Is VAT really single linkage in disguise?

Article *in* Annals of Mathematics and Artificial Intelligence · April 2009

DOI: 10.1007/s10472-009-9157-2 · Source: DBLP

Ann Math Artif Intell
DOI 10.1007/s10472-009-9157-2

Is VAT really single linkage in disguise?

**Timothy C. Havens · James C. Bezdek ·
James M. Keller · Mihail Popescu · Jacalyn M. Huband**

Abstract This paper addresses the relationship between the Visual Assessment of cluster Tendency (VAT) algorithm and single linkage hierarchical clustering. We present an analytical comparison of the two algorithms in conjunction with numerical examples to show that VAT reordering of dissimilarity data is directly related to the clusters produced by single linkage hierarchical clustering. This analysis is important to understanding the underlying theory of VAT and, more generally, other algorithms that are based on VAT-ordered dissimilarity data.

Keywords Clustering · Visual assessment of cluster tendency · Single linkage · Relational data · Minimal spanning tree · Graph theory

Mathematics Subject Classifications (2000) 62H30 · 68T10 · 68R10

T. C. Havens (✉) · J. C. Bezdek · J. M. Keller · M. Popescu
University of Missouri, Columbia, MO 65211, USA
e-mail: havenst@gmail.com

J. C. Bezdek
e-mail: jcbezdek@gmail.com

J. M. Keller
e-mail: kellerj@missouri.edu

M. Popescu
e-mail: popescum@missouri.edu

J. M. Huband
University of West Florida, Pensacola, FL 32514, USA
e-mail: huband@uwf.edu

Published online: 05 August 2009

1 Introduction

Consider a set of n objects $\mathbf{O} = \{o_1, \ldots, o_n\}$, where these objects might be pixels in an image, expressed genes in a microarray experiment, bass guitars, fast food joints, swimming pools, etcetera. Numerical *object* data is represented as $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^p$, where each dimension of the vector $\mathbf{x}_i$ is a feature value of the object o_i (i.e., RGB values, expression, number of strings, food quality, temperature, chlorine-content, calories). Another way to represent the objects in $\mathbf{O}$ is with numerical *relational* data, which consists of n^2 values that represent the similarity between pairs of objects. These data are commonly represented by a relational matrix $\mathbf{R} = \left[r_{ij} = \text{relation}(o_i, o_j) | 1 \leq i, j \leq n\right]$. In this paper, we will represent relational data by the *dissimilarity* matrix $\mathbf{D}$, where, for example, any numerical data $\mathbf{X}$ can be converted to $\mathbf{D}$ by $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ (any vector norm on $\mathbb{R}^p$). We note however that there are dissimilarity data for which no numerical data is available.

A wide array of algorithms exists for clustering unlabeled object data $\mathbf{O}$ [3, 5, 9, 12, 16, 26]. Crisp partitions of the unlabeled data are non-intersecting subsets of $\mathbf{O}$ such that the union of the subsets cover $\mathbf{O}$. A crisp c-partition of $\mathbf{O}$ can be represented by a set of cn values $\{u_{ik}\}$ that can be conveniently arrayed as a $c \times n$ matrix $\mathbf{U} = [u_{ik}]$. The set of all non-degenerate c-partition matrices (no empty or overlapping clusters) for the object set $\mathbf{O}$ is:

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

where u_{ik} is the *membership* of object k in cluster i—the partition element $u_{ik} = 1$ if o_k is labeled i and is 0 otherwise. In this paper, we focus on a special subset of M_{hcn}, viz. aligned c-partitions of $\mathbf{O}$. Section 2 discusses aligned partitions in detail.

Hierarchical clustering has been extensively studied. There are two forms of hierarchical clustering: (a) *agglomerative*, in which clusters are formed by starting with n singleton clusters and merging similar entities until the data comprises one cluster; (b) *divisive*, in which the data are initially partitioned into one cluster and at each successive step a cluster is divided into two subsets until there are n singleton clusters. Hence, agglomerative clustering produces $n - 1$ clusters after the first step, $n - 2$ after the second step, and so on. The three most popular methods to determine which clusters should be joined (or divided) in hierarchical clustering: (a) single linkage; (b) complete linkage; and (c) average linkage. In this paper, we focus on *single linkage* (SL), which is discussed in detail in Section 3.1. For a thorough description of hierarchical clustering, see the reference [26].

Three pertinent questions when clustering data are: (a) tendency—how many clusters are there?; (b) partitioning—which objects belong to which cluster?; and (c) validity—are the partitions "good"? The Visual Assessment of cluster Tendency (VAT) algorithm attempts to answer the tendency question by utilizing a modified Prim's minimal spanning tree algorithm to create an image of a reordered dissimilarity matrix that, in the right circumstances, shows the clusters as dark blocks along the diagonal [1, 23]. Section 3.2 discusses VAT in detail.

There are multiple versions of VAT, each addressing a different cluster tendency problem—very large (unloadable) data sets [11, 15], rectangular data [2], among others [14]. There are also methods that read the VAT image to determine cluster tendency automatically [24, 28]. Researchers have also extended VAT to answer the cluster validity question [4, 10, 13]. *CLustering in Ordered Dissimilarity Data*-CLODD is an algorithm that answers the partitioning and validity questions by automatically creating partitions from the VAT image and providing a validity measure of each partition [12]. Reference [29] describes a clustering algorithm that uses VAT to visualize complex networks. Other researchers have adapted VAT to produce visualizations of an interpersonal psychological approach called "naive psychology" [20]. Our analysis is important to understanding the results of all VAT-related research.

The authors of VAT have long held that SL and VAT are directly related. Both algorithms are expressed in terms of the minimal spanning tree of the connected graph that represents the object data and its dissimilarity values. This observation leads us to explore the nature of a VAT-SL relationship. Section 4 presents an analytic comparison of the two algorithms, while Section 5 illustrates this relationship with numerical examples. We wrap up this paper in Section 6 with a short discussion, conclusions, and ideas for future research.

