63], instead of Dunn's index as proposed in [61], using the same variable-string-length genetic algorithm optimization technique. A better and less computationally expensive approach to computing the PBM index value directly from the VAT image based on a robust

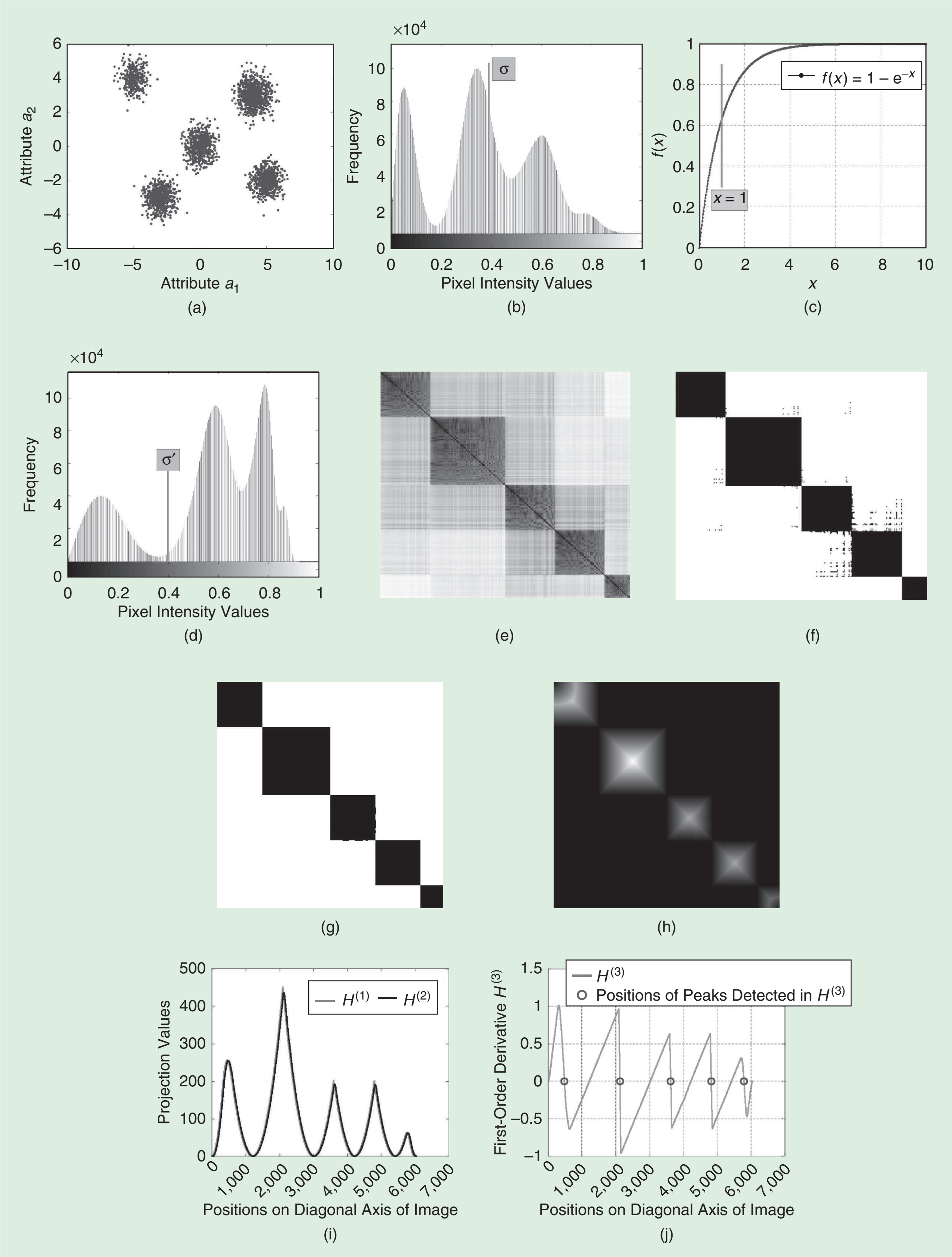


Figure 8. The application of DBE. (a) The data scatterplot. (b) A histogram of D. (c) The transformation function $d'_{ij} = 1 - e^{-d_{ij}/\sigma}$. (d) A histogram of D'. (e) A VAT image of $I(\tilde{D}')$ of D'. (f) The segmentation of $I(\tilde{D}')$. (g) A morphologically filtered image. (h) A distance-transformed image. (i) The diagonal projection signal [$H^{(1)}$] and smoothed signal [$H^{(2)}$]. (j) The first-order derivative [$H^{(3)}$].

and reliable statistically supported technique was proposed in [64], in which Pakhira and Dutta provided a quick solution for extracting k from the VAT RDI.

Extensions of VAT/iVAT in Big Data Applications

Big data is a term coined to describe the exponential growth of structured and unstructured data, which is difficult to capture, store, manage, and process with conventional data-management and -analysis techniques. With the rapid advances in information sensing, IoT devices, remote sensing, software logs, cameras, microphones, radio-frequency identification readers, wireless sensor networks (WSNs), and so on, the world's technological per capita capacity to store information has roughly doubled every 40 months since the 1980s. In 2001, Laney [65] defined the data growth challenge as having three dimensions—volume, velocity, and variety, also called the *three Vs of big data*. (Many more Vs describing various attributes of big data and the associated processing challenges have been added over the years. In 2017, there were 42 documented Vs of big data [66]).

The heterogeneity, ubiquity, and dynamic nature of the resources and devices as well as the wide variety of data make discovering, accessing, processing, integrating, and interpreting big data a challenging task [67]. Much VAT-inspired research has been performed to tackle different aspects of big data analytics to understand a variety of novel data sources and extract actionable knowledge from them. This research can be broadly classified into three categories, which cater to the high-volume, high-velocity (streaming data) and high-dimensionality aspects of big data and are discussed next. In the following sections, we use N and n to denote the sample size of "big data" and "small data," respectively.

Clustering High-Volume Data Sets

Assessing Clustering Tendency Visually

Huband et al. [68] were the first to realize the limitations of VAT to handle even moderately sized data sets (with a few tens of thousands of data points) due to its $O(n^2)$ computational complexity (n being the number of points in the data set). To address this problem, they developed an alternative way to compute the most critical information in the ordered image matrix, that is, the boundary between clusters shown by the dark blocks along the diagonal. They developed the revised VAT (reVAT) algorithm, which achieves results similar to VAT with less computation.

The reVAT algorithm builds and displays a set of profile graphs of specific rows of a pseudo-ordered dissimilarity matrix, rather than displaying an entire ordered image matrix. When presented as an ensemble, the suite of profile graphs visually suggests distinct clusters when they are present. In this case, each profile graph will have a unique peak (i.e., the peaks from one profile to the next do not overlap). However, interpreting a series of profile graphs is more difficult than the visual assessment of a VAT image, especially when the number of clusters in the data set is large—that is, reVAT does not produce a composite picture of possible substructure in the data. Also, the computational complexity of reVAT is still $O(n^2)$ due to the computation of the $n \times n$ input-distance matrix.

To solve the interpretation problem of reVAT, a new algorithm called *bigVAT* [69] was introduced, which combined the quasi-ordering technique used by reVAT with an image display of the set of profile graphs displaying the clustering tendency information with a VAT-like image. bigVAT uses random samples from reVAT-generated quasi-ordered profile graphs to create a VAT-like image to simplify the interpretation of cluster structure in big data.

Although reVAT and bigVAT address the visualization challenge of the VAT RDI for moderately sized data sets, they still suffer from the high memory requirements of storing an $n \times n$ (unordered) distance matrix. To address this challenge, Hathaway et al. [13] developed a new scalable, sample-based version of VAT called *scalable VAT* (*sVAT*) and its iVAT extension, scalable iVAT (siVAT) (algorithm S6 in "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family"), which is feasible for arbitrarily large data sets.

In a nutshell, sVAT/siVAT selects a sample of (approximately) size n from the full set of objects $O = \{o_1, o_2, \dots, o_N\}$ and performs VAT on the distance matrix of the n samples. The sample is chosen so that it (hopefully) contains a cluster structure similar to the full set. One of the most important results in [13] is a (weak) theorem that guarantees that each cluster in the big data is sampled at least once if it is compact and well separated in the sense of Dunn's index. This is done by first picking a set of k' distinguished objects using maximin sampling [70], selected to provide a representation of each of the clusters. Then, the remainder of the sample is built by choosing additional data near each of the distinguished objects. This sampling scheme is called *maximin random sampling* (*MMRS*). The VAT/iVAT algorithm is then applied to the MMRS samples. This yields a sample-based approximation to the cluster heat map of the big data set that cannot be made with VAT or iVAT.

To illustrate this, Figure 9(a) is a scatterplot of $N =$ 1,000,000 2D points drawn from a four-component Gaussian, with 250,000 points per cluster. In this case, we cannot generate a VAT image, indicated by the question mark in Figure 9(b). However, we can generate sVAT and siVAT images for this big data set by sampling $n = 500$ points (0.05% of the total data set) from O. The sVAT image [Figure 9(c)] weakly suggests four clusters, which are seen with much better visual acuity in the siVAT image [Figure 9(d)].

Wang et al. [19] proposed a random-sampling-based extension to their SpecVAT algorithm to enable VCA for large data sets. Prasad and Reddy [71] proposed a sample-based VAT (PSVAT), which uses several distinguished features (DFs) [72] to select the best random samples in a progressive sampling scheme. DFs are selected from the

$N \times N$ (unordered) dissimilarity matrix based on the criteria of selecting the most dissimilar value from the selected row of the dissimilarity matrix. Next, the VAT algorithm is applied to the samples to obtain the RDI, which helps in understanding the cluster structure of the big data. This technique becomes time and memory intensive due to the calculation of the $N \times N$ (unordered) dissimilarity matrix for very large values of N.

A different sampling-based scheme to make VAT applicable to large-volume data sets, called *out-of-core VAT* (*o-VAT*), was proposed in [73]. The preprocessing step of o-VAT compresses the data set by merging data points into small groups called buckets, each of which is then represented by the mean of its contents. The process starts with a single empty bucket and, based on the distance of the new point from existing buckets being smaller or greater than some user-defined threshold, called the *confidence radius* (χ), it is added to an existing bucket or placed in a new bucket of its own. This process leads to a set of n buckets, which represents the complete data set in compressed form. The higher the value of χ, the higher the number of buckets n, and vice versa. The VAT algorithm is then applied to the set of n buckets to infer the cluster structure of the big data.

Recently, Trang et al. [74] used a probabilistic traversing sampling (ProTraS) algorithm [75], which is based on the farthest-first traversal principle, in which a representative is selected that yields the highest probability of cost reduction. Each point of the sample points obtained by ProTraS is then replaced by the center of the set of patterns represented by the point, thereby moving the sample toward the center of the clusters. The VAT/iVAT image of the new sample is thus sharper while the main structure of clusters is maintained.

Shao et al. [76] proposed a VAT-based end-to-end clustering algorithm, called *hybrid and parameter-free clustering method*, which first samples the large data set using a fast version of MMRS, which they dubbed *MMRS**. A more general form of this sampling scheme called *nearly maximin sampling* first appeared in [77]. Subsequently, this method was used in [78], where it was called *MMRS plus* (*MMRS+*) and used as a basis for the siVAT+ algorithm. Shao et al. [76] used the same improvement to MMRS, followed by a different visual assessment image built by eVAT. The method, then, determines the number of clusters using the extraction of pixel blocks, forms different partitions (MST tree cutting), and extends the results to the rest of the data set.

Clustering Algorithms

Although the techniques described in the previous section (sVAT, PSVAT, o-VAT, and so on) help us understand the cluster structure of big data and determine the optimal value of k (the number of clusters to seek based on the visual evidence), they do not partition the data into k clusters. To address this, Havens et al. [79] extended the sVAT algorithm so that it returned SL partitions of the big data set. They named this algorithm *sVAT-SL*, which calculates an SL partition of the sVAT-sampled data and then extends this partition to the entire data set using a nearest (object) prototype rule (NPR).

They showed sVAT-SL to be a scalable instantiation of SL clustering for data sets that contain k compact-separated clusters in the sense of Dunn's index. For data sets that do not contain compact-separated clusters, sVAT-SL produces a good approximation of SL partitions. Kumar et al. [80], [81] renamed the sVAT-SL as the *clusiVAT* algorithm (algorithm S7 in "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family"), and their numerical experiments showed its superiority in terms of cluster quality and computation time over several popular big-data-clustering algorithms, such as MSTs built with the Filter–Kruskal algorithm, *k*-means, single-pass *k*-means, online *k*-means, and clustering using representatives.

It is quite fast to apply clusiVAT to a data set in which the objects are represented by their feature vectors and the Euclidean distance is used as a distance measure between objects, since it was developed with the assumption that the distance function can be computed quickly and can be performed as a batch operation; that is, the Euclidean distance of a data point from $M \gg 1$ data points can be computed as a single operation using matrix properties. This assumption, however, does not hold for many distance measures applicable to graphs and time series [83], [84]. There are many distance measures relevant to problems in different domains that are computationally expensive and can be computed in a pairwise manner

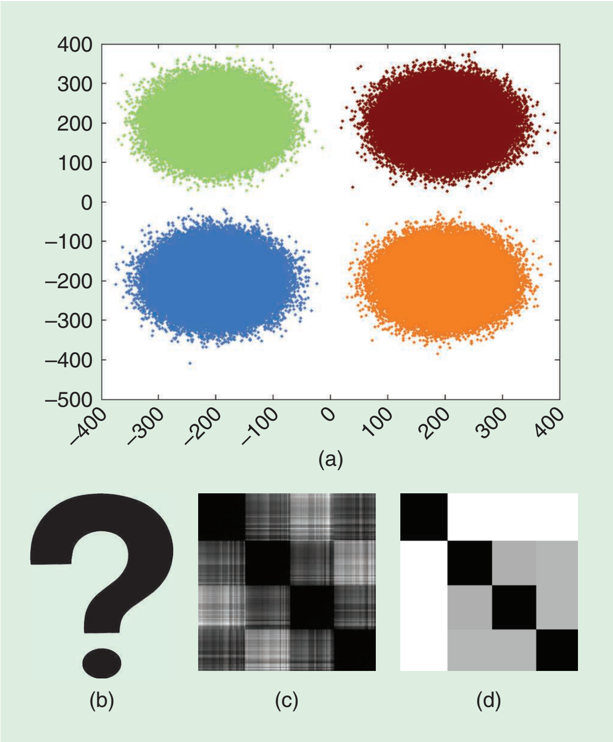


Figure 9. The data scatterplot and VAT, sVAT, and siVAT images for big data Gaussian clusters. The (a) data set of $N = 1{,}000{,}000$, (b) VAT for $N = 1{,}000{,}000$, (c) sVAT for $n = 500$, and (d) siVAT for $n = 500$.

only. To address this problem, Kumar et al. [85] proposed Fast-clusiVAT, an adaptation of clusiVAT for time-consuming distance measures. Essentially, they proposed modifications for the two most time-consuming steps of clusiVAT: MMRS sampling and the NPR extension for faster runtime without significantly compromising accuracy.

Clustering Streaming Data

The growth in network infrastructure, such as the IoT, closed-circuit television recordings, and online activities, has enabled a wide range of human activities, physical objects, and environments to be monitored in fine spatial and temporal detail. Automatic interpretation of these evolving data streams is required for the timely detection of interesting events. Conventional clustering techniques, which provide a static snapshot of each data point's commitment to every group, may not be sufficient for adapting to the presence of new clusters or even the merging of existing data-dense regions for streaming data sets.

To overcome this deficit, Sledge and Keller [82] explored the use of growing neural gas (GNG) for temporal clustering and developed a novel clustering scheme called *GNG clustering* (GNGC) for streaming data sets, which helps increase the stability of previously learned clusters but also promotes plasticity for exploring forming structures. Although GNGC is helpful in online clustering, the visualization of clustering results is equally important. When working with low-dimensional data, say, $X \subset \mathbf{R}^3$, it is easy to display intermittent GNGC results. If the dimensionality of the data set is greater than 3, however, capturing the same spatial information and visualizing the learned distributions is problematic.

The VAT algorithm was modified in [82] to instead use the information obtained from GNGC to create VAT-like images called *neuronal dissimilarity images* (NerDI) by applying the VAT algorithm to the dissimilarity matrix of the neurons in each isolated graph to better understand the evolving cluster structure of the streaming data. Figure 10 illustrates a few snapshots of a temporal data set (top row) and corresponding NerDI images of the GNGC clustering results at those timestamps. As more data are added with time, the corresponding NerDI plots highlight the changing cluster structure of the data set by the changing structure of the dark blocks along the diagonal.

