iVAT and aVAT: Enhanced Visual Analysis for Cluster Tendency Assessment

Liang Wang[1], Uyen T.V. Nguyen[2], James C. Bezdek[2], Christopher A. Leckie[2], and Kotagiri Ramamohanarao[2]

[1] Department of Computer Science
University of Bath, BA2 7AY, United Kingdom
lw356@cs.bath.ac.uk
[2] Department of Computer Science and Software Engineering
The University of Melbourne, Victoria 3010, Australia
{thivun,caleckie,rao}@csse.unimelb.edu.au

Abstract. Given a pairwise dissimilarity matrix $\mathbf{D}$ of a set of n objects, visual methods (such as VAT) for cluster tendency assessment generally represent $\mathbf{D}$ as an $n \times n$ image $\mathrm{I}(\tilde{\mathbf{D}})$ where the objects are reordered to reveal hidden cluster structure as dark blocks along the diagonal of the image. A major limitation of such methods is the inability to highlight cluster structure in $\mathrm{I}(\tilde{\mathbf{D}})$ when $\mathbf{D}$ contains highly complex clusters. To address this problem, this paper proposes an improved VAT (iVAT) method by combining a path-based distance transform with VAT. In addition, an automated VAT (aVAT) method is also proposed to automatically determine the number of clusters from $\mathrm{I}(\tilde{\mathbf{D}})$. Experimental results on several synthetic and real-world data sets have demonstrated the effectiveness of our methods.

Keywords: Visual cluster analysis, cluster tendency assessment, VAT, path-based distance, chamfer matching.

1 Introduction

A general question in the pattern recognition and data mining community is how to organize observed data into meaningful structures or taxonomies. As such, cluster analysis aims at grouping objects of a similar kind into their respective categories. Given a data set $\mathcal{O}$ comprising n objects $\{o_1, o_2, \cdots, o_n\}$, (crisp) clustering partitions the data into c groups $C_1, C_2, \cdots, C_c$, so that $C_i \cap C_j = \emptyset$, if $i \neq j$ and $C_1 \cup C_2 \cup \cdots \cup C_c = \mathcal{O}$. There have been a large number of clustering algorithms reported in the recent literature [1]. In general, clustering of unlabeled data poses three major problems: (1) assessing cluster tendency, *i.e.*, how many groups to seek or what is the value of c? (2) partitioning the data into c groups; and (3) validating the c clusters discovered. Given a pairwise dissimilarity matrix $\mathbf{D} \in \mathcal{R}^{n \times n}$ of $\mathcal{O}$, this paper addresses the problem of determining the number of clusters *prior* to clustering.

M.J. Zaki et al. (Eds.): PAKDD 2010, Part I, LNAI 6118, pp. 16–27, 2010.

Most clustering algorithms require the number of clusters c as an input, so the quality of the resulting clusters is largely dependent on the estimation of c. Various attempts have been made to estimate c. However, most existing methods are *post-clustering* measures of cluster validity [2,1,3,4,5,6,7]. In contrast, tendency assessment attempts to estimate c before clustering occurs. Visual methods for cluster tendency assessment [8,9,10,11,12,13,14,15] generally represent pairwise dissimilarity information about a set of n objects as an $n \times n$ image, where the objects are reordered so that the resulting image is able to highlight potential cluster structure in the data. A "useful" reordered dissimilarity image (RDI) highlights potential clusters as a set of "dark blocks" along the diagonal of the image, and can be viewed as a visual aid to tendency assessment.

Our work is built upon one method for generating reordered dissimilarity images, namely VAT (Visual Assessment of cluster Tendency) of Bezdek and Hathaway [8]. Several algorithms extend VAT for related assessment problems. For example, bigVAT [13] and sVAT [11] offer different ways to approximate the VAT reordered dissimilarity image for very large data sets. CCE [16] and DBE [17] use different schemes to automatically estimate the number of clusters in the VAT images. In addition, Havens *et al.* [18] perform data clustering in ordered dissimilarity images, and coVAT [10] extends the VAT idea to rectangular dissimilarity data. Naturally, the performance of these VAT-based methods is greatly dependent of the quality of the VAT images. However, while VAT has been widely used for cluster analysis, it is usually only effective at highlighting cluster tendency in data sets that contain compact well-separated clusters. Many practical applications involve data sets with highly irregular structure, which invalidate this assumption. In this paper, we propose an improved VAT (iVAT) approach to generating RDIs that combines VAT with a path-based distance transform. The resulting iVAT images can clearly show the number of clusters and their approximate sizes for data sets with highly complex cluster structures. We also propose a new strategy for automated determination of the number of clusters c from RDIs, by detecting and counting dark blocks along the main diagonal of the image. Experimental results on both synthetic and real-world data sets validate our methods.