2 Aligned c-partitions

A subset of the c-partitions $\mathbf{U}$ in $\mathbf{M}_{hcn}$ is the set of *aligned* c-partitions of $\mathbf{O}$ [12]. These are defined as the partitions of $\mathbf{O}$ that form c contiguous blocks of 1's in $\mathbf{U}$, beginning in the upper left corner, and proceeding down and to the right:

$$\begin{aligned} \mathbf{M}^*_{hcn} = \{\mathbf{U} \in \mathbf{M}_{hcn} | u_{1k} = 1, 1 \le k \le n_1; \\ u_{ik} = 1, n_{i-1} + 1 \le k \le n_i, 2 \le i \le c\}. \end{aligned} \tag{2}$$

For example, $\begin{bmatrix} 1\,1\,1\,0\,0 \\ 0\,0\,0\,1\,1 \end{bmatrix}$ and $\begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,1\,1\,0\,0 \\ 0\,0\,0\,0\,1\,1 \end{bmatrix}$ are aligned partitions, while $\begin{bmatrix} 0\,0\,0\,1\,0 \\ 1\,1\,1\,0\,1 \end{bmatrix}$ and $\begin{bmatrix} 0\,0\,0\,0\,1\,1 \\ 1\,0\,0\,0\,0\,0 \\ 0\,1\,1\,1\,0\,0 \end{bmatrix}$ are not. Essentially, the first n_1 objects are partitioned in the first cluster, the next n_2 objects are partitioned in the second cluster, etcetera.

The transformation $\mathbf{U}^T\mathbf{U}$ produces an $n \times n$ matrix with elements that measure the relationship between pairs of objects in each of c clusters [13]. The ij-th element of $\mathbf{U}^T\mathbf{U}$ is $\left(\mathbf{U}^T\mathbf{U}\right)_{ij} = \sum_{k=1}^{c} u_{ki}u_{kj}$; thus, $\left(\mathbf{U}^T\mathbf{U}\right)_{ij}$ is a measure of the binding between objects i and j over all c clusters. In this paper, we will only address crisp partitions $\mathbf{U} \in \mathbf{M}_{hcn}$, where all elements are 1 or 0; hence, $\left(\mathbf{U}^T\mathbf{U}\right)_{ij}$ will be zero unless objects i and j are in the same cluster, in which case the ij-th entry will have the value 1.

A property of aligned c-partitions is that the image of the transformation $\mathbf{U}^T\mathbf{U}$ will have dark blocks along its diagonal: black represents 1 in the image, white represents

0. In this paper, we denote this image of the transformation as $T(\mathbf{U}) = \mathbf{U}^T\mathbf{U}$. The following example shows the transformation for $c = 2$ and $n = 5$; $\mathbf{U} = \begin{bmatrix} 1\,1\,0\,0\,0 \\ 0\,0\,1\,1\,1 \end{bmatrix}$:

$$\mathbf{U}^T\mathbf{U} = \begin{bmatrix} 1\,0 \\ 1\,0 \\ 0\,1 \\ 0\,1 \\ 0\,1 \end{bmatrix} \times \begin{bmatrix} 1\,1\,0\,0\,0 \\ 0\,0\,1\,1\,1 \end{bmatrix} = \begin{bmatrix} 1\,1\,0\,0\,0 \\ 1\,1\,0\,0\,0 \\ 0\,0\,1\,1\,1 \\ 0\,0\,1\,1\,1 \\ 0\,0\,1\,1\,1 \end{bmatrix} \tag{3}$$

As this example shows, the blocks of 1's correspond to the grouping of the objects according to the partition. We can also write the aligned partitions of $\mathbf{O}$ in set-notation. For example, the aligned partition in (3) partitions $\mathbf{O}$ as $\{o_1, o_2\}$ and $\{o_3, o_4, o_5\}$; notice that the partitioned subsets of aligned c-partitions will always be ordered.

3 Algorithms

3.1 Single linkage hierarchical clustering

SL hierarchical clustering can be used on both numerical object data $\mathbf{X}$ and (square) relational data $\mathbf{D}$. We emphasize that $\mathbf{D}$ is square here to distinguish this problem from the case of *rectangular* relational data. SL is undefined for this more general case. In order to simplify the discussion of SL in this paper, we assume that numerical data has been converted to relational data by a vector norm distance measure. In terms of the dissimilarity matrix $\mathbf{D}$, the SL (set) distance between two clusters $\mathbf{C}_1$ and $\mathbf{C}_2$ is defined as

$$d_{SL}(\mathbf{C}_1, \mathbf{C}_2) = \min_{o_p \in \mathbf{C}_1, o_q \in \mathbf{C}_2} d_{pq}. \tag{4}$$

Figure 1 illustrates the SL distance between two clusters composed of two-dimensional numerical object data. This distance measure is used at each step of the SL agglomerative clustering algorithm to determine which two clusters are joined. Algorithm 1 outlines the steps of SL agglomerative clustering.

SL clustering can also be expressed in terms of the *minimal spanning tree* (MST) of the connected graph that represents the object data. The weight of the edge between two objects is the dissimilarity value of those two objects. The MST is defined as

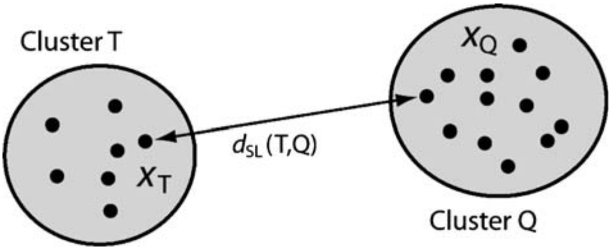


Fig. 1 Single linkage distance between two clusters

Algorithm 1 Single Linkage Agglomerative Clustering [5]

```
Data: $\mathbf{C}^{(n)} = \{\{o_1\}, \{o_2\}, \ldots, \{o_n\}\}$;
D - dissimilarity matrix, $d_{pq} = \text{dissimilarity}(o_p, o_q),\ 1 \le i, j \le n$
for $c = n, \ldots, 2$ do
    $$s_{ij} = d_{SL}\left(\mathbf{C}_i^{(c)}, \mathbf{C}_j^{(c)}\right)\ \forall i, j$$
    Find closest two clusters, $\{l, m\} = \arg\min_{i,j} s_{ij}$.
    $$\mathbf{C}^{(c-1)} \leftarrow \mathbf{C}^{(c)} - \mathbf{C}_l^{(c)} - \mathbf{C}_m^{(c)} + \mathbf{C}_l^{(c)} \cup \mathbf{C}_m^{(c)}$$
end
```

the $n-1$ edges that fully connect the object data $\mathbf{O}$ and have the minimum summed edge weight [8]. A number of algorithms exist that compute the MST of a connected graph, Prim's and Kruskal's algorithms being two of the most popular [17, 23]. The SL distance and the MST are related in that the weight of the MST edge between two subtrees of object data is the SL distance between them. Algorithm 2 outlines Prim's algorithm in terms of the SL distance, where (5) returns the indices of the two closest objects from sets **I** and **J** according to the SL distance.