Although the approach presented by Sledge and Keller [82] is useful, it requires the computation of a VAT image every time a new data point arrives, which, owing to the $O(n^2)$ time complexity of VAT, can become computationally prohibitive. To solve this problem, Kumar et al. [86] proposed a novel MST-based incremental update mechanism to achieve computationally efficient visual assessment. The new algorithms, incremental VAT (inc-VAT), incremental iVAT (inc-iVAT), decremental VAT (dec-VAT), and decremental iVAT (dec-iVAT), provide efficient mechanisms to update the VAT or iVAT RDIs if a new point is added to or an existing point is removed from the current data set. The pseudocodes for these algorithms are given in algorithms S8–S20 in "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family." The time complexities of inc-VAT/dec-VAT and inc-iVAT/dec-iVAT compares favorably to those of VAT and iVAT and are useful for applications of anomaly detection and sliding-window-based online visual assessment of evolving cluster structure in streaming data.

To illustrate the effectiveness of capturing the evolving cluster structure and the computational efficiency of inc-VAT/inc-iVAT/dec-VAT/dec-iVAT compared with VAT/iVAT, an experiment performed on a 2D Gaussian mixture of five clusters, called X, that has a total of 5,000 data points is shown in Figures 11 and 12. The numbers of data points in each of the five clusters are 1,300; 1,000; 400; 1,600; and 700, respectively. The data points in X are arranged according to the cluster they belong to. Hence, the first 1,300 rows of X are x and y coordinates of the data points belonging to the first cluster, the next 1,000 rows represent the data points belonging to the second cluster, and so on. We start with the first two data points and add one data point at a time.

The first column in all of the rows of Figure 11 includes subsets of X consisting of the points belonging to the first cluster, first two clusters, and so on. Their respective inc-VAT and inc-iVAT images in the second and third columns of Figure 11 show that the incrementally built iVAT images correctly track the changing cluster structure of the points in the data set. Although the inc-VAT and inc-iVAT images in Figure 11(k) and (l) seem to show three dark blocks, a closer look at the images shows two subblocks within the top-left dark block along the diagonal, owing to the proximity of the two clusters, $X_{1-1,300}$ and $X_{2,301-2,700}$, as compared to others. To illustrate the difference in time complexities of VAT/iVAT and inc-VAT/inc-iVAT, we perform an experiment on the same 2D Gaussian mixture X, but we randomize the rows of X so that the data points belonging to the same cluster are not indexed sequentially.

We start with the first two data points and add one data point at a time. At each step, the VAT, iVAT, inc-VAT, and inc-iVAT algorithms are executed, and the time taken to compute the respective reordered dissimilarity matrices is recorded. Figure 12(a) shows that the time taken to update the inc-iVAT image is much less than that needed to generate a new iVAT image as n increases. Similarly, to illustrate the time complexity difference between dec-VAT/dec-iVAT and VAT/iVAT, we perform an experiment on the same 2D Gaussian mixture X, but this time, we start with $n = 5{,}000$ data points and remove one randomly chosen data point at a time. Figure 12(b) reveals that the time taken to generate the iVAT image from the distance matrix (black curve showing VAT + iVAT) is of the order of $o(n^2)$ and is much higher than the time taken to update the dec-iVAT image (green curve showing dec-VAT + dec-iVAT) for large values of n.

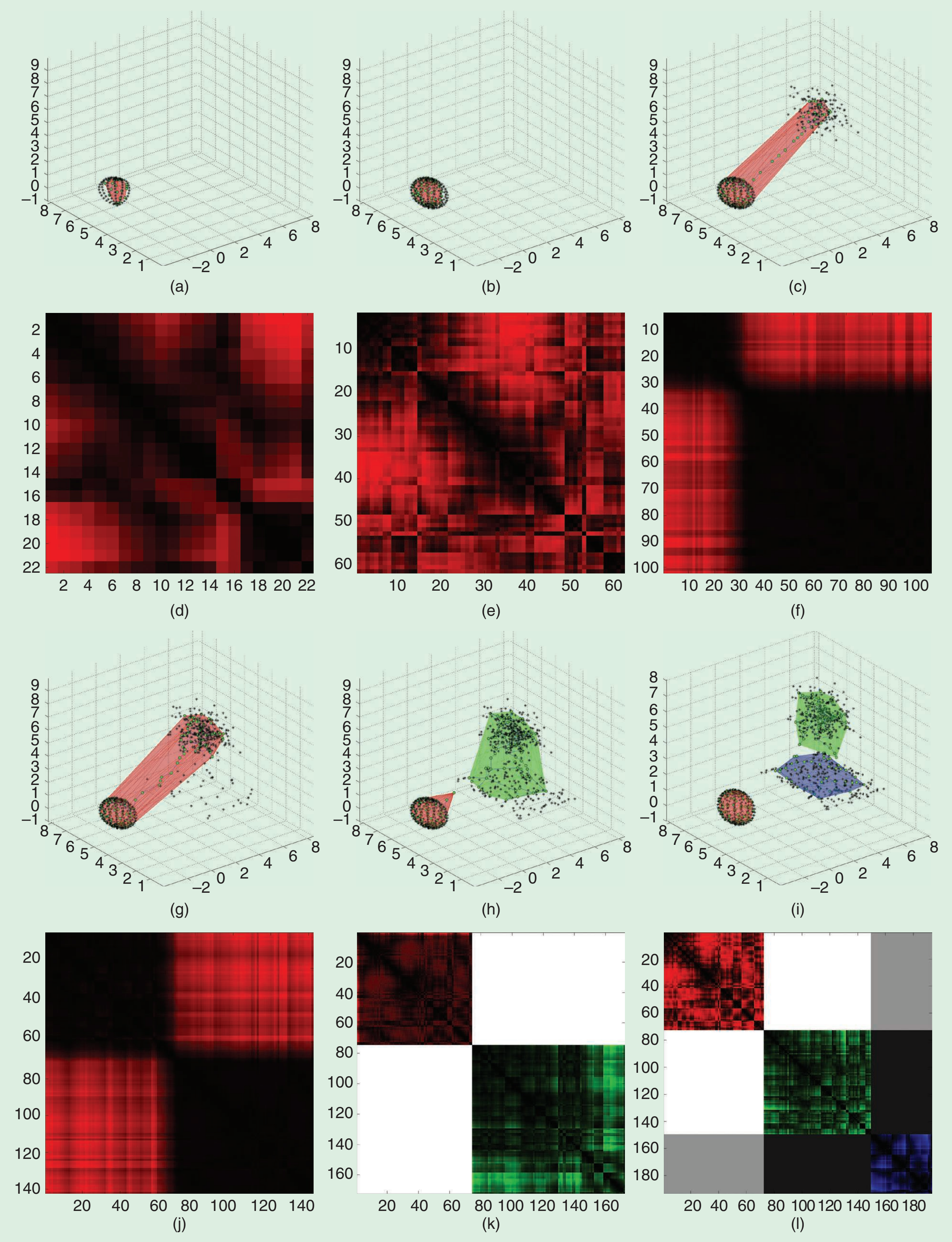


Figure 10. The temporal plots and the corresponding NerDIs of the GNGC clustering results [82]. (a) A single cluster is shown in red, and its convex hull is illustrated by dots. (b) More data are added to the red cluster, but the convex hull shows little change. (c) New data are far from the red cluster causing significant change in the convex hull. (d) The NerDI image displays a single cluster. (e) The NerDI image still shows the presence of a single cluster. (f) The NerDI plots highlight a high dissimilarity between the points in the red convex hull by two dark blocks along the diagonal. (g) More newly arriving data are far away from the original cluster shown in (a). (h) The data now have two clusters (red and green), each having a separate convex hull. (i) With the arrival of new points, a new blue cluster is carved out of the green cluster. (j) The two dark blocks along the diagonal have comparable sizes. (k) The NerDI image now has two blocks (red and green) for two separate clusters. (l) The NerDI image reveals three clusters by different colored (red, green, and blue) blocks.

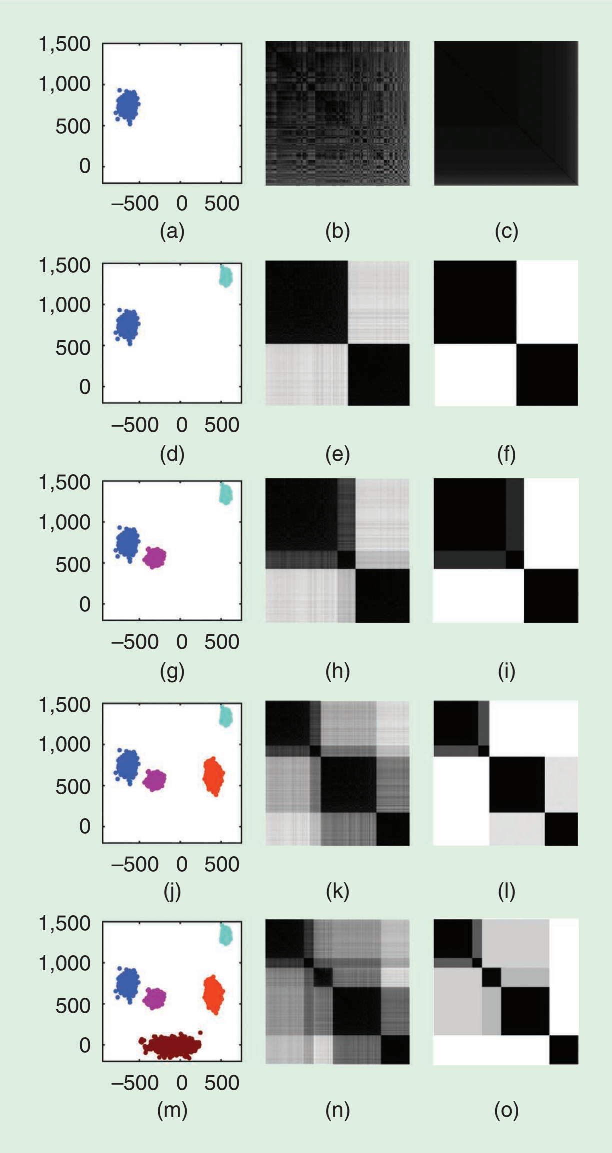


Figure 11. The 2D data scatterplots and (incrementally built) inc-VAT and inc-iVAT images of X at n = 1,300; 2,300; 2,700; 4,300; and 5,000: (a) $X_{1,300}$, (b) inc-VAT ($X_{1,300}$), (c) inc-iVAT ($X_{1,300}$), (d) $X_{2,300}$, (e) inc-VAT ($X_{2,300}$), (f) inc-iVAT ($X_{2,300}$), (g) $X_{2,700}$, (h) inc-VAT ($X_{2,700}$), (i) inc-iVAT ($X_{2,700}$), (j) $X_{4,300}$, (k) inc-VAT ($X_{4,300}$), (l) inc-iVAT ($X_{4,300}$), (m) $X_{5,000}$, (n) inc-VAT ($X_{5,000}$), and (o) inc-iVAT ($X_{5,000}$).

Clustering High-Dimensional Data

The clusiVAT algorithm was proven to be useful in determining the cluster structure of big data with a large number of data points (high volume). However, these approaches are time consuming when the data are large in both the number of samples (N) and the number of dimensions (p). To tackle this issue, Rathore et al. [78] introduced a fast, approximate, scalable iVAT algorithm called *siVAT+*, which combines random sampling with the original sampling scheme (MMRS) of siVAT to reduce the computational cost of siVAT for large volumes of high-dimensional data.

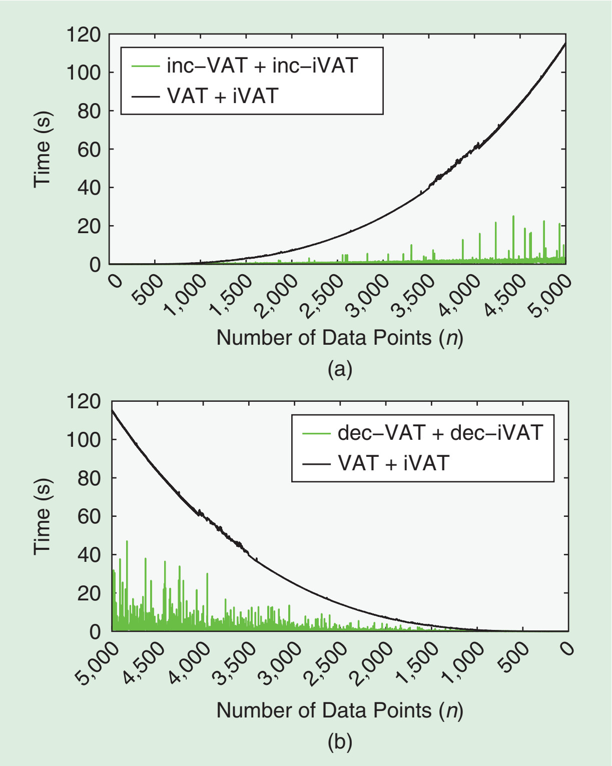


Figure 12. Time comparisons of the VAT, iVAT, inc-VAT, inc-iVAT, dec-VAT, and dec-iVAT algorithms for the 5,000-point 2D data set: (a) VAT + iVAT versus inc-VAT + inc-iVAT and (b) VAT + iVAT versus dec-VAT + dec-iVAT.

The modification of the original sampling scheme is called *MMRS+*, and it randomly selects an object from the entire data set and designates it as the first distinguished object. A second distinguished object is a maximin object from a randomly generated sample of the big data set. A new random subset of the data set is chosen, and a maximin sample is chosen from it. Sampling this way continues until the desired number of distinguished objects are obtained. The MMRS+ samples are then used to build an approximate siVAT image for the very large, high-dimensional data, which provides visual evidence about the potential number of clusters to seek in the big input data set.

To demonstrate siVAT+, Figure 13 (see Figure 2 in [78]) shows the comparison of RDI and computation times between siVAT and siVAT+ on two synthetic data sets, Gaussian mixture (GM) 1 and GM2, each with $N = 1{,}000{,}000$ data points in $p = 1{,}000$ dimensions, constructed by drawing labeled samples from a mixture of $k = 3$ Gaussian distributions. Data set GM1 is a well-separated Gaussian mixture, whereas GM2 has overlapping Gaussian clusters. Experiments were also performed on four publicly available real, high-dimensional (large-volume) data sets: knowledge discovery from data (KDD)-99 cup data [87], forest-cover type data [88], U.S. Census 1990 data [89], and BigCross data [90]. The conclusions that can be made from Figure 13 are as follows.

- siVAT and siVAT+ contain essentially the same visual information about cluster structure in the samples

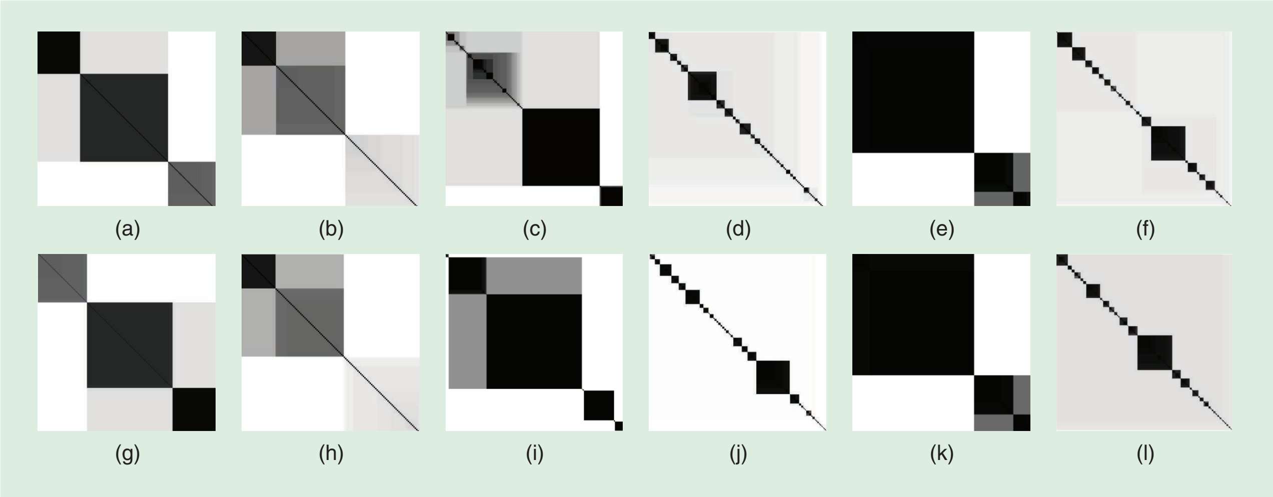


Figure 13. The siVAT and siVAT+ RDIs (D'^{*}_{n}) and runtimes for each of the data sets (parameter values: $k' = 50$ and $n = 500$): (a) GM1: siVAT, 725 s; (b) GM2: siVAT, 738 s; (c) KDD: siVAT, 76.5 s; (d) forest: siVAT, 10.8 s; (e) census: siVAT, 65.5 s; (f) BigCross: siVAT, 2,504 s; (g) GM1: siVAT+, 49 s; (h) GM2: siVAT+, 48 s; (i) KDD: siVAT+, 6 s; (j) forest: siVAT+, 0.8 s; (k) census: siVAT+, 8.1 s; and (l) BigCross: siVAT+, 44.7 s.

processed. The order of the dark blocks along the diagonal may be different, but this is unimportant; their number and size are almost the same.