The remainder of the paper is organized as follows: Section 2 briefly reviews the VAT algorithm. Section 3 illustrates our iVAT algorithm. Section 4 presents our strategy for automatically determining the number of clusters c. The experimental results on both synthetic and real-world data sets are given and analyzed in Section 5, prior to conclusion in Section 6.

2 VAT

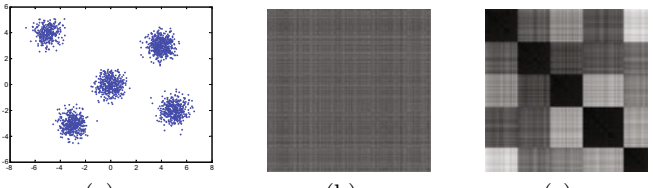
Let $\mathcal{O} = \{o_1, o_2, \cdots, o_n\}$ denote n objects in the data and $\mathbf{D}$ a pairwise matrix of dissimilarities between objects, each element of which $d_{ij} = d(o_i, o_j)$ is the dissimilarity between objects o_i and o_j, and generally, satisfies $1 \geq d_{ij} \geq 0; d_{ij} = d_{ji}; d_{ii} = 0$, for $1 \leq i, j \leq n$. Let $\pi()$ be a permutation of $\{1, 2, \cdots, n\}$ such that $\pi(i)$ is the new index for o_i. The reordered list is thus $\{o_{\pi(1)}, \cdots, o_{\pi(n)}\}$. Let $\mathbf{P}$

be the permutation matrix with $p_{ij} = 1$ if $j = \pi(i)$ and 0 otherwise, then the matrix $\tilde{\mathbf{D}}$ for the reordered list is a similarity transform of $\mathbf{D}$ by $\mathbf{P}$, *i.e.*,

$$\tilde{\mathbf{D}} = \mathbf{P}^T \mathbf{D} \mathbf{P}.$$

The reordering idea is to find $\mathbf{P}$ so that $\tilde{\mathbf{D}}$ is as close to a block diagonal form as possible. The VAT algorithm [8] reorders the row and columns of $\mathbf{D}$ with a modified version of Prim's minimal spanning tree algorithm, and displays a reordered dissimilarity matrix $\tilde{\mathbf{D}}$ as a gray-scale image. If an object is a member of a cluster, then it should be part of a sub-matrix with low dissimilarity values, which appears as one of the dark blocks along the diagonal of the VAT image $\mathrm{I}(\tilde{\mathbf{D}})$, each of which corresponds to one potential cluster.

Figure 1(a) is a scatter plot of 2000 data points in $\mathcal{R}^2$. The 5 visually apparent clusters are reflected by the 5 distinct dark blocks along the main diagonal in Figure 1(c), which is the VAT image of the data. Given the image of $\mathbf{D}$ in the original input order in Figure 1(b), reordering is necessary to reveal the underlying cluster structure of the data. VAT reordering produces neither a partition nor a hierarchy of clusters. It merely reorders the data to (possibly) reveal its hidden structure, which can be viewed as an illustrative data visualization for estimating c. Sometimes, hierarchical structure can be detected by the presence of diagonal sub-blocks within larger diagonal blocks.



(a) (b) (c)

Fig. 1. An example of the VAT algorithm

3 Improved VAT (iVAT)

At a glance, a viewer can estimate the number of clusters c from a VAT image by counting the number of dark blocks along the diagonal if these dark blocks possess visual clarity. However, this is not always possible. Note that a dark block appears only when a compact group exists in the data. For complex-shaped data sets where the boundaries between clusters become less distinct due to either significant overlap or irregular geometries, the resulting VAT images may fail to produce dark blocks even when cluster structure is clearly present. See Figures 4(b) and 5(a) for examples. Different viewers may deduce different numbers of clusters from such poor-quality images, or worse, not be able to estimate c at all. This raises the question of whether we can transform $\mathbf{D}$ into a new form $\mathbf{D}'$ so that the VAT image of $\mathbf{D}'$ is clearer and more informative about the cluster structure.