The SL clusters of the object data **O** can be found by cutting the high weight edges of the MST and examining the resulting subtrees [7]. For example, the 5-partition of **O** can be found by cutting the four highest weight edges in the MST of **O**, which results in five subtrees—each containing the objects that represent a cluster. We use this important result to prove that SL clusters are aligned partitions of the VAT reordered objects $\mathbf{O}^*$.

Algorithm 2 Prim's Algorithm [23]

```
Data: D - dissimilarity matrix; $\mathbf{I} = \emptyset$; $\mathbf{J} = \{o_1, \ldots, o_n\}$
Pick a starting object $o_m$
$\mathbf{I} \leftarrow \{o_m\}, \mathbf{J} \leftarrow \mathbf{J} - \{o_m\}$
for $r = 2, \ldots, n$ do
    Select
    $$(i, j) \in \underset{o_p \in \mathbf{I}, o_q \in \mathbf{J}}{\arg\min}\ d_{pq}. \qquad (5)$$
    Create an edge between $o_i$ and $o_j$.
    $\mathbf{I} \leftarrow \mathbf{I} \cup \{o_j\}$ and $\mathbf{J} \leftarrow \mathbf{J} - \{o_j\}$.
end
```

3.2 VAT

The VAT algorithm is based on *Prim's algorithm* (PA) for finding the MST of a weighted connected graph [1]. Algorithm 3 illustrates the steps of the VAT algorithm; notice the almost line-for-line similarity to PA (Algorithm 2). VAT reorders the dissimilarity matrix in the same order in which PA adds vertices to the MST. Already, it can be seen that there is a direct relation between VAT and the MST (and, subsequently, single linkage). The only differences between VAT and PA is the choice of the starting object (although PA could be initialized with the VAT starting object) and the end result; VAT produces an image, while PA produces a spanning tree.

Algorithm 3 VAT Ordering Algorithm [1]

```
Data: $\mathbf{D}$ - dissimilarity matrix; $\mathbf{K} = \{1, 2, \ldots, n\}$; $\mathbf{I} = \mathbf{J} = \emptyset$; $P = (0, 0, \ldots, 0)$.
Select
    $$(i, j) \in \arg\max_{p \in \mathbf{K}, q \in \mathbf{K}} d_{pq}. \qquad (6)$$
Set $P(1) = i$; $\mathbf{I} = \{i\}$; and $\mathbf{J} = \mathbf{K} - \{i\}$.
for $r = 2, \ldots, n$ do
    Select $(i, j) \in \arg\min_{p \in \mathbf{I}, q \in \mathbf{J}} d_{pq}$.
    Set $P(r) = j$; Replace $\mathbf{I} \leftarrow \mathbf{I} \cup \{j\}$ and $\mathbf{J} \leftarrow \mathbf{J} - \{j\}$.
end
Obtain the ordered dissimilarity matrix $\mathbf{D}^*$ using the ordering array $P$ as:
$d^*_{pq} = d_{P(p),P(q)}$, for $1 \le p, q \le n$.
```

The resulting VAT-reordered dissimilarity matrix $\mathbf{D}^*$ can be normalized and mapped to a gray-scale image with black representing the minimum dissimilarity and white the maximum. Figure 2 is an example of the VAT image for a set of five clusters. The five dark blocks along the diagonal of the VAT image suggest that the object data seen in Fig. 2a possesses five clusters, but the clusters are not identified. Havens et al. [12] developed an algorithm called CLODD-*CLustering in Ordered Dissimilarity Data* that extracts an aligned partition of the objects from a VAT image such as that seen in Fig. 2c.

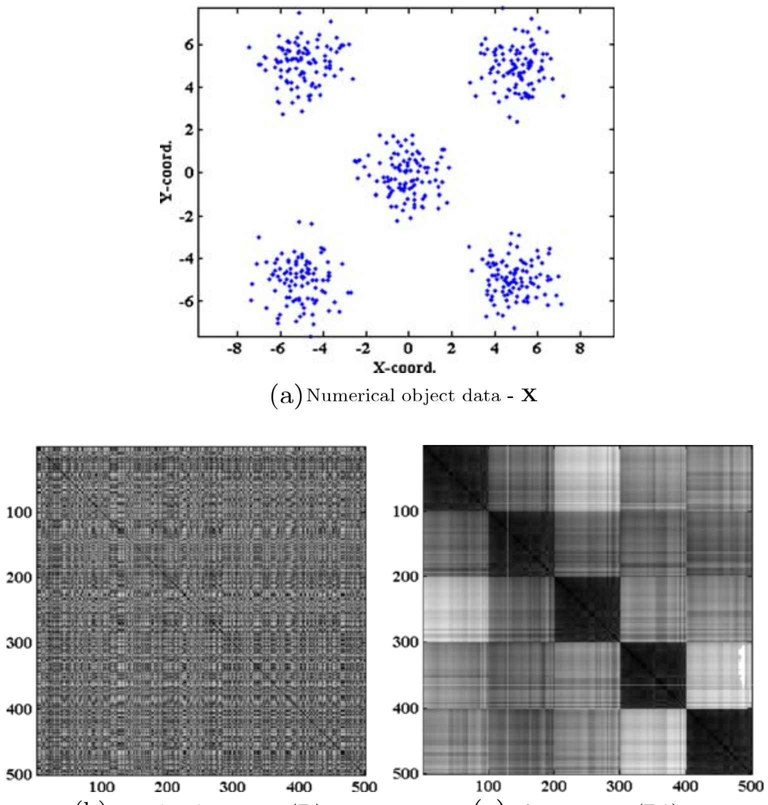
4 Analytical comparison

Consider a dissimilarity matrix $\mathbf{D}$, which is then reordered by VAT with Prim's MST algorithm. The VAT ordering of the MST is a special subset of all possible orderings computed by PA, where VAT imposes the following property by its initialization (see (6) in Algorithm 3),

$$o_1^* = o_i, \ \{i, j\} = \arg\max_{i,j} d_{ij}.$$

Essentially, the first object added to the MST is the object corresponding to the row of the maximum element of $\mathbf{D}$ (in practice, one could choose either object that corresponds to maximum element of $\mathbf{D}$ as the initialization).