- siVAT+ produces this information 10–50 times faster than siVAT.

The pseudocode for siVAT+ is given in algorithm S21 in "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family."

However, like sVAT, siVAT+ only suggests the number of clusters in the data set; it cannot find the actual partition. To find clusters in large volumes of high-dimensional data while simultaneously overcoming both the "curse of dimensionality" problem due to high dimensions and scalability problems due to large sample size, Rathore et al. [91] proposed a new fast clustering algorithm called *fast ensemble siVAT* (FensiVAT), which is a hybrid, ensemble-based clustering algorithm that uses fast data-space reduction and an intelligent sampling strategy.

FensiVAT aggregates multiple distance matrices, computed in a lower-dimensional space, to obtain an approximate iVAT image in a fast and efficient manner, which provides visual evidence about the number of clusters to seek in the original data set. MMRS sampling picks distinguished objects from the data set; therefore, it requires relatively very few samples, compared with random sampling, to yield a diverse subset of the big data that represents the cluster structure in the original (big) data set. To be computationally efficient, FensiVAT performs a near-MMRS sampling, which is done in a randomly projected down space. The samples are then lifted by transferring the sample indices to the (input) up space. An ensemble of Q $n \times n$ distance matrices computed from Q sets of near-MMRS samples in the up space is then used to obtain a reliable output iVAT image, which visually suggests the number of clusters, k, in the data set.

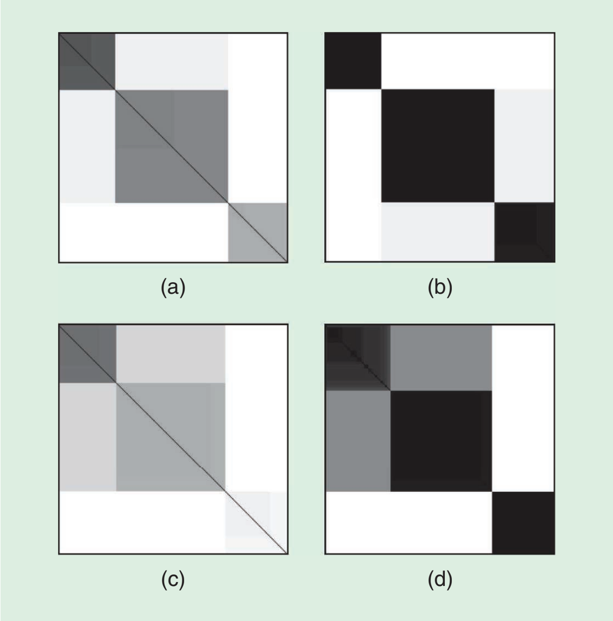


Figure 14. The ClusiVAT and FensiVAT images and runtimes for GM1 and GM2 data sets (parameter values: $k' = 9$, $n = 205$ for GM1 and $k' = 12$, $n = 206$ for GM2; down-space dimensions for FensiVAT: $q = 20$ for GM1 and $q = 50$ for GM2): (a) GM1: clusiVAT, 20.1 s; (b) GM1: FensiVAT, 0.35 s; (c) GM2: clusiVAT, 21.3 s; and (d) GM2: FensiVAT, 0.84 s.

The iVAT images obtained using clusiVAT and FensiVAT for the GM1 and GM2 data sets described previously are included in Figure 14 (see Figure 4 in [91]). Figure 14(a) and (b) shows that clusiVAT and FensiVAT exhibit three (well-separated) dark blocks along the diagonal, suggesting $k = 3$ for GM1. The ground truth partition for GM2 is not compact and well separated because of overlapping clusters; for this data set, FensiVAT produces three dark blocks along the diagonal for GM2, whereas clusiVAT shows three light blocks, including many tiny blocks (data points) along the diagonal.

Although clusiVAT and FensiVAT both show three blocks for GM1 and GM2, FensiVAT provides a more convincing assessment because of the sharper contrast between diagonal blocks and the background. Moreover, FensiVAT takes only a small fraction of the time needed by clusiVAT for both data sets. The sizes of the diagonal blocks in all four images show the relative size of each cluster accurately, which supports the claim that near-MMRS sampling replicates (approximately) the same cluster distribution in the sample as the MMRS sampling used by clusiVAT. Finally, SL clustering of the samples and NPR extension to the rest of the data set is performed in the down space for different random projections, and majority-voting-based schemes are used to assign the final cluster labels. The pseudocode for FensiVAT is given in algorithm S22 in "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family."

Coclustering

The VAT family of algorithms tackles the problem of clustering tendency assessment and subsequent clustering for an $n \times n$ (square) dissimilarity (or, in more general terms, relational) matrix. An even more general form of relational data is rectangular. These data are represented by an $m \times n$ dissimilarity matrix D, where the entries are the pairwise dissimilarity values between m-row objects O_r and n-column objects O_c. An example comes from Web-document analysis, where the row objects are m webpages, the columns are n words, and the (dis) similarity entries are occurrence measures of words in the webpages [92].

Another important problem involving rectangular relational data is the analysis of gene-expression data, where the m rows correspond to genes and the n columns correspond to tissue samples or conditions [93]. In each case, the row and column objects may be nonintersecting sets, so structural relations between the row (or column) objects are unknown. Conventional relational clustering algorithms are ill equipped to deal with rectangular data. Additionally, the definition of a cluster as a group of similar objects takes on a new meaning. There can be groups of similar objects that are composed of only row objects, only column objects, only mixed objects (often called *coclusters*), and, finally, clusters in the union of all of the row and column objects. In other words, a rectangular dissimilarity matrix comprises four different clustering problems.

Bezdek et al. [94] developed an approach for visually assessing cluster tendency for the objects represented by a rectangular relational data matrix D by assuming that D is an $m \times n$ (sub)matrix containing only $m \times n$ of the $(m+n) \times (m+n)$ possible pairwise dissimilarities between objects in $O = O_r \cup O_c$. The full distance matrix $D_{r \cup c}$ of O was assumed to be of the form

$$D_{r \cup c} = \begin{bmatrix} D_r & D \\ D^T & D_c \end{bmatrix},$$

where D^T is the transpose of D.

The coVAT coclustering VAT (coVAT) algorithm proposed in [94] first generates an estimate of D_r and D_c by interpreting the m rows and n columns of D as feature vectors representing m row objects and n column objects, respectively, and imputing the missing values using the (Euclidean) distance between them. The VAT algorithm is then applied to $D_{r \cup c}$ to generate the reordering indices of the objects in $O = O_r \cup O_c$. The coVAT algorithm then unshuffles the row objects from the column objects based on their indices to generate individual row and column reordering arrays: *RP* (for row permutation) and *CP* (for column permutation).

The coVAT image is then produced by displaying a (scaled) version of the rectangular matrix $\tilde{D} = [\tilde{d}_{ij}] = [d_{RP(i),CP(j)}]$, for $1 \le i \le m$ and $1 \le j \le n$, obtained by reordering the rows and columns of the original matrix D using the indices stored in *RP* and *CP*, respectively. Just as with VAT, dark blocks in $I(\tilde{D})$ (not along any diagonal, and not necessarily square) suggest the existence of coclusters. The pseudocode for coVAT is given in algorithm S23 in "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family."

The basic idea of coVAT is embodied in Figure 15, which is excerpted from [94] (Figures 1 and 3). Figure 15(a) depicts a set of $n = 20$ points labeled as row objects (the circles) and $m = 40$ points labeled as column objects (the squares). It may be helpful to imagine the circles as women (○) and the squares as men (□) who have congregated at the locations shown in five small groups. Three of the groups are "pure," or unmixed: the two sets of squares at the top and the centrally located set of circles at the bottom. The lower left and lower right clusters are mixed groups of circles and squares, that is, coclusters. The spatial coordinates of these points are used only to compute Euclidean distances between the circles and squares, yielding a rectangular dissimilarity matrix for input to coVAT. Evidently, there are three clusters (O_r) in the row objects (ignoring the column objects), four clusters (O_c) in the column objects (ignoring the row objects), five clusters $(O_{r \cup c})$ in the union of the row and column objects, and two mixed coclusters in $O_{r \cup c}$.

Figure 15(b)–(e) shows the coVAT images built from the rectangular input data for each of these four cases. The numbers of clusters for each of the four clustering problems are seen in the images as dark blocks: diagonal for the three square subproblems and nondiagonal for the coclustering problem. In this simple example, coVAT images provide a good visual estimate for possible cluster structure in all four problems. The coVAT algorithm was extended to the coiVAT algorithm in [95] by applying a path-based distance transform used by the iVAT algorithm.

Havens et al. [95], [96] proposed an alternate coVAT reordering scheme called *coVAT2*, which does not run VAT on the $D_{r\cup c}$ matrix as is done by coVAT. Instead, it generates a reordering of the row and column indices of D by applying VAT to D_r and D_c, respectively. Based on this row and column reordering, the row and columns of D are reordered to showcase possible coclusters in D. The pseudocode for coVAT2 is given in algorithm S24 in "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family." The coVAT2 algorithm is not limited to just relational data, and it can also be applied to feature data, such as gene microarray data. Another important advantage of coVAT2 is that it does not use VAT to reorder $D_{r\cup c}$, so, unlike coVAT, it can be applied to dissimilarity data that have negative values as well.

Honda et al. [97] proposed using a spectral-ordering-based approach on the full distance matrix $D_{r\cup c}$ for visual cocluster structure assessment. They observed that, by considering the sparse nature of the enlarged matrix $D_{r\cup c}$, the eigenproblem of the full square relational matrix is reduced to a smaller problem with less computational cost. This approach is different from the heuristic approach used by the coVAT algorithm [94] and provides an analytical solution through the minimization of an objective function.

Coclustering Big Data

Similar to VAT, the coVAT and coVAT2 algorithms suffer from high computational complexity and memory requirements as the data size increases. To address this issue, Park et al. [98] pursued the use of the sVAT sampling scheme to extend coVAT to very large data and named the new algorithm *scalable coVAT* (*scoVAT*). The key step in scoVAT is to work out a method for sampling the big $D_{M\times N}$ distance matrix when it exceeds the capacity limits of coVAT. To do so, Park et al. [98] used sVAT on the m row objects and n column objects of $D_{M\times N}$ to generate a representative sample $D_{m\times n}$, where $m \ll M$ and $n \ll N$.

The coVAT algorithm is then applied to the small $D_{m\times n}$ distance matrix to infer coclusters in the big data. The pseudocode for scoVAT is given in algorithm S25 in "Pseudocode for Various Algorithms Belonging to the Visual Assessment of Tendency Family." This procedure is easily extended to scoiVAT by replacing VAT with iVAT in algorithm S25.

Applications of the VAT Family to Different Domains

Due to its applicability to EDA, parameter-light nature, and visual output, the VAT family of algorithms has been extensively used in a variety of applications to understand newly collected data sets; based on this initial analysis, different future tasks are designed for more advanced data analysis. The application areas for VAT are diverse and cover a range of topics, such as audiovisual data processing, biomedical applications, smart city, social media data analysis, WSNs, and so on. Next, we describe some of the works that have utilized the VAT family of algorithms in various application domains.

Application to Multimedia Data

Speech Data Processing

A series of papers by Prasad, Nennuri, Reddy, and Basha [99]–[101] used VAT, iVAT, and SpecVAT with the GMM and cosine distance matrix for the application of speaker classification. They developed new techniques, GMMVAT and cosine-based VAT (cVAT), for clustering speech utterances by the same speaker. In GMMVAT, the GMM mean vectors are derived for a set of speech segments (or utterances), whereas in cVAT, the cosine distance function is used to calculate the distance matrix before applying VAT to it. The experimental evaluations performed on a variety of data sets show that GMMVAT/cVAT gives a better assessment of cluster tendency for speech data. Another VAT-based technique for speaker clustering was proposed in [102], which derives the explicit speaker clustering results directly from VAT instead of using either k-means or MST-based clustering.

Image Processing

Chen [103] proposed a new nonparametric mechanism based on unsupervised learning (VAT) for feature assessment and

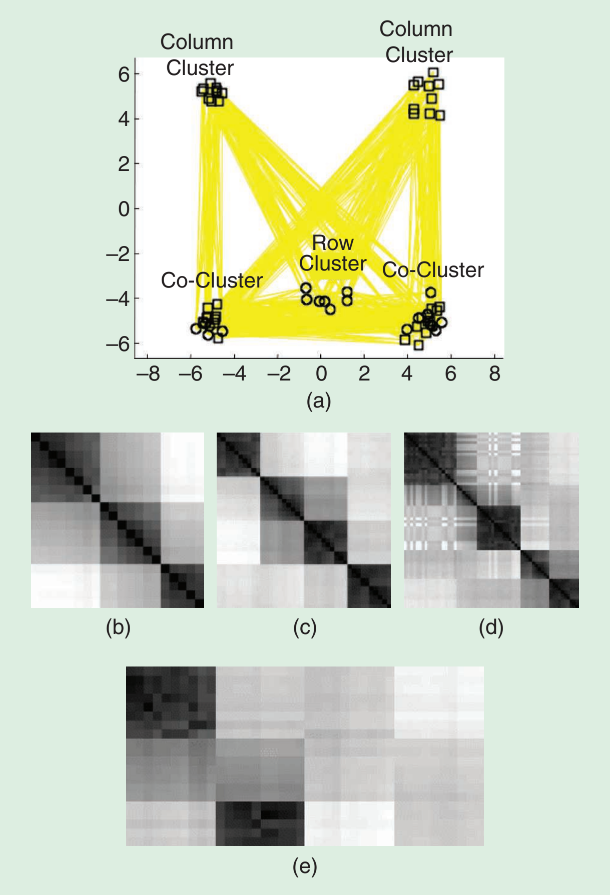


Figure 15. An example illustrating coVAT with (a) 20 row (○) and 40 column (□) objects as points in the plane: (b) $k = 3$ in O_r, (c) $k = 4$ in O_c, (d) $k = 5$ in $O_{r\cup c}$, and (e) $k = 2$ (mixed) in $O_{r\cup c}$. (Source: Bezdek et al. [94].)

selection in image clustering. The proposed approach adaptively uses VAT with consideration of the built graph (VAT-G) to draw an image G composed of vertices V and edges E. The VAT-G model was used to choose the nearest and farthest neighbors for performing clustering to satisfy both within-cluster and between-cluster scatter criteria.

Synthetic aperture radar (SAR) provides a day or night, all-weather means of remote sensing, which provides useful information about Earth. Spaceborne platforms continuously deliver enormous amounts of SAR data as highly complicated images, which are almost impossible to interpret manually. For automatic interpretation of SAR images, Liu et al. [104] proposed an unsupervised classification framework that estimates the number of classes (clusters) in an image using the VAT algorithm and the DBE method [54].

An extension of this article by the same authors for a special type of SAR image called *polarimetric SAR* (*PolSAR*) was presented in [105]. Their approach first partitioned the PolSAR image into superpixels, which are local, coherent regions that preserve most of the characteristics necessary for image information extraction, before using the VAT and DBE algorithm for clustering and classification operations. Zou et al. [106] proposed an alternate approach to cluster superpixels based on mean Freeman decomposition and hyperspectral-image color feature vectors using the VATdt approach [57], [58] to adaptively estimate the number of terrain classes and automatically capture the cluster structure.

Wang et al. [19] present some interesting results related to the segmentation of digital images using their method of E-SpecVAT. Figure 16 is part of a set of outputs made by the E-SpecVAT algorithm (see Figure 7 in [19]). The input data set is a digital image from the Berkeley image segmentation database, shown in Figure 16(a). The size of the image is $431 \times 321 = 154{,}301$ pixels, so the dissimilarity matrix on pairs of pixels has a bit more than $N = 23 \times 10^9$ elements. There are $k = 3$ clearly visible clusters in the input image corresponding to the steps and entrance (darkest); the church (medium dark), and the sky (lightest).