In [12], SpecVAT combines VAT with graph embedding [19,20] to solve this problem. SpecVAT first embeds the data into a k-dimensional subspace spanned by the eigenvectors of the normalized Laplacian matrix and then re-computes a new pairwise dissimilarity matrix in the embedding subspace as the input of the VAT algorithm. However, this method depends on two main parameters, one of which is r for the r-th nearest neighbor based local scale computation when constructing the affinity matrix from $\mathbf{D}$ [21] (for deriving the Laplacian matrix), and the other is k, the number of eigenvectors used. In particular, k largely depends on the number of clusters c. Figure 2 gives an example of SpecVAT images with respect to different values of k. Since c is unknown, a range of k values need to be used for generating a series of SpecVAT images to find the 'best' SpecVAT image that is truly informative of the real structure in the data (*e.g.*, $k = 3$ in this case). In contrast, this work adopts a *parameter-free* method (called iVAT) by combining VAT with a path-based distance transform.

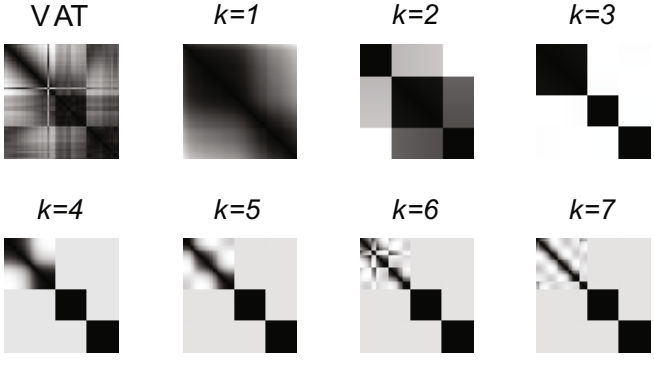


Fig. 2. An example of SpecVAT on synthetic three-circle data set S-3 ($c = 3$)

The path-based dissimilarity measure was introduced in [22]. The intuitive idea is that if two objects o_i, o_j are very far from each other (reflected by a large distance value d_{ij}), but there is a path connecting them consisting 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. The adjustment of d_{ij} 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. This reflects the characteristic of elongated clusters.

Let us treat $\mathbf{D}$ as a fully connected graph $\mathcal{G}$, where each vertex corresponds to an object and the edge weight between vertices i and j is the distance d_{ij}. Suppose that P_{ij} is the set of all possible paths from o_i to o_j, then for each path $p \in P_{ij}$, the effective dissimilarity between objects o_i and o_j along p is the maximum of all edge weights belonging to this path. The path-based distance d'_{ij} is defined as

$$d'_{ij} = \min_{p \in P_{ij}} \{ \max_{1 \le h < |p|} d_{p[h]p[h+1]} \}$$

where $p[h]$ denotes the object at the h-th position in path p and $|p|$ denotes the length of path p. After obtaining $\mathbf{D}' = [d'_{ij}]$, we reorder it using the VAT algorithm to obtain the iVAT image (see examples in Figures 4(c) and 5(b)). The iVAT images are almost always clearer and more informative than the original VAT images in revealing the data structure.

4 Automated VAT (aVAT)

A viewer can simply estimate the number of clusters c (*i.e.*, count the number of dark blocks along the diagonal of a RDI image if these dark blocks possess visual clarity). However, as the boundaries between different clusters become less distinct, the RDI image will degrade considerably with confusing boundaries between potential dark blocks. Accordingly, different viewers may deduce different numbers of clusters from such poor-quality images. Can we automatically determine the number of clusters c, as suggested by $\mathrm{I}(\tilde{\mathbf{D}}')$, in an objective manner, without viewing the visual display? To answer this interesting question, two methods have been developed, *i.e.*, DBE [17] and CCE [16] (see the algorithm details and comparison in Section 5.3). Here we propose an alternative method, called aVAT, using some image processing techniques. The process of aVAT is illustrated in Figure 3. The individual steps are all well known in the field of image processing, so we do not describe their underlying theories.