We generalize our discussion in this section by including all possible orderings of the MST that result from Prim's algorithm. $\mathbf{O}^* = \{o_1^*, \ldots, o_n^*\}$ represents the set of reordered objects as a result of PA (where ANY object can be the starting object in PA). Similarly, we let $\mathbf{D}^*$ correspond to the PA ordering $\mathbf{O}^*$. Since the VAT ordering



(a) Numerical object data - $\mathbf{X}$

(b) Unordered image - $I(\mathbf{D})$

(c) VAT image - $I(\mathbf{D}^*)$

Fig. 2 VAT example (**a**–**c**)

is a subset of the orderings imposed by PA, the discussion in this section also applies to VAT.

Without loss of generality we let the objects in any reordered object data $\mathbf{O}^*$ also represent the vertices of a fully connected graph, where the individual vertices are denoted as o_i^*, $i = 1, \ldots, n$. The edge weights of this graph are computed by your chosen vector norm or dissimilarity function d,

$$\text{edge weight}_{ij} = d(o_i^*, o_j^*). \tag{7}$$

Subsequently, the i-th edge of the MST, denoted as e_i, has a weight that can be computed by

$$w_i = d_{SL}\left(\left\{o_1^*, \ldots, o_i^*\right\}, \left\{o_{i+1}^*\right\}\right), \; i = 1, \ldots, n-1. \tag{8}$$

This is a direct result of PA, which adds edges to the MST by finding the closest safe vertex to the set of vertices previously added to the tree. Notice that in the context of graph theory, vertices (objects) are separated into c subtrees, where in the context of clustering this corresponds to a c-partition of the (corresponding) objects.

4.1 Aligned c-partitions in VAT and SL

Now, consider the relation between SL clustering and the MST. As stated in Section 3.1, SL clustering can be performed by cutting the MST. SL divisive clustering cuts the MST at the highest weight edge(s) and the resulting subtrees are the SL clusters (for a given choice of c [the number of clusters]). Furthermore, if the $\binom{n}{2}$ off-diagonal dissimilarity values are distinct, and the similarity measures (or distance) are consistent, the MST is unique and will be the same regardless of the similarity measure (or MST-finding algorithm used) [18]. We define consistent similarity measures as those which produce the same MST ordering for a given set of objects.

We need two lemmas in order to prove that the SL clusters, at every c, derived by applying PA to any $\mathbf{D}$ with $n(n-1)/2$ distinct off-diagonal dissimilarities are aligned c-partitions of $\mathbf{O}^*$.

Lemma 1 *The edge weight w_i of edge e_i satisfies*

$$w_i < d_{SL}\left(\left\{o_1^*, \ldots, o_i^*\right\}, \left\{o_{i+2}^*, \ldots, o_n^*\right\}\right),$$

for reordered object data with unique dissimilarity values.

Proof First, the dissimilarity values are unique; thus,

$$d_{SL}\left(\left\{o_1^*, \ldots, o_i^*\right\}, \left\{o_{i+2}^*, \ldots, o_n^*\right\}\right) \neq w_i.$$

Second, Prim's algorithm states that the $(i+1)$-th vertex added to the MST is the vertex closest to the subtree $\{o_1^*, \ldots, o_i^*\}$, namely o_{i+1}^*. Hence, all edges that connect the subtrees $\{o_1^*, \ldots, o_i^*\}$ and $\{o_{i+2}^*, \ldots, o_n^*\}$ must be greater in weight than edge e_i, resulting in the identity

$$d_{SL}\left(\left\{o_1^*, \ldots, o_i^*\right\}, \left\{o_{i+2}^*, \ldots, o_n^*\right\}\right) > w_i,$$

which proves the lemma. □

Next we use Lemma 1 to prove

Lemma 2 *If $w_m = \max_j w_j$, then cutting the associated MST edge e_m produces the two ordered subtrees, $\{o_1^*, \ldots, o_m^*\}$ and $\{o_{m+1}^*, \ldots, o_n^*\}$.*

Proof Lemma 1 shows that

$$w_m < d_{SL}\left(\left\{o_1^*, \ldots, o_m^*\right\}, \left\{o_{m+2}^*, \ldots, o_n^*\right\}\right). \tag{9}$$

Also, since e_m is the highest weight edge it follows that

$$w_m > w_i, \ \forall i \neq m. \tag{10}$$

Hence, if there exists an MST edge, other than e_m, that connects the two subtrees $\{o_1^*, \ldots, o_m^*\}$ and $\{o_{m+1}^*, \ldots, o_n^*\}$, this edge must satisfy both (9) and (10). These two conditions cannot be met simultaneously. Thus, there is no edge in the MST that connects the two subtrees, $\{o_1^*, \ldots, o_m^*\}$ and $\{o_{m+1}^*, \ldots, o_n^*\}$, if e_m is cut. □

Please note that the subtrees that result from cutting the MST at its maximum weight edge are analogous to an aligned 2-partition of the ordered object data $\mathbf{O}^*$. Now we are ready to state and prove the main result by applying Lemma 2 to divisive clustering.