Figure 16(b) presents an E-SpecVAT view of the dissimilarity matrix made from a tiny sample of 300 pixels—that is, approximately 0.2% of the input data—and it clearly shows $k = 3$ clusters, with block pixel sizes roughly proportional to the sizes of the three areas in the input image. Figure 16(c) is the E-SpecVAT segmentation of the image using intensity as the input feature (i.e., $d_{j,k} = |I_j - I_k|$). This input image is simple, so its segmentation was particularly challenging, but this example shows the power of visualization with a VAT model in the area of image processing.

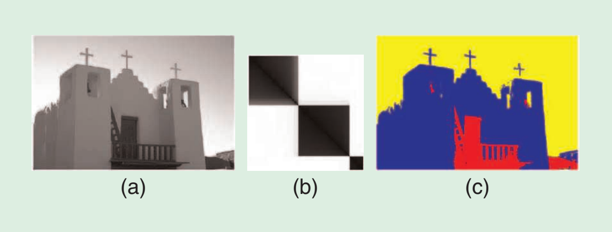


Figure 16. An E-SpecVAT image analysis: (a) the input image, (b) a 300×300 E-SpecVAT image, and (c) the image segmentation.

Video Data Analysis

Discovering actionable knowledge from large volumes of video data are of increasing interest to researchers. Examples of patterns that can be discovered from raw video sequences include determining typical and anomalous patterns of activity, classifying activities into known categories (e.g., walking or riding), and discovering unknown action patterns by clustering. Wang et al. [107] described a tensor space representation for analyzing human activity patterns in monocular videos and used the VAT algorithm for the task of activity discovery from video data.

Application to Organizational Data

Enterprise software systems are large and complex. Security in such systems is of extreme importance for organizations. Role-based access control (RBAC) is an efficient and flexible model for controlling computer resource access and enforcing organizational policies. Deployment and maintenance of RBAC requires role engineering, which defines the set of roles that accurately reflects the needs of the enterprise. Zhang et al. [108] proposed a VAT-based role-engineering tool for the visual assessment of user and permission tendencies (RoleVat) that produces natural groupings of users and permissions and helps determine the role permissions for different individuals within the organization.

Another important challenge for deploying a software system at a large enterprise is ensuring its smooth integration with many other interconnected systems, such as mainframes, directory servers, databases, and other types of software services. To do so, it is necessary to test it in as realistic an environment as possible, before actual deployment. However, getting access to the actual production environment for testing is usually impossible due to the risk of disruption.

To address this problem, Du et al. [109] and Versteeg et al. [110] developed a VAT-based practical, scalable, and fully automated approach to service emulation that uses no explicit knowledge of the services, their message protocols, or structures and yet can simulate—to a high degree of accuracy and scale—a realistic enterprise system deployment environment. In both of these papers, VAT was used to group the transactions by operation type, without assuming any knowledge of the message format, which was later used to generate the simulator output.

Application to Biomedical Engineering

The VAT family of algorithms has been extensively used for understanding the data generated from biomedical applications, such as electrocardiography (ECG), the effects of drugs, amino acids, and so on. Lourenco and Fred [111] applied VAT-based clustering to analyze ECG recordings performed during the execution of a cognitive task using the computer, such as a concentration task where two grids with 800 digits were presented, and participants were given the goal of identifying every pair of digits that added 10; the activity was designed for an average completion time of 10 min. This task was meant to induce stress in participants, and clustering of the ECG signals helped researchers understand the typical patterns of the temporal evolution of the ECG-extracted features.

In the field of pharmacology, Stallaert et al. [112] studied the drugs targeting a single G-protein-coupled receptor, which is involved in many diseases and is also the target of approximately 34% of all modern medicinal drugs. These drugs can differentially modulate distinct subsets of the receptor signaling repertoire, but they create a challenge for drug discovery at these important therapeutic targets.

Recognizing that impedance responses provide an integrative assessment of ligand activity, Stallaert et al. [112] screened a collection of β_2- adrenergic ligands to determine if differences in the signaling repertoire engaged by compounds would lead to distinct impedance signatures. To this end, they visualized the pairwise differences among ligand signatures using the VAT algorithm; this suggested that the ligands fall into five distinct clusters, which were later confirmed by hierarchical clustering. To help pharmaceutical companies analyze their ever-increasing corporate database of compounds for internal diversity or the diversity that they add to the current compounds, Rivera-Borroto et al. [113] used VAT and Dunn's index as a measure of cluster separability to assess the classification accuracy of various clustering algorithms tested on eight pharmacological data sets.

Amino acids are the basic building blocks of proteins, which are critical to life, and they have many important functions in living cells. The AAindex is a database of numerical indices representing various physicochemical and biochemical properties of amino acids and pairs of amino acids. Saha et al. [114] presented a novel method of partitioning the bioinformatics data using consensus fuzzy clustering and visualized the clustering solution using VAT reordering. The discovered clusters describe some of the properties of amino acids, such as the isoelectric point, polarity, molecular weight, average accessible surface area, mutability, hydration potential, refractivity, optical activity, and flexibility. These cluster structures were then used to resolve the problem of unknown amino acid indices by assigning them to clusters that have defined biological meaning.

Recent advances in high-throughput lipid profiling by liquid chromatography/electrospray ionization/tandem mass spectrometry have made it possible to quantify hundreds of individual molecular lipid species (e.g., fatty acyls, glycerolipids, glycerophospholipids, sphingolipids) in a single experimental run for hundreds of samples. This can help identify lipid biomarkers significantly associated with disease risk, progression, and treatment response. Clinically, these lipid biomarkers can be used to construct classification models for disease screening or diagnosis.

However, the inclusion of a large number of highly correlated biomarkers within a model may reduce classification performance, unnecessarily inflate associated costs of a diagnosis or a screening, and reduce the feasibility of clinical translation. Kingwell et al. [115] proposed an unsupervised feature-reduction approach by estimating the degree of correlation in a lipid data set using the VAT-generated MST, which helps reduce feature redundancy in lipidomic biomarkers by limiting the number of highly correlated lipids while retaining informative features to achieve good classification performance for various clinical outcomes.

Gene-Expression Data

Microarrays are one of the latest breakthroughs in experimental molecular biology, and they allow monitoring of the expression levels of tens of thousands of genes in parallel. This field produces huge amounts of valuable data [116], but the analysis and handling of such data are major bottlenecks in the utilization of microarray analysis. Keller et al. [117] were the first to use VAT on gene ontology (GO) data. They built similarity relations on pairs of terms that are used in the GO as linguistic descriptors of genes and gene products. The VAT algorithm was then used to discover the tendencies of groups of gene products to cluster them together.

Along similar lines, Kim et al. [118] proposed a VAT-based method they call *user-interacted cluster*. The method presents the RDI as basic information for user interaction because it helps an operator visually grasp the clustering tendency of a given data set. Havens et al. [119] proposed a methodology to couple the results of a microarray experiment with the GO annotations of each gene to produce aggregate relational data. The two relational matrices, one derived from a fuzzy GO similarity measure and the other derived from the microarray data using a statistical similarity measure, are then combined and used as an input to the non-Euclidean relational fuzzy c-means clustering algorithm [120]. Then, a validity measure called *correlation cluster validity* (CCV) is used to validate the resulting clusters in the relational data.

Figure 17 illustrates the methodology proposed in [119], which was applied to a selected set of *Arabidopsis* (a leafy plant) genes in the presence of insect feeding and wounding stress. This framework was extended for very large gene-expression data sets (e.g., a human genome data set consisting of approximately 30,000 genes, which would produce a $30{,}000 \times 30{,}000$ distance matrix) by Popescu et al. [121]. The authors extend the CCV algorithm to a new validity measure: extension to correlation cluster validity, which consists of two steps: sampling of the large matrix,

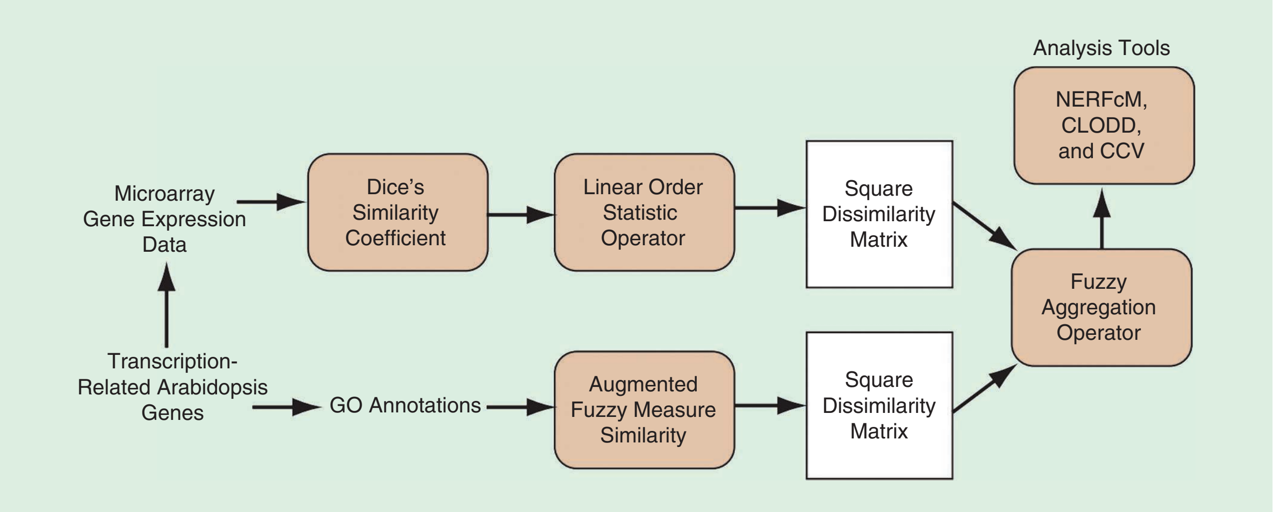

Analysis Tools
NERFcM, CLODD, and CCV
Microarray Gene Expression Data
Dice's Similarity Coefficient
Linear Order Statistic Operator
Square Dissimilarity Matrix
Fuzzy Aggregation Operator
Transcription-Related Arabidopsis Genes
GO Annotations
Augmented Fuzzy Measure Similarity
Square Dissimilarity Matrix

Figure 17. A fuzzy cluster analysis framework for GO data proposed by Havens et al. [119].

followed by estimation of the number of clusters applying CCV to the sampled data.

To cluster real-life benchmark gene-expression data, a novel interactive genetic-algorithm-based, multiobjective approach was proposed by Mukhopadhyay et al. [122] that simultaneously finds the clustering solution and evolves the set of validity measures that are to be optimized. The proposed method interactively takes the input from a human decision maker based on the VAT-based visualization tool and adaptively learns from that input to obtain the final set of validity measures along with the final clustering result.

Monitoring System for Older Individuals

With the significant increase in the population of older individuals in developed countries and the limited number of care centers, the concept of *aging in place* (*AIP*) has gained significant attention. AIP revolves around the notion of independent or partially assisted living and the ability to continuously receive any necessary support for growing needs. For successful implementation of AIP projects, personalizing the care of older individuals through environmental monitoring is essential.

Sledge et al. [123] established a framework for recognizing temporal trends in feature data extracted from passive sensors (e.g., infrared motion and pneumatic bed sensors; bed restlessness, pulse, and respiration sensors) used to monitor individuals. The GNGC algorithm was used for temporal clustering to assign different activity names to different sensor measurement profiles, and the VAT image was used to visualize the changing cluster structure of the temporal data stream.

As a part of passive fall-risk-assessment research in home environments, Banerjee et al. [124] presented a method to identify older residents at risk by using features extracted from their gait information from a single-depth camera. The VAT algorithm was used to determine the number of clusters, which was then used as an input to the PCM clustering technique [125]. The analysis helps detect changes in gait patterns, which can be used to analyze fall risk for older residents by passively observing them in their home environments. Li et al. [126] proposed an acoustic fall-detection system called *acoustic-FADE* that employs an eight-microphone circular array to automatically detect falls. The iVAT algorithm was used to analyze the relationship between fall and nonfall acoustic signatures in conjunction with the nearest-neighbor-based distance to find and remove the most challenging false alarms based on an efficient feature-selection technique.

Natural-Language Processing

Document and word clustering are well studied problems in the natural-language-processing community. Most algorithms cluster documents and words separately but not simultaneously. Dhillon [92] proposed a novel algorithm to cluster documents and words simultaneously as a bipartite graph-partitioning problem solved using spectral techniques. However, since most spectral clustering techniques require the number of clusters to seek as an input, Liu and Lu [127] explored a VAT-based method for determining the number of clusters present in the given data set for coclustering documents and words. It includes three main steps. First, generate a VAT image of the input matrix, which is produced by spectral coclustering documents and words. Next, use some common image-processing techniques, such as a grayscale morphological operation to filter the VAT image. Finally, the cluster number is estimated by computing the eigengap of the grayscale matrix of the filtered image.

In the area of automated support for argument reconstruction from natural-language texts, Winkels et al. [128], [129] investigated several possibilities to support a manual process of extracting arguments, which is a nontrivial task and requires extensive training and expertise. They used natural-language processing to classify pieces of text as either argumentative or nonargumentative and clustered the text fragments in the hope that these clusters would contain similar arguments. The VAT image was used to assess clustering tendency in the data set; although it showed some small clusters, it was inconclusive in suggesting major cluster structure in the data. This prompting the authors to conclude that the analysis cannot go far without an extensive pretagged corpus.

Building Linguistic Summaries From the Sensor Data

As information technology advances, more and more data are created, stored, and analyzed. However, this vast mountain of data is beyond human cognitive capabilities and comprehension skills. Hence, methods to summarize data and analyze these summaries are becoming increasingly important. Several approaches for linguistic summarization have been proposed in the literature, which generates linguistic summaries from sensor data so that people can read and take appropriate action. For instance, summaries of sensor data on older residents in independent-living facilities, including nighttime motion activity and restlessness while lying in bed, provide indications of potential abnormal conditions [130], [131].

As the number of sensors grows, so does the complexity and size of the set of linguistic descriptions. Hence, it is necessary to perform some automated analysis to condense this information. A set of papers by Wilbik et al. [132]–[134] develop an approach to generating linguistic prototypes from a group of time blocks that represent a normal condition from a care environment for older individuals. Then, the set of summaries for new time blocks is compared to the prototypes to flag anomalous conditions, thereby reducing the burden on the human. Wilbik et al. developed novel distance measures between linguistic summaries and use VAT/iVAT to assess and validate cluster structure in the linguistic summaries, which allows for the creation of linguistic prototypes from clusters of summaries over some temporal range. Anomalies are detected as observations that are considerably different from the linguistic prototypes in a moving temporal window.

Web User Data Analysis

The World Wide Web generates a humongous amount of data in the form of weblogs, user activity, browsing preference, social media, user-generated content, and so on. Web analytics is the measurement, collection, analysis, and reporting of web data for purposes of understanding and optimizing web usage. However, web analytics is not just a process for measuring web traffic; it can be used as a tool for business and market research and to assess and improve the effectiveness of a website. Various researchers have used some members of the VAT family of algorithms to analyze web-generated data to solve different problems. We turn to this application next.

Modeling User Behavior

Clustering web sessions to identify visitors' choices while browsing web pages is an important problem in web mining. The sequence of pages viewed by a user in a particular time frame (i.e., the session) captures his/her interest in a specific topic. The clustering of these sessions can be used to provide customized services. Chakraborty and Bandyopadhyay [135] and Sisodia et al. [136] explored the use of clustering web sessions to provide customized services to users with similar interests. The VAT algorithm was used to visualize the clustering tendency of these data, which was later fed as an input to the actual clustering algorithm.

Sun et al. [137] worked on the problem of detecting threats to the security, privacy, and integrity of computer networks and infrastructure from insiders—those who have (or had) authorized access to an organization's network, system, or data and intentionally exceeded or misused that access in a manner that negatively affected the confidentiality, integrity, or availability of the organization's information or information systems. The authors modeled system users' behavior and developed fast and efficient techniques to predict and detect insider threats. A necessary step for this task is to understand legitimate resource-usage access patterns, which can help identify abnormal or suspicious user behavior. Cluster analysis based on visual assessment using the VAT algorithm enabled them to detect communities of users based on the projects they access and the networks they use. Based on normal behavior patterns characterized by different clusters, abnormal behavior, detected by observing a deviation from normal user behavior, provides a pathway to developing an insider-threat detection system.