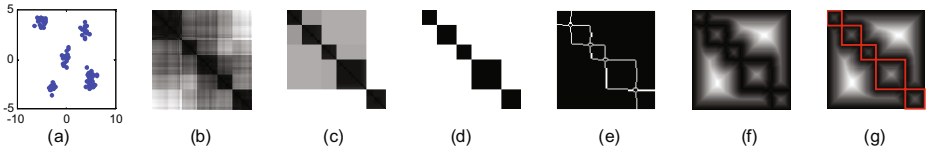


Fig. 3. Illustration of the aVAT algorithm. From left to right: scatter plot ($n = 100, c = 5$), VAT image of (a), iVAT image of (a), binarized image of (c), edge map of (d), DT image of (e), and detected squares in (f) imposed using red lines.

Since information about possible cluster structure in the data is embodied in the *square dark blocks* along the diagonal of a RDI, we propose to detect and count them using shape-based Chamfer matching [23]. As a preprocessing step, the RDI is firstly binarized to extract regions of interest (Figure 3(d)). Otsu's method [24] is used to automatically choose a global threshold. To make within-cluster distances smaller and between-cluster distances larger (*i.e.*, increasing contrast) to obtain a more reliable threshold, we transform the image intensities using a "monotonic" function

$$f(t_{xy}) = 1 - \exp(-t_{xy}^2/\sigma^2)$$

where t_{xy} denotes the intensity value of the image pixel on the location (x, y), and σ is empirically set as the mean value of all pixel intensities.

Chamfer matching was first proposed by Barrow *et al.* [25]. Assume that two point sets are $\mathcal{U} = \{\mathbf{u}_i\}_{i=1}^{N}$ and $\mathcal{V} = \{\mathbf{v}_i\}_{i=1}^{M}$, the chamfer distance is defined as

$$d_{cham}(\mathcal{U}, \mathcal{V}) = \frac{1}{N} \sum_{\mathbf{u}_i \in \mathcal{U}} \min_{\mathbf{v}_j \in \mathcal{V}} \|\mathbf{u}_i - \mathbf{v}_j\|.$$

The symmetric chamfer distance can be obtained by adding $d_{cham}(\mathcal{V},\mathcal{U})$. The chamfer distance between two shapes can be efficiently computed using a distance transform (DT, Figure 3(f)), which takes a binary image as input, and assigns to each pixel in the image the distance to its nearest feature. We use Canny edges as image feature points (Figure 3(e)) and the Euclidean distance for DT, and the model points are the projected contours of a 2D (rigid) square template. The distance between the template and the edge map can then be computed as the mean of the DT values at the template point coordinates.

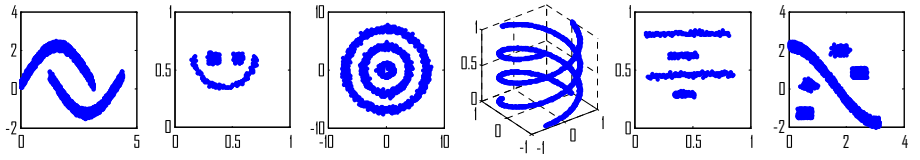
In general, matching consists of translating, rotating and scaling the template shape at various locations of the distance image. Fortunately, in the RDI, we just need to search for squares along the diagonal axis and scale the template to different sizes to adapt to various cluster sizes. There is no need for template rotation, because there are no orientation changes in VAT RDIs. This greatly reduces the complexity of common shape detection using Chamfer matching. The exact matching cost is ideally 0, but in practice the edges in an image are slightly displaced from their ideal locations. Thus in our experiments, when the matching cost lies below a certain threshold τ, the target shape is considered to have been detected (Figure 3(g)).

5 Experimental Results

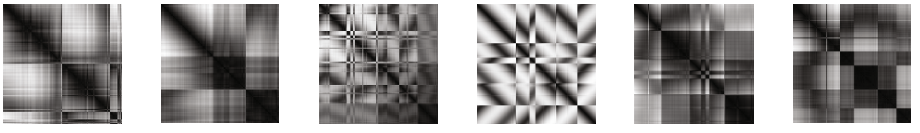
In order to evaluate our methods, we have carried out a number of experiments on 6 artificially generated data sets, as well as 6 real-world data sets. Unless otherwise mentioned, in the following experiments the (Euclidean) distance matrix **D** was computed in the attribute space (if the object vectorial representation is available). All experiments were implemented in a Matlab 7.2 environment on a PC with an Intel 2.4GHz CPU and 2GB memory running Windows XP.