Proposition 1 *For a set of object data for which unique dissimilarity values can be computed (or chosen) from a consistent dissimilarity measure, the SL clusters will be aligned c-partitions of the PA reordered object data* $\mathbf{O}^*$ *for every value of c.*

Proof Sort the edges of the MST according to their weight in decreasing order, where $e_{(1)} > e_{(2)} > \ldots > e_{(n-1)}$ are the sorted edges. The $(c = 2)$ SL clusters can be computed by cutting the MST at the highest weighted edge $e_m = e_{(1)}$. Lemma 2 proves that the resulting subtrees correspond to the aligned 2-partition, $\{o_1^*, \ldots, o_m^*\}$ and $\{o_{m+1}^*, \ldots, o_n^*\}$. Furthermore, $\{e_1, \ldots, e_{m-1}\}$ are the MST edges (in PA order) of the subtree $\{o_1^*, \ldots, o_m^*\}$, and $\{e_{m+1}, \ldots, e_{n-1}\}$ are the MST edges (in PA order) of the subtree $\{o_{m+1}^*, \ldots, o_n^*\}$. The $(c = 3)$ SL clusters are found by cutting the subtrees at edge $e_{(2)}$. Note that this results in only one of the subtrees being cut, which occurs at its maximum weight edge. Hence, recursive application of Lemma 2 for each of the $n - 1$ steps from $c = n$ to $c = 1$ shows that the resulting subtrees of this cut will also represent aligned partitions. Thus, all SL c-partitions of the PA ordered object data are aligned. □

Remark 1 The VAT image of $\mathbf{D}$ may suggest that the data it represents contains c clusters, but as previously noted, it is CLODD [12] that extracts an aligned c-partition of $\mathbf{O}^*$ from the image. Proposition 1 shows that when the MST of $\mathbf{D}$ is unique, CLODD extracts SL clusters from the VAT image. The important point is that CLODD extracts only the aligned SL partition at the (VAT suggested) "best value" for c, whereas SL produces $n - 1$ aligned c-partitions of $\mathbf{O}$. Thus, [VAT + CLODD] is, for some (but not all) $\mathbf{D}$'s, equivalent to [SL + some heuristic means for choosing the "best" SL partition].

Remark 2 Proposition 1 only applies to relational data that have a unique MST (which, arguably, includes most real-world data sets). It is beyond the scope of this paper to extend our analysis to data that have more than one MST. However, if needed, a small perturbation of *any* data set will transform it into one that does have a unique MST by adding a small random noise to each element in the data. Furthermore, these small perturbations will not affect the cluster structure in the data as they are negligible, in practice. Hence, the analysis in this paper does apply to most, if not all, real-world data sets.

5 Numerical examples

This section contains three examples that illustrate the relationship of SL and VAT explained in the previous section. We compute dissimilarity data from numerical data by the Euclidean norm for all examples, except the Bioinformatics example (which is a pure relational data set). In order to quantize the *compact, separated* (CS) property of the visually apparent partition in the first two examples, we use *Dunn's validity index* (DI) [6]. DI is defined as

$$\alpha(c, \mathbf{U}) = \frac{\min_{1 \le p \le c} \min_{1 \le q \le c, q \ne p} d_{SL}(\mathbf{C}_p, \mathbf{C}_q)}{\max_{1 \le r \le c} \operatorname{diam}(\mathbf{C}_r)}, \tag{11}$$

and let

$$\bar{\alpha}(c) = \max_{\mathbf{U} \in \mathbf{M}_{hcn}} \alpha(c, \mathbf{U}). \tag{12}$$

With SL clustering, only one partition $\mathbf{U}$ is computed at each c; thus, (12) reduces to (11). By Dunn's definition, if $\bar{\alpha}(c) > 1$ then $\mathbf{U}$ partitions $\mathbf{X}$ into CS clusters. We use Dunn's CS index to describe the CS property of the clusters in examples 5.1 and 5.2

5.1 Six Gaussian clouds

This example will show how SL and VAT detect two-dimensional symmetrical Gaussian-distributed clusters. Table 1 shows the statistics of each of the six Gaussian-distributed clouds shown in Fig. 3a. Note the covariance of each cloud is $\sigma^2 I_2$, where I_2 is the 2×2 identity matrix. DI for the visually apparent crisp 6-partition of this data is 1.2, so by Dunn's definition, this data contain six CS clusters.

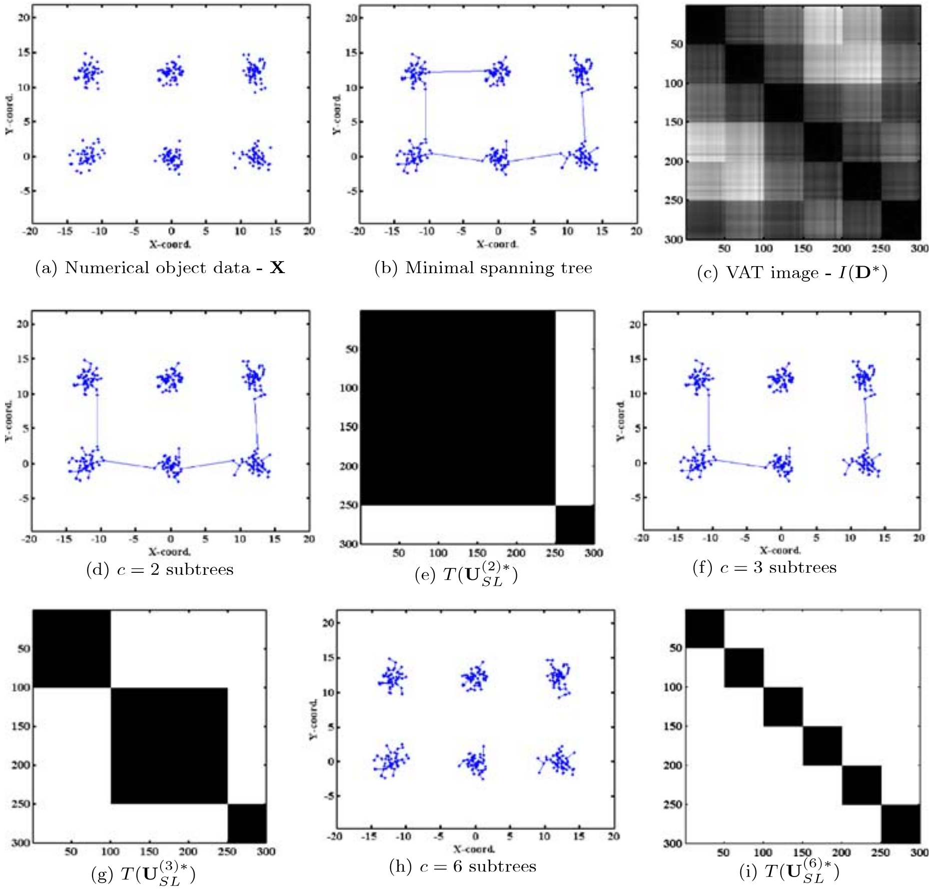
Figure 3b shows the minimal spanning tree for this data set and Fig. 3c shows the VAT image. The VAT image clearly shows six dark blocks on the diagonal, indicating a tendency for six clusters. Figure 3d–i illustrates the results of SL clustering for $c = 2$ clusters, $c = 3$ clusters, and $c = 6$ clusters. In order to show that each partition of the VAT-reordered data is aligned, we show the image of the partition transformation $T(\mathbf{U}_{SL}^{(c)*})$, where c is the number of clusters, SL indicates single linkage, and $*$ indicates that the partition is of VAT-reordered objects. These figures clearly show that the SL partitions at $c = 2$, $c = 3$, and $c = 6$ of the VAT-reordered objects are aligned (as they must be according to Proposition 1); the clusters appear as c contiguous dark blocks along the diagonal of the image of $T(\mathbf{U}_{SL}^{(c)*})$.