An important component of web design is user privacy because people are disclosing more and more personal information on online platforms. However, there is a problematic gap between existing online privacy controls and actual user disclosure behavior, which motivates researchers to focus on the design and development of intelligent privacy controls. Intelligent controls decrease the burden of privacy decision making and generate user-tailored privacy suggestions. The first necessary step is to analyze user privacy preferences. Kaskina [138] used VAT to assess clustering tendency in this context and then applied a fuzzy clustering approach to a real-world data set collected from a political platform. The fuzzy membership degree values were used to calculate more precise personalized privacy suggestions.

Fraud Detection

On social media networks, automated social agents (i.e., bots) are increasingly becoming a problem. Fraud in bot messaging, email spam, opinion spam, and so on, is a major threat to the credibility of web-based services and applications. Spam or unwanted email is one of the potential issues of Internet security, and classifying user emails correctly from penetration of spam is an important issue for antispam researchers.

Islam and Chowdhury [139] presented a spam classification technique using a clustering approach to categorize the features. They used the VAT algorithm to assess the extracted features and then passed the information into a classification engine (consisting of tree-based classifiers, nearest-neighbor algorithms, statistical algorithms, and so on). Cornelissen et al. [140] proposed that the social network topology of a user would be sufficient to determine whether the user is an automated agent or a human. They tested their conjecture on a publicly available data set containing users on Twitter labeled as either automated social agents or humans. The VAT algorithm was used to determine the best distance measure, which provides the best performance in classifying users.

Graph Data Analysis

A graph represents data consisting of nodes (representing objects with certain attributes) and the relationship between different nodes (represented by edges between pairs of nodes). Common examples of graph data include social (people–people relationship), coauthorship, road, power, and water networks, among others. Analyzing graphs is useful for determining general trends, relating the results of an experiment to the hypothesis, and formulating hypotheses for future investigations. Four widely used types of graph analytics include path, connectivity, community, and centrality analyses. In this section, we describe some papers that have applied a member of the VAT family of algorithms to various graph-data analysis problems.

Community Detection

Community detection is one of the most popular topics of modern network science. Communities, or clusters, are usually thought of as groups of vertices that have a higher probability of being connected to each other than to members of other groups. However, identifying a community is an ill-defined problem. There are no universal protocols for the fundamental ingredients, such as the definition of the community itself, or for other crucial issues, including the validation of algorithms and comparison of their performances.

Yang et al. [141] were the first to use the VAT algorithm to detect communities in a graph. The application of VAT to graph data is not straightforward since there is not a general meaningful distance in a graph. The authors introduced a new distance between nodes to measure the dissimilarity between nodes and obtain the distance matrix, which was then reordered using VAT to detect the community structure hidden in complex networks.

An important challenge in many graph clustering applications is that the clusters are not crisp (graph nodes may be partially associated with several clusters), leading to fuzzy clusters in graphs. Runkler and Bezdek [142] proposed the use of relational fuzzy clustering—more specifically, NERF c-means—to the relational data (obtained from the adjacency matrix of a graph). The clustering results were visualized using VAT and were illustrated on Zachary's karate-club benchmark data [27].

Papers by Havens et al. [143] and Su and Havens [144], [145] used various approaches, such as genetic algorithms and fuzzy modularity maximization for fuzzy community detection in social network graph data. The Newman–Girvan (NG) modularity function that measures how vertices in a community share more edges than expected in a randomized network were used as a cluster validity function. All of these papers used VAT, iVAT, and SpecVAT to visualize the results of various clustering approaches.

Ganji et al. [146] proposed a generalized modularity measure called *GM*, which has a more sophisticated interpretation of vertex similarity than vertex similarity as measured by the NG modularity function. GM takes into account the number of longer paths between vertices, compared to what would be expected in a randomized network, something that the NG modularity function does not consider. The VAT algorithm was used to illustrate how well-generalized modularity can reveal the underlying community structures in real-world graph data.

Visualizing Networks

Visualization of small-world networks is challenging owing to the large size of the data and their property of being "locally dense but globally sparse." Generally, networks are represented using graph layouts and images of adjacency matrices, which have shortcomings of occlusion and spatial complexity in direct form. Parveen and Sreevalsan-Nair [29] proposed a technique to enable effective and efficient visualization of small-world networks in the similarity space, as opposed to the attribute space, using a similarity matrix representation. They used VAT seriation to perform multilevel clustering on the matrix form and visualize a series of similarity matrices from the same data set using parallel-sets-like representation.

IoT and Smart Cities

The IoT infrastructure for the creation of smart cities consists of Internet-connected sensors, devices, and citizens. This IoT infrastructure generates an enormous amount of data in the form of city-scale physical measurements and public opinions, constituting big data. Smart cities aim to efficiently use this wealth of data to manage and solve the problems faced by modern cities for better decision making. Interpreting the massive amount of smart-city-generated big data to create actionable knowledge is a challenging task. Many researchers have utilized various algorithms belonging to the VAT family to analyze smart-city-generated data to gain actionable insights from them as discussed in the next section.

Smart City Urban Mobility

In an urban environment, high-quality mobility is a necessity for the success of other sectors and the creation of jobs, and it plays a key role in cultivating an attractive environment for residents and businesses. With the increase in urban population, ensuring fast, efficient, reliable, and cheap mobility to urban residents is a challenge for city authorities. However, with advances in the digital IoT infrastructure being deployed across cities and the data collected from them, novel techniques and technologies are being developed to improve urban mobility.

The clustering of taxi GPS mobility data helps with understanding the spatiotemporal dynamics for the applications of urban planning and transportation. Kumar et al. [147] clustered the origin–destination pairs of the passenger taxi rides using a hybrid algorithm consisting of clusiVAT sampling and DBSCAN to provide useful insights about city mobility patterns, urban hot spots, road-network usage, and general patterns of the crowd movement in the city of Singapore.

There is a growing interest in the problem of extracting useful information from massive trajectory data sets derived by various sensing methods. Understanding patterns of pedestrian movement is useful in applications, such as pedestrian-flow management, public security, and safety. Extracting pedestrian movement patterns and determining anomalous regions/periods is a useful data-mining task to be performed on the massive trajectory data sets generated by the smart city IoT infrastructure.

Li and Leckie [148] applied contour maps and iVAT to visually identify and analyze areas/periods with anomalous distributions of pedestrian flows. Contour maps are adopted as the visualization method of the origin–destination flow matrix to describe the distribution of pedestrian movement in terms of entry/exit areas. By transforming the origin–destination flow matrix into a dissimilarity matrix, the iVAT algorithm is used to visually cluster the most popular and related areas.

Kumar et al. [149] proposed a novel application of VAT-based clustering algorithms (VAT, iVAT, and clusiVAT) for trajectory analysis. They introduced a new clustering-based anomaly detection framework named iVAT+ and clusiVAT+ and used it for trajectory anomaly detection. Their approach is based on partitioning the VAT-generated minimum spanning tree using an efficient thresholding scheme. Trajectories are classified as normal or anomalous based on the number of paths in the clusters. Experiments on the trajectories of vehicles and pedestrians from a parking lot scene from the real-life Massachusetts Institute of Technology trajectories data set showcase the ability of the proposed method to find natural and informative trajectory clusters and anomalies.

Another important type of trajectory data collected in a smart city framework is vehicular trajectories, especially public transport, such as buses, taxis, and so on, Analysis of large-scale vehicle trajectories is important for understanding urban traffic patterns, particularly for optimizing public transport routes and frequencies and improving the decisions made by authorities. Cluster analysis is a fundamental challenge in trajectory mining, but existing trajectory clustering algorithms are not well suited to large numbers of trajectories in a city road network because of inadequate distance measures between two trajectories.

Kumar et al. [85], [150] proposed a novel Dijkstra-based dynamic time warping (DTW) distance measure called *trajDTW*, which is suitable for large numbers of overlapping trajectories in a dense road network. They also developed a novel fast-clusiVAT algorithm that can suggest the number of clusters in a trajectory data set and identify and visualize the trajectories belonging to each cluster much faster than clusiVAT. Empirical experiments conducted on a large-scale taxi trajectory data set consisting of 3.28 million trajectories obtained from the GPS traces of 15,061 taxis in Singapore for one month suggest several trajectory clusters spanning the major expressways of Singapore. For each cluster, this scheme provides a time-based distribution of trajectories that affords insights into how urban mobility patterns change with the time of day.

Taking this trajectory-analysis task a step further toward prediction, Rathore et al. [151] proposed a scalable-clustering and Markov-chain-based hybrid framework, called *Traj-clusiVAT-based trajectory prediction*, for both short- and long-term trajectories that can handle a large number of overlapping trajectories in a dense road network.

Smart Grid

The rollout of electricity grid assets with advanced communications capabilities enables new ways to steal energy, such as false data attacks and remote meter disconnection. On the other hand, data communicated by these devices have the potential to improve utility companies' abilities to combat fraud through computational intelligence techniques. Viegas and Vieira [152] proposed a clustering-based novelty detection scheme to uncover electricity theft. The scheme starts by extracting easily interpreted consumption indicators from data collected by smart meters. Fuzzy clustering is then used to capture the structure of the data that consists of indicators from benign consumers. The VAT algorithm is used before clustering to analyze the possibility of data structure characterized by multiple clusters. The extracted clusters provide the basis for a distance-based novelty-detection model to uncover abnormal data sent by consumers.

Time Series Data Analysis

Time series data (a measurement of sensor values obtained from a physical process over some time) is a common form of data. To study the temporal behavior of an environmental system, scientists need to detect positions with similar temporal dynamics in large sets of time series. A well-established approach to quantifying the number and duration of recurrent states is recurrence quantification analysis (RQA) [153].

Sips et al. [154] address the scientific question of whether the clustering of time series based on their RQA measures produces a clustering structure that is interpretable by human experts. They used iVAT to visualize the time series (each time series is represented by 16 different RQA measures, which are used to calculate a Euclidean distance matrix) and found that the iVAT visualization of cluster structure was interpretable and consistent with the clustering structure based on the expert opinion.

Iredale et al. [155] proposed a novel shape-based measure of similarity, which is invariant under uniform time shift and uniform amplitude scaling. Using this measure to calculate the distance matrix, the authors used the VAT algorithm to assess large time-series data sets and demonstrated its advantages in terms of complexity and propensity for implementation in a distributed computing environment. In the field of neuroscience research, Mahallati et al. [156] experimented on a variety of distance measures—Euclidean distance, correlation distance, DTW, and shape-based distance—in the recording of extracellular action potential (spike) waveforms generated by neuronal activity, using iVAT to distinguish the number of units present within recordings from a single electrode.

WSNs for Environmental Monitoring

Understanding the behavior of complex ecosystems requires analysis of detailed observations of an environment under a range of different conditions. WSNs provide a flexible platform to collect data for environmental modeling. A WSN comprises a set of low-powered nodes, each with its sensors, power supply, CPU, and radio transceiver, which can self-organize into a network for collecting and reporting sensor measurements. Although WSNs provide raw data from the monitored environment, an open challenge is how to build and utilize models of "normal" behavior and "interesting" (anomalous) events from that data.

A series of papers by Bezdek et al. [157], [158], Moshtaghi et al. [159], and Rajasegarar et al. [160] used hyperellipsoidal models and VAT/iVAT visualization to detect anomalies in a WSN for environmental monitoring. The proposed approach generates a set of hyperellipsoids to summarize the data generated by the WSN. These papers proposed three measures of similarity between pairs of ellipsoids (compound, transformation energy, and Bhattacharya coefficient similarity) to convert model ellipsoids into dissimilarity data, which was then fed to VAT/iVAT to discover clusters in the data. Finally, the authors used various clustering algorithms (SL, CLODD, and so on) to extract normal clusters and anomalies from the input data. This framework was empirically evaluated on a variety of data sets, viz., data collected by a WSN installed at the Intel Berkeley Research Lab to measure parameters (e.g., humidity, temperature, light, and so on) and the Heron Island WSN in the Great Barrier Reef, among others.

Miscellaneous Applications

Humans and Society

In the psychology of motivation, balance theory is a theory of attitude change proposed by Fritz Heider [161]. Heider's structural balance theory explains social processes and is used to account for social actors' attitudes toward one another. Notsu et al. [162] propose a new social value emergence model in the form of an agent-based simulation model. In this model, structural balance theory is used to explain feelings, attitudes, and beliefs. Each agent tries to reach balanced states and communicates with others. The VAT algorithm was used to understand the agent group's macrolevel mechanism and represents the social groups to which different agents belong. As an extension to their previous work, Notsu et al. [163] adapted VAT to a network model by reinterpreting positive/negative relationships in naive psychology as dissimilarity, such as "near" or "similar"/"distant" or "unlike" to improve mutual understanding among people.

Human Geography

Human geography is concerned with how human-related factors, such as cultural, economic, religious, and political issues, influence the spatial behavior of individuals and groups of people. The study of human geography is important for several application areas, including, for example, preparing for disaster response and relief, identifying medically underserved areas, and so on. Buck et al. [164] proposed a VAT-based approach to summarizing various human-related factors that can be mapped for a geographic region for visualization and easy understanding by domain experts. To combine various human geographic factors for a region, a human geography data cube (consisting of spatial dimensions of the region and one human-geographic dimension) was created. Different data cubes (belonging to different spatial locations) were then visualized using VAT to present a complete picture of the spatial distribution of various human-related factors and their interactions with each other.

Instance-Based Machine Learning

In instance-based machine learning (e.g., nearest-neighbor classifiers), algorithms often suffer from high storage requirements because of the large number of training instances. This results not only in large computer memory usage and long response time but also, very often, in oversensitivity to noise, which degrades algorithmic performance. To tackle such problems, various instance reduction algorithms have been developed that remove noisy and redundant patterns. Inspired by the data seriation approach of the VAT algorithm, Nikolaidis et al. [165] introduced a new approach: instance seriation for prototype abstraction, which is a data-condensation method that generates a new set of prototypes. This helps reduce the storage requirements of instance-based algorithms and make them resistant to data noise.

Optimization

For continuous multiobjective optimization problems, there are an infinite number of Pareto optimal solutions. However, many multiobjective evolutionary algorithms [166] fail to find and preserve all of the multimodal solutions in the nondominated-solutions set. They can identify only one set of decision vectors out of the multimodal solutions and terminate at any one global optimum out of the multiple global optima present in the multiobjective multimodal problems. Finding all multimodal solutions would allow the decision maker greater flexibility when choosing between solutions. For example, in chemical process optimization, the decision maker would want to know about different temperature settings for which the process can deliver the same results.

Kudikala et al. [167] presented an extended version of the Pareto estimation method, which can be used to increase the number of multimodal solutions. The method uses VAT to identify and separate different clusters in the space of decision variables, which correspond to the multimodal Pareto optimal solutions. Then, Pareto estimation is employed for these individual clusters, which increases the density of available multimodal solutions in multiobjective problems.

Predicting Ground Vibrations in Mines

Ground vibrations, measured in terms of peak particle velocity (PPV) is one of the hazard effects induced by blasting operations in open-pit mines, which can affect surrounding structures, particularly the stability of benches and slopes in open-pit mines, and their impact on underground water, railway, highway, and so on can be puzzling for neighboring communities. Therefore, controlling, predicting, and mitigating the effects of blast-induced PPV is desirable.

Nguyen et al. [168] developed a new clustering-based computational model for predicting blast-induced PPV. To assess whether the data set is suitable for the use of clustering algorithms, the VAT algorithm was applied, and once the data were deemed suitable for clustering, a novel hybrid artificial-intelligence model based on the hierarchical *k*-means (HKM) clustering algorithm and an artificial neural network (ANN)—namely, an HKM-ANN model—was developed for predicting blast-caused PPV in open-pit mines.

Master's and Ph.D. Theses Inspired by the VAT Family of Algorithms

The VAT family of algorithms has been a part of the research of various master's and Ph.D. students, which, in turn, has also contributed to many of the algorithms and applications described in this article. Following is the (chronological) list of dissertations that have used or contributed to the VAT family of algorithms:

- "Extracting Textual Information From Images and Videos for Automatic Content-Based Annotation and Retrieval," 2007 [169]
- "Visual Data Analysis in Air Traffic Management," 2007 [170]
- "Information Visualization Techniques for Metabolic Engineering," 2007 [171]
- "A System for Change Detection and Human Recognition in Voxel Space Using Stereo Vision," 2010 [172]
- "Clustering in Relational Data and Ontologies," 2010 [173]
- "A Study on Parallel Versus Sequential Relational Fuzzy Clustering Methods," 2011 [174]
- "Sparse and Discriminative Clustering for Complex Data," 2012 [175]
- "Visualization of Transformation of Graphs Based on Similarity Functions," 2013 [176]
- "Support Vector Machine-based Fuzzy Systems for Quantitative Prediction of Peptide Binding Affinity," 2015 [177]
- "Experimental Study of Random Projections Below the JL Limit," 2015 [178]
- "Clustering von Recurrence Plots," 2015 [179]
- "Big Data Clustering for Smart City Applications," 2016 [180]
- "Enabling Automatic Creation of Virtual Services for Service Virtualisation," 2016 [181]
- "Big Data Cluster Analysis and Its Applications," 2018 [182].