5.1 Test Datasets

Six synthetic data sets were used in our experiments, whose scatter plots are shown in Figure 4(a). These data sets involve irregular data structures, in which an obvious cluster centroid for each group is not necessarily available. Six real-world data sets were also considered to evaluate our algorithms, 3 of which were taken from the UCI Machine Learning Repository, *i.e.,* Iris, Vote and Multiple Features. The *Face* data set [26] contains 1755 images of 3 individuals, each of which was down-sampled to 30×40 pixels. The *Gene* data set [27] is a 194×194 matrix consisting of pairwise dissimilarities of a set of gene products from 3 protein families. The *Iris* data set contains 3 types of iris plants, 50 instances each. The *Vote* data set consists of 435 vote records (267 democrats and 168



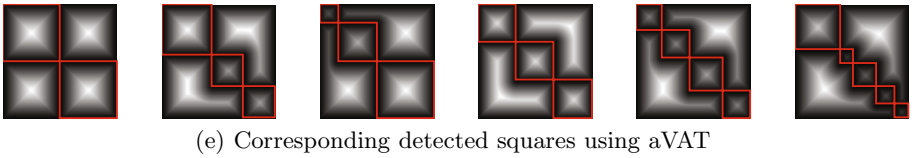
(a) Scatter plots. From left to right: S-1 to S-6



(b) Corresponding VAT images

(c) Corresponding iVAT images

(d) Corresponding binarized images of iVAT images



(e) Corresponding detected squares using aVAT

Fig. 4. Visual analysis on 6 synthetic data sets

republicans). Votes were numerically encoded as 0.5 for "yea", -0.5 for "nay" and 0 for "unknown disposition". The *Action* data set [28] is an 198×198 pairwise dissimilarity matrix derived from 198 human action clips. The *Multiple-Features* (MF) data set consists of binary image features of 10 handwritten numerals, 200 patterns per class. The characteristics of these synthetic and real data sets are summarized in Table 1.

5.2 Results and Analysis

For each of the data sets, we first applied the VAT algorithm. The VAT images are shown in Figure 4(b) for synthetic data and Figure 5(a) for real data, respectively. It can be seen that the cluster structure of the data in the VAT images is not necessarily clearly highlighted, especially for complex-shaped data. Accordingly, viewers have difficulties in giving a consistent result about the number of clusters, and different viewers may deduce different estimates of c. Next

Table 1. Summary of results of estimating c using different methods

Data				VAT				iVAT			
Name	c_p	# attri.	n	Manual	DBE	CCE	aVAT	Manual	DBE	CCE	aVAT
S-1	2	2	2000	≥ 1	6	*3*	6	**2**	**2**	**2**	**2**
S-2	3	2	266	≥ 2	5	**3**	**3**	**3**	**3**	**3**	**3**
S-3	3	2	1800	≥ 1	*10*	11	11	**3**	**3**	**3**	**3**
S-4	3	3	3000	-	8	9	*7*	**3**	**3**	**3**	**3**
S-5	4	2	512	≥ 1	*5*	*5*	*5*	**4**	**4**	**4**	**4**
S-6	5	2	2500	≥ 5	8	*6*	**5**	**5**	**5**	**5**	**5**
Vote	2	16	435	≥ 2	**2**	**2**	*3*	**2**	**2**	*3*	**2**
Iris	3	4	150	*2*	**3**	*2*	*2*	*2*	*2*	*2*	*2*
Gene	3	-	194	≥ 3	**3**	*4*	**3**	**3**	5	**3**	*4*
Face	3	1200	1755	3 or 4	*4*	**3**	6	*4*	*4*	5	5
Action	10	-	198	≥ 9	*8*	7	7	**10**	*7*	*7*	*7*
MF	10	649	2000	≥ 8	*9*	5	12	8	7	6	*9*
AAE				-	2.17	2.25	2.25	0.33	0.83	0.92	0.67
ARE				-	0.73	0.59	0.66	0.07	0.16	0.18	0.14

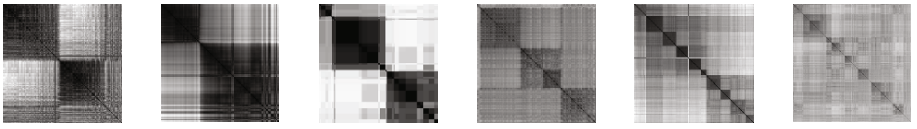
we carried out our iVAT algorithm for each of the data sets used. The resulting iVAT images are shown in Figure 4(c) for synthetic data and Figure 5(b) for real data, respectively. In contrast to the original VAT images, the iVAT images have generally clearer displays in terms of block structure, thus better highlighting the hidden cluster structure.