5.2 Three lines

This data set consists of 100 numerical objects arranged as three "parallel lines". Figure 4a illustrates this data set. Clearly, there are three clusters in the form of long strings of closely-spaced objects. However, DI for the visually appealing 3-partition of this data (i.e., the partition that groups together all the points along each line) is just 0.12. Consequently, the visually apparent clusters are not CS clusters. This does not, of course, preclude the possibility that the three lines data have some as yet

Table 1 Six Gaussian clouds data characteristics

No. of objects	Mean, μ	Var., $\sigma^2 I_2$	Indices in VAT
50	(−12,0)	1	151–200
50	(0,0)	1	101–150
50	(0,12)	1	51–100
50	(−12,12)	1	201–250
50	(0,12)	1	251–300
50	(12,12)	1	1–50



(a) Numerical object data - $\mathbf{X}$

(b) Minimal spanning tree

(c) VAT image - $I(\mathbf{D}^*)$

(d) $c = 2$ subtrees

(e) $T(\mathbf{U}_{SL}^{(2)*})$

(f) $c = 3$ subtrees

(g) $T(\mathbf{U}_{SL}^{(3)*})$

(h) $c = 6$ subtrees

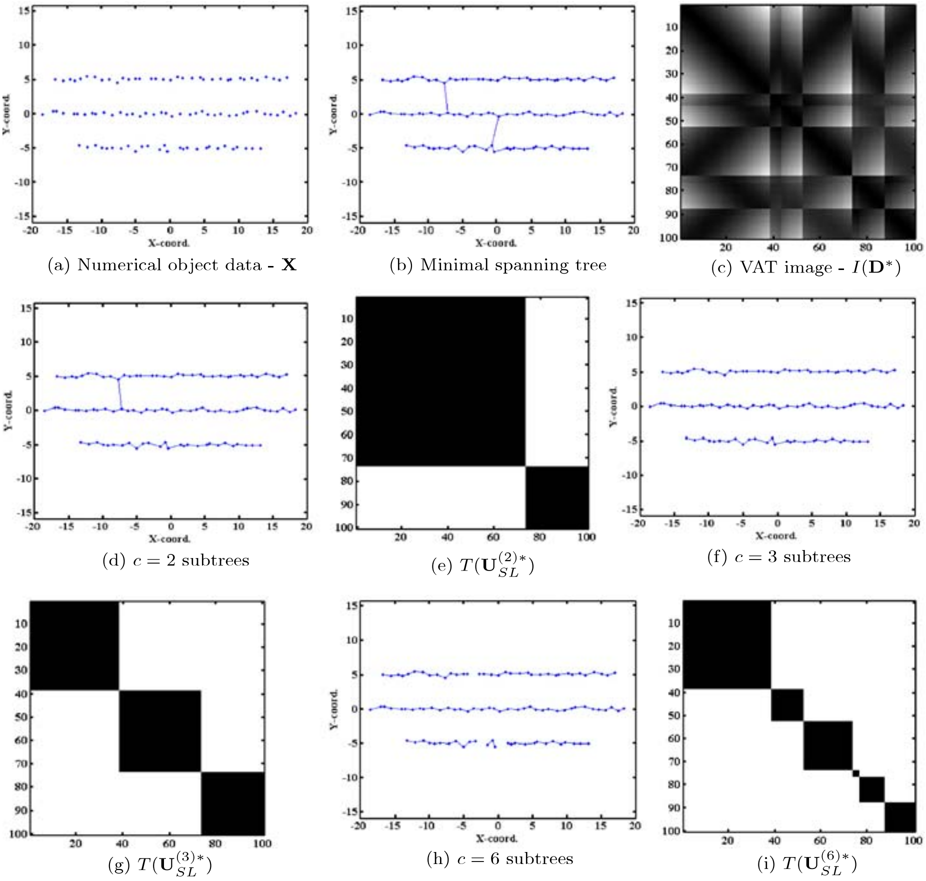
(i) $T(\mathbf{U}_{SL}^{(6)*})$

Fig. 3 Six Gaussian clouds example (**a**–**i**)

undiscovered c-partition for which Dunn's index is greater than 1, but it is hard to imagine that such could be the case. However, these data do display a strength of SL clustering: the ability to label long stringy groups of objects as belonging to the same cluster. Contrastively, this is a case where VAT fails to indicate the tendency to three clusters. Figure 4c shows the VAT image of these data. We leave it to you, the reader, to determine how many clusters you see in the VAT image—it surely is not three!

We show the resulting subtrees of the numerical object data in Fig. 4d, f, h as a result of SL clustering and the respective images of the transformation $T(\mathbf{U}_{SL}^{(c)*})$ in Fig. 4e, g, i. Notice that the $c = 2$ and $c = 3$ cases show that the VAT image has definite edges at the cut locations. In addition, despite the fact that $c = 6$ clusters is obviously a poor choice for this data set, the $c = 6$ partition of the VAT-reordered data is still aligned. This is the case for *all* choices of c, where $1 \leq c \leq 100$ for this data set.