Conclusion

This survey article attempts to capture and summarize work related to the theory and applications of the VAT/iVAT family of models and algorithms. Inevitably, we will have missed some articles that belong here, but it is impossible to know about every work related to this type of cluster heat map and its widespread applications. Readers may wonder whether subjective analysis of the visual representation of numerical data, such as that offered by cluster heat maps, is really useful.

Perhaps a fitting conclusion to this contribution is a statement about the utility of data visualization made by Sir Ronald Fisher [183] almost 100 years ago: "The preliminary examination of most data are facilitated by the use of diagrams. Diagrams prove nothing, but bring outstanding features readily to the eye; they are therefore no substitute for critical tests as may be applied to the data, but are valuable in suggesting such tests, and in explaining conclusions founded upon them." This is the rationale for using methods such as VAT/iVAT and, more generally, other types of cluster heat maps that abound in the scientific literature. They are there to help us understand the structure in the data that we cannot see ourselves.

About the Authors

Dheeraj Kumar (dheeraj.kumar@ece.iitr.ac.in) is with the Department of Electronics and Communication Engineering, Indian Institute of Technology Roorkee.

James C. Bezdek (jcbezdek@gmail.com) is with the School of Computing and Information Systems, the University of Melbourne, Australia. He is a Life Fellow of the IEEE.

References

[1] "VAT family of algorithms," GitHub, San Francisco. Accessed on: Oct. 30, 2019. [Online]. Available: https://github.com/genuine-dheeraj/VAT_family_of_algorithms

[2] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recog. Lett.*, vol. 31, no. 8, pp. 651–666, 2010. doi: 10.1016/j.patrec.2009.09.011.

[3] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ: Prentice Hall, 1988.

[4] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. San Diego, CA: Academic Press, 1999.

[5] J. Bezdek, *A Primer on Cluster Analysis: Four Basic Methods That (Usually) Work*. Sarasota, FL: First Edition Design, 2017.

[6] B. Everitt, *Graphical Techniques for Multivariate Data*. New York: North-Holland Press, 1978.

[7] A. Kalogeratos and A. Likas, "Dip-means: An incremental clustering method for estimating the number of clusters," in *Proc. Int. Conf. Neural Information Processing Systems*, vol. 2. Red Hook, NY: Curran Associates Inc., 2012, pp. 2393–2401.

[8] M. O. Ahmed and G. Walther, "Investigating the multimodality of multivariate data with principal curves," *Comput. Stat. Data Anal.*, vol. 56, no. 12, pp. 4462–4469, 2012. doi: 10.1016/j.csda.2012.02.020.

[9] B. Hopkins and J. G. Skellam, "A new method for determining the type of distribution of plant individuals," *Ann. Bot.*, vol. 18, no. 2, pp. 213–227, 1954. doi: 10.1093/oxfordjournals.aob.a083391.

[10] A. Adolfsson, M. Ackerman, and N. C. Brownstein, "To cluster, or not to cluster: An analysis of clusterability methods," *Pattern Recog.*, vol. 88, pp. 13–26, 2019. doi: 10.1016/j.patcog.2018.10.026. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320318303777

[11] J. C. Bezdek and R. J. Hathaway, "VAT: A tool for visual assessment of (cluster) tendency," in *Proc. Int. Joint Conf. Neural Networks*, May 2002, vol. 3, pp. 2225–2230. doi: 10.1109/IJCNN.2002.1007487.

[12] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389–1401, Nov. 1957. doi: 10.1002/j.1538-7305.1957.tb01515.x.

[13] R. J. Hathaway, J. C. Bezdek, and J. M. Huband, "Scalable visual assessment of cluster tendency for large data sets," *Pattern Recog.*, vol. 39, no. 7, pp. 1315–1324, 2006. doi: 10.1016/j.patcog.2006.02.011.

[14] L. Wang, U. T. V. Nguyen, J. C. Bezdek, C. A. Leckie, and K. Ramamohanarao, "iVAT and aVAT: Enhanced visual analysis for cluster tendency assessment," in *Advances in Knowledge Discovery and Data Mining*. M. J. Zaki, J. X. Yu, B. Ravindran, and V. Pudi, Eds. Berlin: Springer-Verlag, 2010, pp. 16–27.

[15] T. C. Havens and J. C. Bezdek, "An efficient formulation of the improved visual assessment of cluster tendency (iVAT) algorithm," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 813–822, May 2012. doi: 10.1109/TKDE.2011.33.

[16] Z. Zhou, L. Zhong, and L. Wang, "Locally incremental visual cluster analysis using Markov random field," *Neurocomputing*, vol. 136, pp. 49–55, July 2014. doi: 10.1016/j.neucom.2014.01.032.

[17] C. Zhong, X. Yue, and J. Lei, "Visual hierarchical cluster structure: A refined co-association matrix based visual assessment of cluster tendency," *Pattern Recog. Lett.*, vol. 59, pp. 48–55, July 2015. doi: 10.1016/j.patrec.2015.03.007.

[18] L. Wang, X. Geng, J. Bezdek, C. Leckie, and R. Kotagiri, "SpecVAT: Enhanced visual cluster analysis," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 638–647. doi: 10.1109/ICDM.2008.18.

[19] L. Wang, X. Geng, J. Bezdek, C. Leckie, and R. Kotagiri, "Enhanced visual analysis for cluster tendency assessment and data partitioning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1401–1414, Oct. 2010. doi: 10.1109/TKDE.2009.192.

[20] D. Wishart, "Mode analysis: A generalization of nearest neighbor which reduces chaining effects," in *Numerical Taxonomy*, A. J. Cole, Ed. London: Academic Press, 1969, pp. 282–311.

[21] D. Kumar, Z. Ghafoori, J. C. Bezdek, C. Leckie, K. Ramamohanarao, and M. Palaniswami, "Dealing with inliers in feature vector data," *Int. J. Uncertain. Fuzz. Knowl.-Based Syst.*, vol. 26, no. S2, pp. 25–45, 2018. doi: 10.1142/S021848851840010X.

[22] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, no. 3, pp. 403–410, 1990. doi: 10.1016/S0022-2836(05)80360-2.

[23] S. Sampson, "A novitiate in a period of change. An experimental and case study of social relationships," Ph.D. thesis, Dept. Soc., Cornell Univ., Ithaca, NY, 1968.

[24] R. L. Breiger, S. A. Boorman, and P. Arabie, "An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling," *J. Math. Psychol.*, vol. 12, no. 3, pp. 328–383, 1975. doi: 10.1016/0022-2496(75)90028-0.

[25] T. C. Havens, J. C. Bezdek, C. Leckie, and M. Palaniswami, "Extension of iVAT to asymmetric matrices," in *Proc. IEEE Int. Conf. Fuzzy Systems*, July 2013, pp. 1–6. doi: 10.1109/FUZZ-IEEE.2013.6622300.

[26] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications* (Structural Analysis in the Social Sciences). Cambridge, U.K.: Cambridge Univ. Press, 1994.

[27] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, 1977. doi: 10.1086/jar.33.4.3629752.

[28] L. A. F. Park, J. C. Bezdek, C. Leckie, R. Kotagiri, J. Bailey, and M. Palaniswami, "Visual assessment of clustering tendency for incomplete data," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3409–3422, Dec. 2016. doi: 10.1109/TKDE.2016.2608821.

[29] S. Parveen and J. Sreevalsan-Nair, "Visualization of small world networks using similarity matrices," in *Big Data Analytics* (Lecture Notes in Computer Science, vol. 8302), V. Bhatnagar and S. Srinivasa, Eds. Cham, Switzerland: Springer-Verlag, 2013, pp. 151–170.

[30] O. Boruvka, "O jistém problému minimálním (About a certain minimal problem) (in Czech, German summary)," *Práce Mor. Prírodoved. Spol. v Brne III*, vol. 3, no. 3, pp. 37–58, 1926.

[31] V. Vineet, P. Harish, S. Patidar, and P. J. Narayanan, "Fast minimum spanning tree for large graphs on the GPU," in *Proc. Conf. High Performance Graphics*, 2009, pp. 167–171. doi: 10.1145/1572769.1572796.

[32] T. Meng and B. Yuan, "Parallel visual assessment of cluster tendency on GPU," in *Advances in Knowledge Discovery and Data Mining* (Lecture Notes in Computer Science, vol. 10235), J. Kim, K. Shim, L. Cao, J. G. Lee, X. Lin, and Y. S. Moon, Eds. Cham, Switzerland: Springer-Verlag, 2017, pp. 429–440.

[33] T. Meng and B. Yuan, "Parallel edge-based visual assessment of cluster tendency on GPU," *Int. J. Data Sci. Anal.*, vol. 6, no. 4, pp. 287–295, Dec. 2018. doi: 10.1007/s41060-018-0100-7.

[34] T. C. Havens, J. C. Bezdek, J. M. Keller, M. Popescu, and J. M. Huband, "Is VAT really single linkage in disguise?" *Ann. Math. Artif. Intell.*, vol. 55, nos. 3–4, pp. 237–251, Aug. 2009. doi: 10.1007/s10472-009-9157-2.

[35] S. Mahallati, J. C. Bezdek, M. R. Popovic, and T. A. Valiante, "Cluster tendency assessment in neuronal spike data," *PLoS One*, vol. 14, no. 11:e0224547, 2019. doi: 10.1371/journal.pone.0224547.

[36] R. J. Hathaway, J. M. Huband, and J. C. Bezdek, "Kernelized non-Euclidean relational fuzzy c-means algorithm," in *Proc. Int. Conf. Fuzzy Systems*, May 2005, pp. 414–419. doi: 10.1109/FUZZY.2005.1452429.

[37] I. Sledge, J. Bezdek, T. Havens, and J. Keller, "A relational dual of the fuzzy possibilistic c-means algorithm," in *Proc. Int. Conf. Fuzzy Systems*, July 2010, pp. 1–9. doi: 10.1109/FUZZY.2010.5584846.

[38] D. T. Anderson, J. M. Keller, O. Sjahputera, J. C. Bezdek, and M. Popescu, "Comparing soft clusters and partitions," in *Proc. IEEE Int. Conf. Fuzzy Systems*, June 2011, pp. 924–931. doi: 10.1109/FUZZY.2011.6007474.

[39] A. Vathy-Fogarassy, A. Kiss, and, and J. Abonyi, "Improvement of Jarvis-Patrick clustering based on fuzzy similarity," in *Applications of Fuzzy Sets Theory* (Lecture Notes in Computer Science, vol. 4578), F. Masulli, S. Mitra, and G. Pasi, Eds. Berlin: Springer-Verlag, 2007, pp. 195–202.

[40] K. R. Prasad and B. E. Reddy, “An efficient visualized clustering approach (VCA) for various datasets,” in *Proc. IEEE Int. Conf. Signal Processing, Informatics, Communication and Energy Systems*, Feb. 2015, pp. 1–5. doi: 10.1109/SPICES.2015.7091373.

[41] L. E. B. d. Silva and D. C. Wunsch, “A study on exploiting VAT to mitigate ordering effects in fuzzy art,” in *Proc. Int. Joint Conf. Neural Networks*, July 2018, pp. 1–8. doi: 10.1109/IJCNN.2018.8489724.

[42] R. J. Hathaway and J. C. Bezdek, “Visual cluster validity for prototype generator clustering models,” *Pattern Recog. Lett.*, vol. 24, nos. 9–10, pp. 1563–1569, 2003. doi: 10.1016/S0167-8655(02)00395-1.

[43] J. C. Bezdek and R. J. Hathaway, “Visual cluster validity (VCV) displays for prototype generator clustering methods,” in *Proc. IEEE Int. Conf. Fuzzy Systems*, May 2003, vol. 2, pp. 875–880. doi: 10.1109/FUZZ.2003.1206546.

[44] R. L. Ling, “A computer generated aid for cluster analysis,” *Commun. ACM*, vol. 16, no. 6, pp. 355–361, June 1973. doi: 10.1145/362248.362263.

[45] Y. Ding and R. F. Harrison, “Relational visual cluster validity (RVCV),” *Pattern Recog. Lett.*, vol. 28, no. 15, pp. 2071–2079, 2007. doi: 10.1016/j.patrec.2007.06.002.

[46] S. Gunnersen, K. Smith-Miles, and V. Lee, “SpecVCMV: Improving cluster visualisation,” in *Proc. Annu. Conf. IEEE Industrial Electronics Society*, Nov. 2011, pp. 2255–2260. doi: 10.1109/IECON.2011.6119660.

[47] L. Zelnik-Manor and P. Perona, “Self-tuning spectral clustering,” in *Proc. 17th Int. Conf. Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2004, pp. 1601–1608.

[48] J. M. Huband and J. C. Bezdek, “VCV2: Visual cluster validity,” in *Computational Intelligence: Research Frontiers* (Lecture Notes in Computer Science, vol. 5050), J. M. Zurada, G. G. Yen, J. Wang, Eds. Berlin: Springer-Verlag, 2008, pp. 293–308.

[49] T. C. Havens, J. C. Bezdek, J. M. Keller, and M. Popescu, “Dunn’s cluster validity index as a contrast measure of VAT images,” in *Proc. Int. Conf. Pattern Recognition*, Dec. 2008, pp. 1–4. doi: 10.1109/ICPR.2008.4761772.

[50] J. C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973. doi: 10.1080/01969727308546046.

[51] J. M. Keller and I. J. Sledge, “A cluster by any other name,” in *Proc. Annu. Meeting of the North American Fuzzy Information Processing Society*, June 2007, pp. 427–432. doi: 10.1109/NAFIPS.2007.383877.

[52] I. J. Sledge, T. C. Havens, J. M. Huband, J. C. Bezdek, and J. M. Keller, “Finding the number of clusters in ordered dissimilarities,” *Soft Comput.*, vol. 13, no. 12, pp. 1125–1142, Oct. 2009. doi: 10.1007/s00500-009-0421-5.

[53] I. J. Sledge, J. M. Huband, and J. C. Bezdek, “(Automatic) cluster count extraction from unlabeled data sets,” in *Proc. Int. Conf. Fuzzy Systems and Knowledge Discovery*, Oct. 2008, pp. 3–13. doi: 10.1109/FSKD.2008.552.

[54] L. Wang, C. Leckie, K. Ramamohanarao, and J. Bezdek, “Automatically determining the number of clusters in unlabeled data sets,” *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 3, pp. 335–350, Mar. 2009. doi: 10.1109/TKDE.2008.158.

[55] P. Prabhu and K. Duraiswamy, “Enhanced dark block extraction method performed automatically to determine the number of clusters in unlabeled data sets,” *Int. J. Comput. Commun. Control*, vol. 8, no. 2, pp. 275–293, 2013. doi: 10.15837/ijccc.2013.2.308.

[56] P. Prabhu and K. Duraiswamy, “Enhanced VAT for cluster quality assessment in unlabeled datasets,” *J. Circuits Syst. Comput.*, vol. 21, no. 1, pp. 12500016, 2012. doi: 10.1142/S0218126612500016.

[57] Y. Hu and R. J. Hathaway, “An algorithm for clustering tendency assessment,” *WSEAS Trans. Math.*, vol. 7, no. 7, pp. 441–450, July 2008.

[58] Y. Hu and R. J. Hathaway, “Tendency curves for visual clustering assessment,” in *Proc. WSEAS Int. Conf. Applied Computing Conf.* Stevens Point, WI: WSEAS, 2008, pp. 274–279.

[59] Y. Hu, “VATdt: Visual assessment of cluster tendency using diagonal tracing,” *Am. J. Comput. Math.*, vol. 2, no. 1, pp. 27–41, 2012. doi: 10.4236/ajcm.2012.21004.

[60] T. C. Havens, J. C. Bezdek, J. M. Keller, and M. Popescu, “Clustering in ordered dissimilarity data,” *Int. J. Intell. Syst.*, vol. 24, no. 5, pp. 504–528, 2009. doi: 10.1002/int.20344.