Table 1 summarizes the number of clusters determined from iVAT images automatically, along with the results estimated from the VAT and iVAT images using manual inspection by the authors for comparison. From Table 1, we can see that

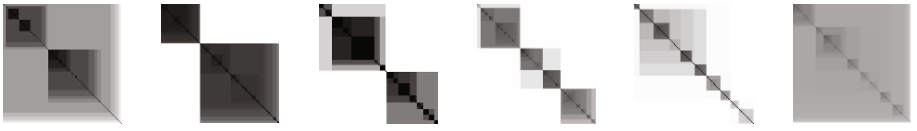
1. The results estimated from the iVAT images by manual inspection are clearly better than those estimated from the original VAT images by manual inspection, whether for synthetic or real-world data sets.
2. The results of cluster number estimation from the iVAT images for all synthetic data sets are accurate in terms of the number of real physical classes (c_p), whether it was estimated automatically by our aVAT algorithm or by manual inspection.
3. For real-world data sets, some estimates deviate slightly from the number of real physical classes using our aVAT algorithm.

Overall, these results highlight the benefits of converting $\mathbf{D}$ to $\mathbf{D'}$ by the path-based distance transform for obtaining a good estimation of c (whether automatically or manually). We would like to note several points:

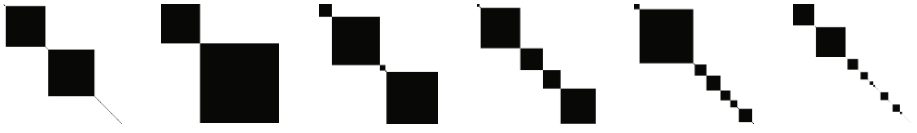
1. Though some estimates are imperfect for the real data, the aVAT algorithm correctly detected all squares in the binarized images (see Figure 5(c)). This suggests that we may need to seek more sophisticated methods of image binarization (*e.g.*, multiple local thresholds) for avoiding the loss of some



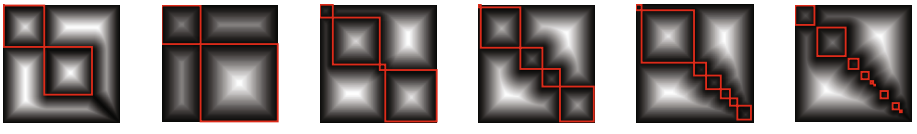
(a) VAT images. From left to right: vote, iris, gene, face, action and MF



(b) Corresponding iVAT images



(c) Corresponding binarized images of iVAT images



(d) Corresponding detected squares using aVAT

Fig. 5. Visual analysis on 6 real data sets

physically meaningful blocks (*e.g.*, for Action data, some small blocks corresponding to different action classes were transferred into a bigger block after binarization).

2. We could use a 'square size' threshold to filter some detected very 'small' blocks corresponding to either noise/inliers or subtle sub-structures (*e.g.*, for Face and Gene data sets).
3. As a side product, for 'perfect' iVAT images (such as those of the 6 synthetic data sets), the exactly detected squares may be directly used to retrieve data partitions (*i.e.*, each square corresponds to one potential cluster and its size corresponds to the cluster size).

5.3 Algorithm Comparison

We compared our aVAT algorithm with CCE [16] and DBE [17] in terms of estimating the number of clusters. Note that we did not compare aVAT to index-based methods for post-clustering assessment of cluster validity, as our interest is in estimating the number of clusters *before* clustering. The major steps for CCE are summarized as follows: 1) Threshold the VAT image with Otsu's algorithm; 2) Apply the FFT to the segmented VAT and a correlation filter of size s and multiply the transformed image with the complex conjugate of the transformed filter; 3) Compute the inverse FFT for the filtered image; 4) Take

the q-th off-diagonal pixel values of the back-transformed image and compute its histogram; and 5) Cut the histogram at a horizontal line $y = b$, and count the number of spikes. The major steps of DBE are summarized as follows: 1) Perform intensity transform and segmentation of the VAT image, followed by directional morphological filtering with size of αn; 2) Apply a distance transform to the filtered image and project the pixel values onto the main diagonal axis to form a projection signal; 3) Smooth the projection signal by an average filter with a length of $2\alpha n$, compute its first-order derivative, and then detect the number of major peaks by ignoring minor ones using a filter with size of $2\alpha n$.