This example brings up an interesting question regarding the efficacy of VAT to show the cluster tendency of data such as this—why is there such a dichotomy



(a) Numerical object data - $\mathbf{X}$

(b) Minimal spanning tree

(c) VAT image - $I(\mathbf{D}^*)$

(d) $c = 2$ subtrees

(e) $T(\mathbf{U}_{SL}^{(2)*})$

(f) $c = 3$ subtrees

(g) $T(\mathbf{U}_{SL}^{(3)*})$

(h) $c = 6$ subtrees

(i) $T(\mathbf{U}_{SL}^{(6)*})$

Fig. 4 Three parallel lines example (**a**–**i**)

between the strength of SL with long stringy clusters and the weakness of VAT to show the tendency of these clusters? Moreover, when VAT fails, [VAT + CLODD] must also fail to deliver the "preferred" SL partition of **D**. We did not apply the usual validation heuristic to the SL hierarchy to see if it selects the visually preferable 3-partition of **D**, but Fig. 4f makes it clear that SL does find this solution. We plan to address this question in the future.

5.3 Bioinformatics data

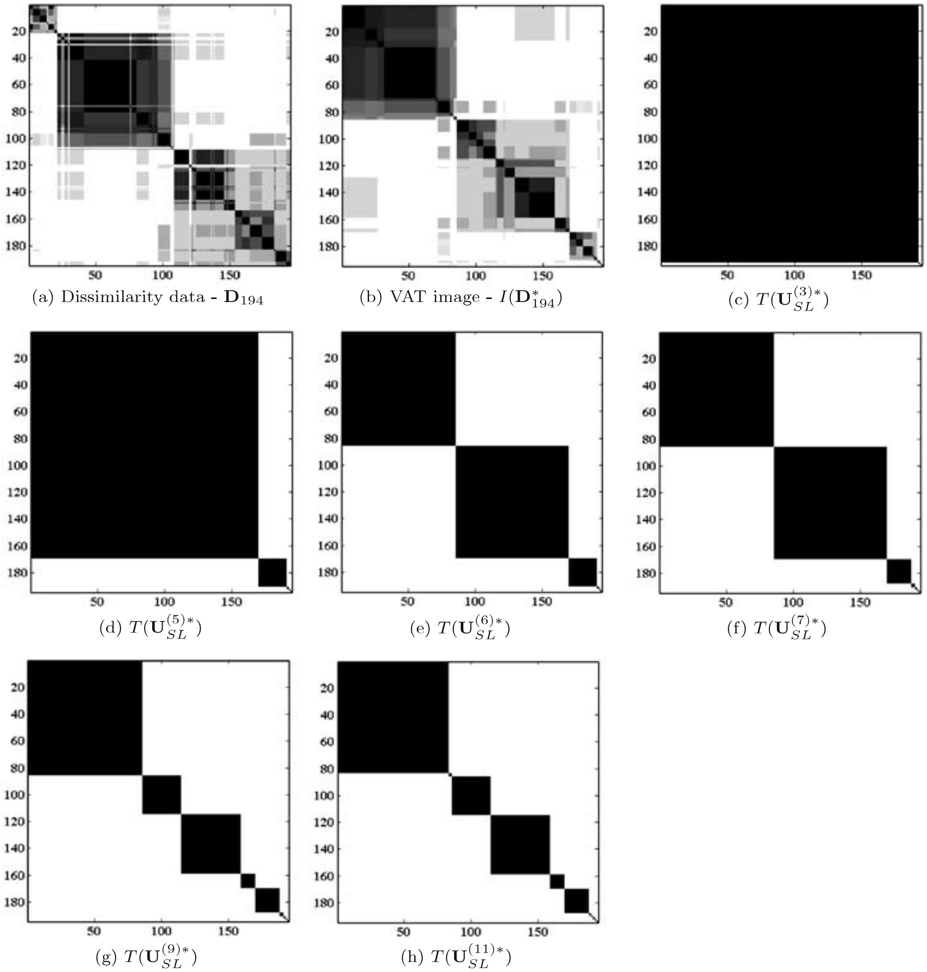
This example uses one version of a real world data set $GPD194_{12.10.03}$, denoted here as $\mathbf{D}_{194}$. These data are different from those of the previous examples in that those are derived from object data, while these data are derived directly from a (dis)similarity relation built with a fuzzy measure applied to annotations of 194 human gene products which appear in the Gene Ontology [25]. Popescu et al. [22] contains a detailed description of the construction of this data, which is summarized

Table 2 Characteristics of the $GPD194_{12.10.03}$ data set extracted from ENSEMBL [27]

ENSEMBL family ID	F_i = Protein family	N_i = No. of sequences	Indices in Fig. 5a
339	Myotubularin	21	1–21
73	Receptor precursor	87	22–108
42	Collagen alpha chain	86	109–194

in Table 2. The data, $\mathbf{D}_{194}$, are displayed in Fig. 5a and the VAT image is shown in Fig. 5b. The three ENSEMBL families are clearly visible in both views.

The images of the SL partition transformation $T(\mathbf{U}_{SL}^{(c)*})$ are displayed in Fig. 5d–h. First, you can see that each partition of the VAT-reordered data is aligned. Second, the images show an interesting aspect of the SL clustering of this data. SL is known to be very susceptible to outliers and this is evident in these views. The $c = 3$ partition, displayed in Fig. 5c, shows that SL clustering partitions the data into one large cluster



(a) Dissimilarity data - $\mathbf{D}_{194}$

(b) VAT image - $I(\mathbf{D}_{194}^{*})$

(c) $T(\mathbf{U}_{SL}^{(3)*})$

(d) $T(\mathbf{U}_{SL}^{(5)*})$

(e) $T(\mathbf{U}_{SL}^{(6)*})$

(f) $T(\mathbf{U}_{SL}^{(7)*})$

(g) $T(\mathbf{U}_{SL}^{(9)*})$

(h) $T(\mathbf{U}_{SL}^{(11)*})$

Fig. 5 Bioinformatics example (**a**–**h**)

and two very small clusters (located at the bottom right of the image). This is due to four gene products, which are outliers in this data set. Our previous research has shown that these four outlier gene products, indexed as 120, 121, 30, and 107, actually cluster into three clusters, namely {120, 121}, {30}, and {107} [22]. The $c = 6$ partition, shown in Fig. 5e, shows that SL clustering partitions this data set into the expected three ENSEMBL families (the receptor precursor family is the first block, the collagen alpha chain is the second, and the myotubularins are the third, from the top of the image), plus the three outlier clusters. Figure 5f–h illustrates that the families have further substructure. The tendency of the collagens to break up into three groups, as shown in Fig. 5g, is supported by [19].