[61] M. K. Pakhira and A. Dutta, “Determination of number of clusters using VAT images and genetic algorithms,” in *Proc. Int. Conf. Emerging Applications of Information Technology*, Feb. 2011, pp. 357–360. doi: 10.1109/EAIT.2011.53.

[62] M. K. Pakhira and A. Dutta, “Finding number of clusters using VAT image, PBM index and genetic algorithms,” in *Proc. Int. Conf. Intelligent Interactive Technologies and Multimedia*. New York: ACM, 2010, pp. 217–221. doi: 10.1145/1963564.1963601.

[63] M. K. Pakhira, S. Bandyopadhyay, and U. Maulikc, “Validity index for crisp and fuzzy clusters,” *Pattern Recog.*, vol. 37, no. 3, pp. 487–501, 2004. doi: 10.1016/j.patcog.2003.06.005.

[64] M. K. Pakhira and A. Dutta, “Computing approximate value of the PBM index for counting number of clusters using genetic algorithm,” in *Proc. Int. Conf. Recent Trends in Information Systems*, Dec. 2011, pp. 241–245. doi: 10.1109/ReTIS.2011.6146875.

[65] D. Laney, “3D data management: Controlling data volume, velocity, and variety,” META Group, Stamford, CT, Tech. Rep., Feb. 2001. [Online]. Available: http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf

[66] KDnuggets, “The 42 V’s of big data and data science.” Accessed on: May 1, 2019. [Online]. Available: https://www.kdnuggets.com/2017/04/42-vs-big-data-data-science.html

[67] P. Barnaghi, A. Sheth, and C. Henson, “From data to actionable knowledge: Big data challenges in the web of things,” *IEEE Intell. Syst.*, vol. 28, no. 6, pp. 6–11, Nov. 2013. doi: 10.1109/MIS.2013.142.

[68] J. M. Huband, J. C. Bezdek, and R. J. Hathaway, “Revised visual assessment of (cluster) tendency (reVAT),” in *Proc. IEEE Annu. Meeting of the Fuzzy Information Processing Society*, June 2004, pp. 101–104. doi: 10.1109/NAFIPS.2004.1336257.

[69] J. M. Huband, J. C. Bezdek, and R. J. Hathaway, “bigVAT: Visual assessment of cluster tendency for large data sets,” *Pattern Recog.*, vol. 38, no. 11, pp. 1875–1886, 2005. doi: 10.1016/j.patcog.2005.03.018.

[70] M. Johnson, L. Moore, and D. Ylvisaker, “Minimax and maximin distance designs,” *J. Stat. Plan. Inference*, vol. 26, no. 2, pp. 131–148, 1990. doi: 10.1016/0378-3758(90)90122-B.

[71] K. R. Prasad and B. E. Reddy, “Assessment of clustering tendency through progressive random sampling and graph-based clustering results,” in *Proc. IEEE Int. Advance Computing Conf.*, Feb. 2013, pp. 726–731. doi: 10.1109/IAdCC.2013.6514316.

[72] L. Wang, C. Leckie, R. Kotagiri, and J. Bezdek, “Approximate pairwise clustering for large data sets via sampling plus extension,” *Pattern Recog.*, vol. 44, no. 2, pp. 222–235, 2011. doi: 10.1016/j.patcog.2010.08.005.

[73] M. K. Pakhira, “Out-of-core assessment of clustering tendency for large data sets,” in *Proc. IEEE Int. Advance Computing Conf.*, Feb. 2010, pp. 29–33. doi: 10.1109/IADCC.2010.5423044.

[74] L. H. Trang, P. Van Ngoan, and N. Van Duc, “A sample-based algorithm for visual assessment of cluster tendency (VAT) with large datasets,” in *Future Data and Security Engineering* (Lecture Notes in Computer Science, vol. 11251), T. Dang, J. Küng, R. Wagner, N. Thoai, and M. Takizawa, Eds. Cham, Switzerland: Springer-Verlag, 2018, pp. 145–157.

[75] F. Ros and S. Guillaume, “ProTras: A probabilistic traversing sampling algorithm,” *Exp. Syst. Appl.*, vol. 105, pp. 65–76, Sept. 2018. doi: 10.1016/j.eswa.2018.03.052.

[76] H. Shao, P. Zhang, X. Chen, F. Li, and G. Du, “A hybrid and parameter-free clustering algorithm for large data sets,” *IEEE Access*, vol. 7, pp. 24,806–24,818, Feb. 19, 2019. doi: 10.1109/ACCESS.2019.2900260.

[77] I. Steponavičė, M. Shirazi-Manesh, R. J. Hyndman, K. Smith-Miles, and L. Villanova, “On sampling methods for costly multi-objective black-box optimization,” in *Advances in Stochastic and Deterministic Global Optimization* (Springer Optimization and Its Applications, vol. 107), P. Pardalos, A. Zhigljavsky, and J. Žilinskas, Eds. Cham, Switzerland: Springer-Verlag, 2016, pp. 273–296.

[78] P. Rathore, J. C. Bezdek, D. Kumar, S. Rajasegarar, and M. Palaniswami, "Approximate cluster heat maps of large high-dimensional data," in *Proc. Int. Conf. Pattern Recognition*, Aug. 2018, pp. 195–200. doi: 10.1109/ICPR.2018.8545519.

[79] T. C. Havens, J. C. Bezdek, and M. Palaniswami, "Scalable single linkage hierarchical clustering for big data," in *Proc. IEEE Int. Conf. Intelligent Sensors, Sensor Networks and Information Processing*, Apr. 2013, pp. 396–401. doi: 10.1109/ISSNIP.2013.6529823.

[80] D. Kumar, J. C. Bezdek, M. Palaniswami, S. Rajasegarar, C. Leckie, and T. C. Havens, "A hybrid approach to clustering in big data," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2372–2385, Oct. 2016. doi: 10.1109/TCYB.2015.2477416.

[81] D. Kumar, M. Palaniswami, S. Rajasegarar, C. Leckie, J. C. Bezdek, and T. C. Havens, "clusiVAT: A mixed visual/numerical clustering algorithm for big data," in *Proc. IEEE Int. Conf. on Big Data*, Oct. 2013, pp. 112–117. doi: 10.1109/BigData.2013.6691561.

[82] I. J. Sledge and J. M. Keller, "Growing neural gas for temporal clustering," in *Proc. Int. Conf. Pattern Recognition*, Dec. 2008, pp. 1–4. doi: 10.1109/ICPR.2008.4761768.

[83] A. Kianimajd et al., "Comparison of different methods of measuring similarity in physiologic time series," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11,005–11,010, 2017. doi: 10.1016/j.ifacol.2017.08.2479.

[84] T. Li, H. Dong, Y. Shi, and M. Dehmer, "A comparative analysis of new graph distance measures and graph edit distance," *Inf. Sci.*, vol. 403–404, pp. 15–21, Sept. 2017. doi: 10.1016/j.ins.2017.03.036.

[85] D. Kumar, H. Wu, S. Rajasegarar, C. Leckie, S. Krishnaswamy, and M. Palaniswami, "Fast and scalable big data trajectory clustering for understanding urban mobility," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 11, pp. 3709–3722, Nov. 2018. doi: 10.1109/TITS.2018.2854775.

[86] D. Kumar et al., "Adaptive cluster tendency visualization and anomaly detection for streaming data," *ACM Trans. Knowl. Discov. Data*, vol. 11, no. 2, pp. 1–40, Dec. 2016. doi: 10.1145/2997656.

[87] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD cup 99 data set," in *Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications*, July 2009, pp. 1–6. doi: 10.1109/CISDA.2009.5356528.

[88] J. Blackard and J. Denis, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables," *Comput. Electron. Agric.*, vol. 24, no. 3, pp. 131–151, 2000. doi: 10.1016/S0168-1699(99)00046-0.

[89] C. Meek, B. Thiesson, and D. Heckerman, "The learning-curve sampling method applied to model-based clustering," *J. Mach. Learn. Res*, vol. 2, pp. 397–418, Mar. 2002. doi: 10.1162/153244302760200678.

[90] M. Shindler, A. Wong, and A. Meyerson, "Fast and accurate k-means for large datasets," in *Proc. Int. Conf. Neural Information Processing Systems*. New York: Curran Associates Inc., 2011, pp. 2375–2383.

[91] P. Rathore, D. Kumar, J. C. Bezdek, S. Rajasegarar, and M. S. Palaniswami, "A rapid hybrid clustering algorithm for large volumes of high dimensional data," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 641–654, 2018. doi: 10.1109/TKDE.2018.2842191.

[92] I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2001, pp. 269–274. doi: 10.1145/502512.502550.

[93] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra, "Minimum sum-squared residue coclustering of gene expression data," in *Proc. 2004 SIAM Int. Conf. Data Mining*, pp. 114–125. doi: 10.1137/1.9781611972740.11.

[94] J. C. Bezdek, R. J. Hathaway, and J. M. Huband, "Visual assessment of clustering tendency for rectangular dissimilarity matrices," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 5, pp. 890–903, Oct. 2007. doi: 10.1109/TFUZZ.2006.889956.

[95] T. C. Havens, J. C. Bezdek, and J. M. Keller, "A new implementation of the co-VAT algorithm for visual assessment of clusters in rectangular relational data," in *Artificial Intelligence and Soft Computing* (Lecture Notes in Computer Science, vol. 6113), Berlin: Springer-Verlag, 2010, pp. 363–371.

[96] T. C. Havens and J. C. Bezdek, "A new formulation of the coVAT algorithm for visual assessment of clustering tendency in rectangular data," *Int. J. Intell. Syst.*, vol. 27, no. 6, pp. 590–612, 2012. doi: 10.1002/int.21539.

[97] K. Honda, T. Sako, S. Ubukata, and A. Notsu, "Visual assessment of co-cluster structure through cooccurrence-sensitive ordering," in *Proc. World Congr. Int. Fuzzy Systems Association*, June 2017, pp. 1–6. doi: 10.1109/IFSA-SCIS.2017.8023336.

[98] L. A. F. Park, J. C. Bezdek, and C. A. Leckie, "Visualization of clusters in very large rectangular dissimilarity data," in *Proc. Int. Conf. Autonomous Robots and Agents*, Feb. 2000, pp. 251–256. doi: 10.1109/ICARA.2000.4803948.

[99] K. R. Prasad, R. Nennuri, and T. V. V. Reddy, "Improving of clustering results for speech data by visual approach," in *Proc. Int. Conf. Signal Processing, Communication, Power and Embedded System*, Oct. 2016, pp. 691–696. doi: 10.1109/SCOPES.2016.7955527.

[100] K. R. Prasad and M. S. Basha, "Improving the performance of speech clustering method," in *Proc. Int. Conf. Intelligent Systems and Control*, Jan 2016, pp. 1–5. doi: 10.1109/ISCO.2016.7726878.

[101] B. Eswara Reddy and K. Rajendra Prasad, "Improving the performance of visualized clustering method," *Int. J. Syst. Assur. Eng. Manag.*, vol. 7, no. 1, pp. 102–111, Dec. 2016. doi: 10.1007/s13198-015-0342-x.

[102] T. Suneetha Rani and M. H. M. Krishna Prasad, "Access the cluster tendency by visual methods for robust speech clustering," *Int. J. Syst. Assur. Eng. Manag.*, vol. 8, no. 1, pp. 465–477, Jan. 2017. doi: 10.1007/s13198-015-0393-z.

[103] C. H. Chen, "A non-parametric feature assessment mechanism by identifying representative neighbors for image clustering," *Knowl.-Based Syst.*, vol. 67, pp. 328–341, Sept. 2014. doi: 10.1016/j.knosys.2014.04.026.

[104] B. Liu, H. Hu, K. Wang, X. Liu, and W. Yu, "A number-of-classes-adaptive unsupervised classification framework for SAR images," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp.*, July 2011, pp. 3799–3802. doi: 10.1109/IGARSS.2011.6050058.

[105] B. Liu, H. Hu, H. Wang, K. Wang, X. Liu, and W. Yu, "Superpixel-based classification with an adaptive number of classes for polarimetric SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 907–924, Feb. 2013. doi: 10.1109/TGRS.2012.2203358.

[106] H. Zou, N. Shao, M. Li, C. Chen, and X. Qin, "Superpixel-based unsupervised classification of polsar images with adaptive number of terrain classes," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp.*, July 2018, pp. 2390–2393. doi: 10.1109/IGARSS.2018.8519365.

[107] L. Wang, C. Leckie, X. Wang, R. Kotagiri, and J. C. Bezdek, "Tensor space learning for analyzing activity patterns from video sequences," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Oct. 2007, pp. 63–68. doi: 10.1109/ICDMW.2007.70.

[108] D. Zhang, K. Ramamohanarao, S. Versteeg, and R. Zhang, "RoleVAT: Visual assessment of practical need for role based access control," in *Proc. Annu. Computer Security Applications Conf.*, Dec. 2009, pp. 13–22. doi: 10.1109/ACSAC.2009.11.

[109] M. Du, S. Versteeg, J. G. Schneider, J. Han, and J. Grundy, "Interaction traces mining for efficient system responses generation," *SIGSOFT Softw. Eng. Notes*, vol. 40, no. 1, pp. 1–8, Feb. 2015. doi: 10.1145/2693208.2693221.

[110] S. Versteeg, M. Du, J. Schneider, J. Grundy, J. Han, and M. Goyal, "Opaque service virtualisation: A practical tool for emulating endpoint systems," in *Proc. IEEE/ACM Int. Conf. Software Engineering Companion*, May 2016, pp. 202–211. doi: 10.1145/2889160.2889242.

[111] A. Lourenco and A. Fred, "Unveiling intrinsic similarity: Application to temporal analysis of ECG," in *Proc. Int. Conf. Bio-Inspired Systems and Signal Processing*. Setúbal, Portugal: SciTePress, 2008, pp. 104–109.

[112] W. Stallaert, J. F. Dorn, E. van der Westhuizen, M. Audet, and M. Bouvier, "Impedance responses reveal β_2-adrenergic receptor signaling pluridimensionality and allow classification of ligands with distinct signaling profiles," *PLoS One*, vol. 7, no. 1, p. e29420, 2012. doi: 10.1371/journal.pone.0029420.

[113] O. M. Rivera-Borroto, M. Rabassa-Gutirrez, R. d C. Grau-balo, Y. Marrero-Ponce, and J. M. Garca-de la Vega, "Dunn's index for cluster tendency assessment of pharmacological data sets," *Can. J. Physiol. Pharmacol.*, vol. 90, no. 4, pp. 425–433, 2012. doi: 10.1139/y2012-002.

[114] I. Saha, U. Maulik, S. Bandyopadhyay, and D. Plewczynski, "Fuzzy clustering of physicochemical and biochemical properties of amino acids," *Amino Acids*, vol. 43, no. 2, pp. 583–594, Aug. 2012. doi: 10.1007/s00726-011-1106-9.

[115] B. A. Kingwell, C. Leckie, G. Wong, J. Chan, and P. J. Meikle, "LICRE: Unsupervised feature correlation reduction for lipidomics," *Bioinformatics*, vol. 30, no. 19, pp. 2832–2833, 2014. doi: 10.1093/bioinformatics/btu381.

[116] A. Brazma and J. Vilo, "Gene expression data analysis," *FEBS Lett.*, vol. 480, no. 1, pp. 17–24, 2000. doi: 10.1016/S0014-5793(00)01772-5.

[117] J. M. Keller, J. C. Bezdek, M. Popescu, N. R. Pal, J. A. Mitchell, and J. M. Huband, "Gene ontology similarity measures based on linear order statistics," *Int. J. Uncertain. Fuzz. Knowl.-Based Syst.*, vol. 14, no. 6, pp. 639–661, 2006. doi: 10.1142/S0218488506004254.

[118] M. Kim, H. Jung, M. Chung, P. Kim, S. Park, and S. Park, "Semi-automated clustering of gene expression data sets," in *Proc. Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society*, Aug. 2007, pp. 4625–4628.

[119] T. C. Havens, J. M. Keller, M. Popescu, J. C. Bezdek, E. M. Rehrig, H. M. Appel, and J. C. Schultz, "Fuzzy cluster analysis of bioinformatics data composed of microarray expression data and gene ontology annotations," in *Proc. Annu. Meeting of the North American Fuzzy Information Processing Society*, May 2008, pp. 1–6. doi: 10.1109/NAFIPS.2008.4531322.

[120] R. J. Hathaway and J. C. Bezdek, "Nerf *c*-means: Non-Euclidean relational fuzzy clustering," *Pattern Recog.*, vol. 27, no. 3, pp. 429–437, 1994. doi: 10.1016/0031-3203(94)90119-8.