As suggested in [16,17], we used $s = 20$, $q = 1$ and $b = 0$ for CCE and $\alpha = 0.03$ for DBE. We used both VAT and iVAT images to make our comparisons, and the results are summarized in Table 1, in which we used *bold* figures to show that the estimate is equal to the number of real physical classes c_p and *italic* figures to show results that are relatively closer to c_p. AAE and ARE represent average absolute error and average relative error between the number of estimated clusters and the number of real physical classes, respectively. From Table 1, it can be seen that:

1. For synthetic data sets, all of these three methods give correct results when using the iVAT images, while aVAT and CCE are slightly better than DBE when using the original VAT images (*i.e.*, 2 correct and 2 closer for aVAT, 1 correct and 3 closer for CCE, and 2 closer to DBE).
2. For real-world data sets plus the use of VAT images, DBE performs best, then CCE and finally aVAT (*i.e.*, 3 correct and 3 closer for DBE, 2 correct and 2 closer for CCE, and 1 correct and 2 closer for aVAT); while for real-world data sets plus the use of iVAT images, aVAT is a little better than both CCE and DBE (*i.e.*, 1 correct and 4 closer for aVAT, and 1 correct and 3 closer for both CCE and DBE).
3. Specifically, when using iVAT images, aVAT, CCE and DBE yield the same estimate for the Iris and Action data sets. They all yield acceptable (but different) estimates for the Gene and Face data sets. They disagree for the Vote and MF data sets.

Overall, these three methods are comparable to each other and there is no clear winner (at least based on the results on these data sets used currently). However, we can see that the positions of peaks and valleys in the projection signal in DBE *implicitly* correspond to centers and ranges of sub-blocks (or clusters). It is hard to see a similar phenomena from the CCE histograms. In contrast, aVAT is better in this aspect because it *explicitly* shows the number of clusters, positions and ranges of each block (or clusters) within the image itself, in a more intuitive manner.

6 Conclusion

This paper has presented a new visual technique for cluster tendency assessment. Our contributions include: 1) The VAT algorithm was enhanced by using

a path-based distance transform. The iVAT algorithm can better reveal the hidden cluster structure, especially for complex-shaped data sets. 2) Based on the iVAT image, the cluster structure in the data can be reliably estimated by visual inspection. As well, the aVAT algorithm was proposed for automatically determining the number of clusters *c*. 3) We performed a series of primary and comparative experiments on 6 synthetic data sets and 6 real-world data sets, and our methods obtained encouraging results.

In addition to further performance evaluation on more data sets with various structures, future work will mainly focus on increasing the robustness of our algorithms, *e.g.*, exploring more sophisticated image thresholding methods [29] and robust path-based distance computation [30].