In summary, these three examples confirm the relationship of VAT and SL that is described analytically in Section 4—namely, that SL clusters of VAT-reordered data are aligned partitions.

6 Conclusions

Our analysis has proven that the VAT algorithm and SL hierarchical clustering are intimately related. VAT reorders the relational data with a modified PA that is directly related to the MST. We leveraged the relation of the MST to both VAT and SL clustering to prove that SL clustering always produces aligned clusters of the VAT-reordered data for the case of a unique MST.

We would like to note that in cases of data that may have more than one MST, SL will still, in practice, produce aligned clusters of VAT-reordered data. Simply put, if both VAT and SL are initialized so that they lead to the same MST, our analysis applies. We are currently exploring this topic and, in the future, we plan to generalize the results of this paper to all, including *rectangular*, relational data sets.

We supported our analysis with three numerical examples: two examples used numerical object data with very different cluster structure while the third example concerned a small, real world data set that is purely relational. Each example confirmed our proposition that, indeed, SL clusters are aligned partitions of the VAT-reordered object data. One of the more intriguing questions raised by examples 5.1 and 5.2 is whether or not Dunn's CS index is enough to characterize the data sets for which [VAT+CLODD] is in fact equivalent to an automatic way to extract the "preferred" SL clusters from **D**.

In conclusion, the results of the analysis in this paper are important to furthering the understanding of clustering-related algorithms that are based on VAT and, more generally, Prim's algorithm. These algorithms include, but are not limited to, sVAT-*scalable VAT* [11], bigVAT [15], reVAT-*revised VAT* [14], coVAT [2], VCV-*Visual Cluster Validity* [4, 10, 13], CLODD-*CLustering in Ordered Dissimilarity Data* [12], CCV-*Correlation Cluster Validity* [21], and CCE-*Cluster Count Extraction* [24, 28].

References

1. Bezdek, J., Hathaway, R.: VAT: a tool for visual assessment of (cluster) tendency. In: Proc. IJCNN 2002, pp. 2225–30. Piscataway (2002)
2. Bezdek, J., Hathaway, R., Huband, J.: Visual assessment of clustering tendency for rectangular dissimilarity matrices. IEEE Trans. Fuzzy Syst. **15**(5), 890–903 (2007)

3. Bezdek, J., Keller, J., Krishnapuram, R., Pal, N.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer, Norwell (1999)
4. Ding, Y., Harrison, R.: Relational visual cluster validity (RVCV). Pattern Recogn. Lett. **28**, 2071–2079 (2007)
5. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. Wiley, New York (2000)
6. Dunn, J.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J. Cybern. **3**(3), 32–57 (1974)
7. Gower, J., Ross, G.: Minimum spanning trees and single linkage cluster analysis. Appl. Stat. **18**, 54–64 (1969)
8. Harary, F.: Graph Theory. Addison-Wesley, Reading (2004)
9. Hartigan, J.: Clustering Algorithms. Wiley, New York (1975)
10. Hathaway, R., Bezdek, J.: Visual cluster validity for prototype generator clustering models. Pattern Recogn. Lett. **24**, 1563–1569 (2003)
11. Hathaway, R., Bezdek, J., Huband, J.: Scalable visual asseessment of cluster tendency for large data sets. Pattern Recogn. **39**(7), 1315–1324 (2006)
12. Havens, T., Bezdek, J., Keller, J., Popescu, M.: Clustering in ordered dissimilarity data. Int. J. Intell. Syst. **24**(5), 504–528 (2009)
13. Huband, J., Bezdek, J.: Computational Intelligence: Research Frontiers, chap. VCV2—Visual Cluster Validity, pp. 293–308. Springer, Berlin (2008)
14. Huband, J., Bezdek, J., Hathaway, R.: Revised visual assessment of (cluster) tendency (reVAT). In: Proc. NAFIPS, pp. 101–104. IEEE Press, Banff (2004)
15. Huband, J., Bezdek, J., Hathaway, R.: bigVAT: visual assessment of cluster tendency for large data sets. Pattern Recogn. **38**(11), 1875–1886 (2005)
16. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
17. Kruskal, J.: On the shortest spanning subtree of a graph and the traveling salesman problem. In: Proc. of the Am. Math. Soc., vol. 7, pp. 48–50 (1956)
18. Lynch, N.: Distributed Algorithms, 4th edn. Morgan Kaufmann, San Fransisco (1996)
19. Myllyharju, J., Kivirikko, K.: Collagens, modifying enzymes, and their mutation in humans, flies, and worms. Trends Genet. **20**(1), 33–43 (2004)
20. Notsu, A., Ichihashi, H., Honda, K., Katai, O.: Visualization of balancing systems based on naive psychological approaches. AI Soc. **23**(2), 281–296 (2007)
21. Popescu, M., Bezdek, J., Keller, J., Havens, T., Huband, J.: A new cluster validity measure for bioinformatics relational datasets. In: Proc. FUZZ-IEEE, pp. 726–731. Hong Kong, China (2008)
22. Popescu, M., Keller, J., Mitchell, J., Bezdek, J.: Functional summarization of gene product clusters using Gene Ontology similarity measures. In: Proc. 2004 ISSNIP, pp. 553–559. IEEE, Piscataway (2004)
23. Prim, R.: Shortest connection networks and some generalisations. Bell Syst. Tech. J. **36**, 1389–1401 (1957)
24. Sledge, I., Havens, T., Huband, J., Bezdek, J., Keller, J.: Finding the number of clusters in ordered dissimilarities. Soft Comput. **13**, 1125–1142 (2009)
25. The Gene Ontology Consortium: The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res. **32**, D258–D261 (2004)
26. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 3rd edn. Academic, San Diego (2006)
27. Hubbard, T.J.P., et al.: Ensembl 2009. Nucleic Acids Res. **37**, D690–D697 (2009)
28. Wang, L., Leckie, C., Rao, K., Bezdek, J.: Automatically determining the number of clusters from unlabeled data sets. IEEE Trans. Knowl. Data Eng. **21**(3), 335–350 (2009)
29. Yang, S., Luo, S., Li, J.: Advanced data mining and applications. In: Lecture Notes in Computer Science, vol. 4093, chap. A novel visual clustering algorithm for finding community in complex systems, pp. 396–403. Springer, Berlin (2006)