[121] M. Popescu, J. C. Bezdek, and J. M. Keller, "eCCV: A new fuzzy cluster validity measure for large relational bioinformatics datasets," in *Proc. IEEE Int. Conf. Fuzzy Systems*, Aug. 2009, pp. 1003–1008. doi: 10.1109/FUZZY.2009.5277214.

[122] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "An interactive approach to multiobjective clustering of gene expression patterns," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 1, pp. 35–41, Jan. 2013. doi: 10.1109/TBME.2012.2220765.

[123] I. J. Sledge, J. M. Keller, and G. L. Alexander, "Emergent trend detection in diurnal activity," in *Proc. Int. Conf. IEEE Engineering in Medicine and Biology Society*, Aug. 2008, pp. 3815–3818. doi: 10.1109/IEMBS.2008.4650040.

[124] T. Banerjee, J. M. Keller, and M. Skubic, "Resident identification using kinect depth image data and fuzzy clustering techniques," in *Proc. Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society*, Aug. 2012, pp. 5102–5105. doi: 10.1109/EMBC.2012.6347141.

[125] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, May 1993. doi: 10.1109/91.227387.

[126] Y. Li, M. Popescu, and K. C. Ho, "Improving automatic sound-based fall detection using iVAT clustering and GA-based feature selection," in *Proc. Annu. Int. Conf. IEEE Engineering in Medicine and Biology Society*, Aug. 2012, pp. 5867–5870. doi: 10.1109/EMBC.2012.6347328.

[127] N. Liu and M. Lu, "A morphology method for determining the number of clusters present in spectral co-clustering documents and words," in *Computational Geometry, Graphs and Applications* (Lecture Notes in Computer Science, vol. 7033). Berlin: Springer-Verlag, 2011, pp. 130–141.

[128] R. Winkels, J. Douw, and S. Veldhoen, "Experiments in automated support for argument reconstruction," in *Proc. Int. Conf. Artificial Intelligence and Law*. New York: ACM, 2013, pp. 232–236. doi: 10.1145/2514601.2514633.

[129] R. Winkels, J. Douw, and S. Veldhoen, "State of the art: An argument reconstruction tool," in *Proc. Int. Workshop on Semantic Processing of Legal Texts*, 2014, pp. 17–23. doi: 10.13140/2.1.1012.4162.

[130] M. Ros et al., "Linguistic summarization of long-term trends for understanding change in human behavior," in *Proc. IEEE Int. Conf. Fuzzy Systems*, June 2011, pp. 2080–2087. doi: 10.1109/FUZZY.2011.6007509.

[131] A. Wilbik, J. M. Keller, and G. L. Alexander, "Linguistic summarization of sensor data for eldercare," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Oct. 2011, pp. 2595–2599. doi: 10.1109/ICSMC.2011.6084067.

[132] A. Wilbik, J. M. Keller, and J. C. Bezdek, "Generation of prototypes from sets of linguistic summaries," in *Proc. IEEE Int. Conf. Fuzzy Systems*, June 2012, pp. 1–8. doi: 10.1109/FUZZ-IEEE.2012.6250831.

[133] A. Wilbik and J. M. Keller, "Anomaly detection from linguistic summaries," in *Proc. IEEE Int. Conf. Fuzzy Systems*, July 2013, pp. 1–7.

[134] A. Wilbik, J. M. Keller, and J. C. Bezdek, "Linguistic prototypes for data from eldercare residents," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 1, pp. 110–123, Feb. 2014. doi: 10.1109/TFUZZ.2013.2249517.

[135] A. Chakraborty and S. Bandyopadhyay, "Clustering of web sessions by FOGSAA," in *Proc. IEEE Recent Advances in Intelligent Computational Systems*, Dec. 2013, pp. 282–287. doi: 10.1109/RAICS.2013.6745488.

[136] D. S. Sisodia, S. Verma, and O. P. Vyas, "A discounted fuzzy relational clustering of web users' using intuitive augmented sessions dissimilarity metric," *IEEE Access*, vol. 4, pp. 6883–6893, Oct. 16, 2016. doi: 10.1109/ACCESS.2016.2611682.

[137] L. Sun, S. Boztas, K. Horadam, A. Rao, and S. Versteeg, "Analysis of user behaviour in accessing a source code repository," RMIT University, Melbourne, Australia, and CA Technologies, New York City, Tech. Rep., 2013. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.472.8654&rep=rep1&type=pdf#page=22

[138] A. Kaskina, "Exploring nuances of user privacy preferences on a platform for political participation," in *Proc. Int. Conf. eDemocracy & eGovernment*, Apr. 2018, pp. 89–94. doi: 10.1109/ICEDEG.2018.8372317.

[139] M. R. Islam and M. U. Chowdhury, "Detecting unwanted email using VAT," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (Studies in Computational Intelligence, vol. 368). R. Lee, Ed. Berlin: Springer-Verlag, 2011, pp. 113–126.

[140] L. A. Cornelissen, R. J. Barnett, P. Schoonwinkel, B. D. Eichstadt, and H. B. Magodla, "A network topology approach to bot classification," in *Proc. Annu. Conf. South African Institute of Computer Scientists and Information Technologists*. New York: ACM, 2018, pp. 79–88. doi: 10.1145/3278681.3278692.

[141] S. Yang, S. Luo, and J. Li, "A novel visual clustering algorithm for finding community in complex network," in *Advanced Data Mining and Application* (Lecture Notes in Computer Science, vol. 4093), X. Li, O. R. Zaïane, and Z. Li, Eds. Berlin: Springer-Verlag, 2006, pp. 396–403.

[142] T. A. Runkler and J. C. Bezdek, "Fuzzy relational approaches to graph clustering and visualization," in *Proc. GMA/GI Workshop Computational Intelligence, Dortmund,* 2013, pp. 39–56.

[143] T. C. Havens et al., "Clustering and visualization of fuzzy communities in social networks," in *Proc. IEEE Int. Conf. Fuzzy Systems*, July 2013, pp. 1–7. doi: 10.1109/FUZZ-IEEE.2013.6622299.

[144] J. Su and T. C. Havens, "Fuzzy community detection in social networks using a genetic algortihm," in *Proc. IEEE Int. Conf. Fuzzy Systems*, July 2014, pp. 2039–2046. doi: 10.1109/FUZZ-IEEE.2014.6891611.

[145] J. Su and T. C. Havens, "Quadratic program-based modularity maximization for fuzzy community detection in social networks," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1356–1371, Oct. 2015. doi: 10.1109/TFUZZ.2014.2360723.

[146] M. Ganji, A. Seifi, H. Alizadeh, J. Bailey, and P. J. Stuckey, "Generalized modularity for community detection," in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science, vol. 9285), A. Appice, P. Rodrigues, V. Santos Costa, J. Gama, A. Jorge, and C. Soares, Eds. Cham, Switzerland: Springer-Verlag, 2015, pp. 655–670.

[147] D. Kumar, H. Wu, Y. Lu, S. Krishnaswamy, and M. Palaniswami, "Understanding urban mobility via taxi trip clustering," in *Proc. IEEE Int. Conf. Mobile Data Management*, June 2016, pp. 318–324. doi: 10.1109/MDM.2016.54.

[148] L. Li and C. Leckie, "Trajectory pattern identification and anomaly detection of pedestrian flows based on visual clustering," in *Intelligent Information Processing VIII* (IFIP Advances in Information and Communication Technology, vol. 486), Z. Shi, S. Vadera, and G. Li, Eds. Cham, Switzerland: Springer-Verlag, 2016, pp. 121–131.

[149] D. Kumar, J. C. Bezdek, S. Rajasegarar, C. Leckie, and M. Palaniswami, "A visual-numeric approach to clustering and anomaly detection for trajectory data," *Visual Comput.*, vol. 33, no. 3, pp. 265–281, Mar. 2017. doi: 10.1007/s00371-015-1192-x.

[150] D. Kumar, S. Rajasegarar, M. Palaniswami, X. Wang, and C. Leckie, "A scalable framework for clustering vehicle trajectories in a dense road network," in *Proc. ACM SIGKDD Int. Workshop Urban Computing*, 2015.

[151] P. Rathore, D. Kumar, S. Rajasegarar, M. S. Palaniswami, and J. C. Bezdek, "A scalable framework for trajectory prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, 2019. doi: 10.1109/TITS.2019.2899179.

[152] J. L. Viegas and S. M. Vieira, "Clustering-based novelty detection to uncover electricity theft," in *Proc. IEEE Int. Conf. Fuzzy Systems*, July 2017, pp. 1–6. doi: 10.1109/FUZZ-IEEE.2017.8015546.

[153] N. Marwan, M. Romano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Phys. Rep.*, vol. 438, nos. 5–6, pp. 237–329, 2007. doi: 10.1016/j.physrep.2006.11.001.

[154] M. Sips, C. Witt, T. Rawald, and N. Marwan, "Towards visual analytics for the exploration of large sets of time series," in *Recurrence Plots and Their Quantifications: Expanding Horizons* (Springer Proceedings in Physics, vol. 180), C. Webber Jr., C. Ioana, and N. Marwan, Eds. Cham, Switzerland: Springer-Verlag, 2016, pp. 3–17.

[155] T. B. Iredale, S. M. Erfani, and C. Leckie, "An efficient visual assessment of cluster tendency tool for large-scale time series data sets," in *Proc. IEEE Int. Conf. Fuzzy Systems*, July 2017, pp. 1–8. doi: 10.1109/FUZZ-IEEE.2017.8015587.

[156] S. Mahallati, J. C. Bezdek, D. Kumar, M. R. Popovic, and T. A. Valiante, "Interpreting cluster structure in waveform data with visual assessment and Dunn's index," in *Frontiers in Computational Intelligence* (Studies in Computational Intelligence, vol. 739), S. Mostaghim, A. Nürnberger, and C. Borgelt, Eds. Cham, Switzerland: Springer-Verlag, 2018, pp. 73–101.

[157] J. C. Bezdek et al., "Clustering elliptical anomalies in sensor networks," in *Proc. Int. Conf. Fuzzy Systems*, July 2010, pp. 1–8. doi: 10.1109/FUZZY.2010.5584464.

[158] J. C. Bezdek, S. Rajasegarar, M. Moshtaghi, C. Leckie, M. Palaniswami, and T. C. Havens, "Anomaly detection in environmental monitoring networks," *IEEE Comput. Intell. Mag.*, vol. 6, no. 2, pp. 52–58, May 2011. doi: 10.1109/MCI.2011.940751.

[159] M. Moshtaghi et al., "Clustering ellipses for anomaly detection," *Pattern Recog.*, vol. 44, no. 1, pp. 55–69, 2011. doi: 10.1016/j.patcog.2010.07.024.

[160] S. Rajasegarar, J. C. Bezdek, M. Moshtaghi, C. Leckie, T. C. Havens, and M. Palaniswami, "Measures for clustering and anomaly detection in sets of higher dimensional ellipsoids," in *Int. Joint Conf. Neural Networks*, June 2012, pp. 1–8. doi: 10.1109/IJCNN.2012.6252703.

[161] F. Heider, *The Psychology of Interpersonal Relations*. Hoboken, NJ: Wiley, 1958.

[162] A. Notsu, H. Ichihashi, and K. Honda, "Agent-based simulation about social value emergence based on perceptual balance," *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, vol. 1, pp. 724–728, Oct. 2006. doi: 10.1109/ICSMC.2006.384472.

[163] A. Notsu, H. Ichihashi, K. Honda, and O. Katai, "Visualization of balancing systems based on naïve psychological approaches," *AI Soc.*, vol. 23, no. 2, pp. 281–296, Mar. 2009. doi: 10.1007/s00146-007-0142-1.

[164] A. Buck, A. Zare, J. Keller, and M. Popescu, "Endmember representation of human geography layers," in *Proc. IEEE Symp. Computational Intelligence in Big Data*, Dec. 2014, pp. 1–6. doi: 10.1109/CIBD.2014.7011520.

[165] K. Nikolaidis, E. Rodriguez, J. Y. Goulermas, and Q. H. Wu, "Instance seriation for prototype abstraction," in *Proc. IEEE Int. Conf. Bio-Inspired Computing: Theories and Applications*, Sept. 2010, pp. 1351–1355. doi: 10.1109/BICTA.2010.5645066.

[166] K. Deb, "Multi-objective evolutionary algorithms," in *Springer Handbook of Computational Intelligence,* J. Kacprzyk and W. Pedrycz, Eds. Berlin: Springer-Verlag, 2015, pp. 995–1015.

[167] R. Kudikala, I. Giagkiozis, and P. Fleming, "Increasing the density of multi-objective multi-modal solutions using clustering and pareto estimation techniques," in *Proc. World Congr. Computer Science Computer Engineering and Applied Computing*, 2013. [Online]. Available: http://worldcomp-proceedings.com/proc/p2013/GEM3028.pdf

[168] H. Nguyen, C. Drebenstedt, X.-N. Bui, and D. T. Bui, "Prediction of blast-induced ground vibration in an open-pit mine by a novel hybrid model based on clustering and artificial neural network," *Natural Resources Research*, Mar. 2019. doi: 10.1007/s11053-019-09470-z.

[169] J. Gllavata, "Extracting textual information from images and videos for automatic content-based annotation and retrieval," Ph.D. dissertation, Dept. Math. Comput. Sci., Philipps-Univ., Marburg, 2007. [Online]. Available: https://d-nb.info/984094075/34

[170] F. Rehm, "Visual data analysis in air traffic management," Ph.D. dissertation, Otto von Guericke Univ., Magdeburg, Germany, 2007. [Online]. Available: https://core.ac.uk/download/pdf/51447907.pdf

[171] E. Qeli, "Information visualization techniques for metabolic engineering," Ph.D. dissertation, Dept. Math. Comput. Sci., Philipps Univ., Marburg, 2007. [Online]. Available: https://archiv.ub.uni-marburg.de/ubfind/Record/urn:nbn:de:hebis:04-z2007-0108/Description#tabnav

[172] R. Luke, "A system for change detection and human recognition in voxel space using stereo vision," Ph.D. dissertation, Dept. Comput. Eng., Univ. Missouri, 2010. [Online]. Available: https://mospace.umsystem.edu/xmlui/handle/10355/8877

[173] T. C. Havens, "Clustering in relational data and ontologies," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. of Missouri, 2010.

[174] R. M. M. Felizardo, "A study on parallel versus sequential relational fuzzy clustering methods," Master's thesis, Dept. Inform. Technol., Universidade Nova de Lisboa, 2011. [Online]. Available: https://run.unl.pt/bitstream/10362/5663/1/Felizardo_2011.pdf

[175] C. Brunet, "Sparse and discriminative clustering for complex data," Ph.D. dissertation, Appl. Math. (Stat.), Univ. d'Evry-Val d'Essonne, 2012. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00671333/document

[176] S. Parveen, "Visualization of transformation of graphs based on similarity functions," Master's thesis, Dept. Inform. Technol., International Institute of Information Technology, Bangalore, 2013.

[177] V. Uslan, "Support vector machine-based fuzzy systems for quantitative prediction of peptide binding affinity," Ph.D. dissertation, De Montfort Univ., 2015. [Online]. Available: https://www.dora.dmu.ac.uk/handle/2086/11170

[178] X. Ye, "Experimental study of random projections below the JL limit," Master's thesis, Dept. Comput. Eng., Univ. Missouri, Columbia, 2015. [Online]. Available: https://mospace.umsystem.edu/xmlui/bitstream/handle/10355/47052/research.pdf?sequence=2\&isAllowed=y

[179] C. Witt, "Clustering von recurrence plots," Master's thesis, Dept. Comput. Sci., Humboldt-Universitt zu Berlin, 2015.

[180] D. Kumar, "Big data clustering for smart city applications," Ph.D. dissertation, Dept. Electr. & Electron. Eng., Univ. Melbourne, 2016. [Online]. Available: http://hdl.handle.net/11343/129505

[181] M. Du, "Enabling automatic creation of virtual services for service virtualisation," Ph.D. dissertation, Faculty Sci., Eng. Technol., Swinburne Univ. Technol., 2016. [Online]. Available: https://arxiv.org/pdf/1608.04885.pdf

[182] P. Rathore, "Big data cluster analysis and its applications," Ph.D. dissertation, 2018. [Online]. Available: http://hdl.handle.net/11343/219493

[183] R. A. Fisher, *Statistical Methods for Research Workers*. Edinburgh: Oliver and Boyd, 1932.

[184] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979. doi: 10.1109/TSMC.1979.4310076.

SMC