References

1. Xu, R., II, D.W.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
2. Maulik, U., Bandyopadhyay, S.: Performance evaluation of some clustering algorithms and validity indices. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(12), 1650–1654 (2002)
3. Hu, X., Xu, L.: A comparative study of several cluster number selection criteria. In: Liu, J., Cheung, Y.-m., Yin, H. (eds.) IDEAL 2003. LNCS, vol. 2690, pp. 195–202. Springer, Heidelberg (2003)
4. Bezdek, J.C., Pal, N.R.: Some new indices of cluster validity. IEEE Transactions on System, Man and Cybernetics 28(3), 301–315 (1998)
5. Tibshirani, R., Hastie, G.W.,, T.: Estimating the number of clusters in a dataset via the gap statistics. Journal of the Royal Statistical Society. Series B, Statistical Methodology 63(2), 411–423 (2001)
6. Calinski, R.B., Harabasz, J.: A dendrite method for cluster analysis. Communications in Statistics 3(1), 1–27 (1974)
7. Dunn, J.C.: Indices of partition fuzziness and the detection of clusters in large sets. Fuzzy Automata and Decision Processes (1976)
8. Bezdek, J.C., Hathaway, R.J.: VAT: A tool for visual assement of (cluster) tendency. In: International Joint Conference on Neural Networks, vol. 3, pp. 2225–2230 (2002)
9. Tran-Luu, T.: Mathematical Concepts and Novel Heuristic Methods for Data Clustering and Visualization. PhD Thesis, University of Maryland, College Park, MD (1996)
10. Bezdek, J.C., Hathaway, R., Huband, J.: Visual assessment of clustering tendency for rectangular dissimilarity matrices. IEEE Transactions on Fuzzy Systems 15(5), 890–903 (2007)
11. Hathaway, R., Bezdek, J.C., Huband, J.: Scalable visual assessment of cluster tendency. Pattern Recognition 39(7), 1315–1324 (2006)
12. Wang, L., Geng, X., Bezdek, J., Leckie, C., Kotagiri, R.: SpecVAT: Enhanced visual cluster analysis. In: International Conference on Data Mining, pp. 638–647 (2008)
13. Huband, J., Bezdek, J.C., Hathaway, R.: bigVAT: Visual assessment of cluster tendency for large data sets. Pattern Recognition 38(11), 1875–1886 (2005)
14. Ling, R.: A computer generated aid for cluster analysis. Communications of the ACM 16(6), 355–361 (1973)

15. Rousseeuw, P.J.: A graphical aid to the interpretations and validation of cluster analysis. Journal of Computational and Applied Mathematics 20(1), 53–65 (1987)
16. Sledge, I., Huband, J., Bezdek, J.C. (Automatic) cluster count extraction from unlabeled datasets. In: Joint International Conference on Natural Computation and International Conference on Fuzzy Systems and Knowledge Discovery, vol. 1, pp. 3–13 (2008)
17. Wang, L., Leckie, C., Kotagiri, R., Bezdek, J.: Automatically determining the number of clusters in unlabeled data sets. IEEE Transactions on Knowledge and Data Engineering 21(3), 335–350 (2009)
18. Havens, T.C., Bezdek, J.C., Keller, J.M., Popescu, M.: Clustering in ordered dissimilarity data. International Journal of Intelligent Systems 24(5), 504–528 (2009)
19. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in Neural Information Processing Systems 14, 585–591 (2002)
20. Chung, F.: Spectral graph theory. In: CBMS Regional Conference Series in Mathematics, American Mathematical Society, vol. 92 (1997)
21. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. Advances in Neural Information Processing Systems 17, 1601–1608 (2004)
22. Fisher, B., Zoller, T., Buhmann, J.: Path based pairwise data clustering with application to texture segmentation. In: Figueiredo, M., Zerubia, J., Jain, A.K. (eds.) EMMCVPR 2001. LNCS, vol. 2134, pp. 235–250. Springer, Heidelberg (2001)
23. Thayananthan, A., Stenger, B., Torr, P., Cipolla, R.: Shape context and chamfer matching in cluttered scenes. In: International Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 127–133 (2003)
24. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics 9(1), 62–66 (1979)
25. Barrow, H.G.: tenenbaum, J.M., Bolles, R.C., Wolf, H.C.: Parametric correspondence and chamfer matching: Two new techniques for image matching. In: Intternational Joint Conference on Artificial Intelligence, vol. 2, pp. 659–663 (1977)
26. Breitenbach, M., Grudic, G.: Clustering through ranking on manifolds. In: International Conference on Machine Learning, vol. 119, pp. 73–80 (2005)
27. Pal, N., Keller, J., Popescu, M., Bezdek, J.C., Mitchell, J., Huband, J.: Gene ontology-based knowledge discovery through fuzzy cluster analysis. Journal of Neural, Parallel and Scientific Computing 13(3-4), 337–361 (2005)
28. Wang, L., Leckie, C., Wang, X., Kotagiri, R., Bezdek, J.: Tensor space learning for analyzing activity patterns from video sequences. In: ICDM Workshop on Knowledge Discovery and Data Mining from Multimedia Data and Multimedia Applications, pp. 63–68 (2007)
29. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging 13(1), 146–165 (2004)
30. Chang, H., Yeung, D.Y.: Robust path-based spectral clustering with application to image segmentation. In: International Conference on Computer Vision, vol. 1, pp. 278–285 (2005)