American Journal of Computational Mathematics, **2012, 2, 27-41**
http://dx.doi.org/10.4236/ajcm.2012.21004 Published Online March 2012 (http://www.SciRP.org/journal/ajcm)

VATdt: Visual Assessment of Cluster Tendency Using Diagonal Tracing

Yingkang Hu
Department of Mathematical Sciences, Georgia Southern University, Statesboro, USA
Email: yhu@georgiasouthern.edu

Received December 31, 2011; revised January 31, 2012; accepted February 10, 2012

ABSTRACT

The visual assessment of tendency (VAT) technique, for visually finding the number of meaningful clusters in data, developed by J. C. Bezdek, R. J. Hathaway and J. M. Huband, is very useful, but there is room for improvements. Instead of displaying the ordered dissimilarity matrix (ODM) as a 2D gray-level image for human interpretation as is done by VAT, we trace the changes in dissimilarities along the diagonal of the ODM. This changes the 2D data structure (matrices) into 1D arrays, displayed as what we call the *tendency curves*, which enables one to concentrate only on one variable, namely the height. One of these curves, called the *d*-curve, clearly shows the existence of cluster structure as patterns in peaks and valleys, which can be caught not only by human eyes but also by the computer. Our numerical experiments showed that the computer can catch cluster structures from the *d*-curve even in some cases where the human eyes see no structure from the visual outputs of VAT. And success on all numerical experiments was obtained using the same (fixed) set of program parameter values.

Keywords: Clustering; Dissimilarity Measures; Data Visualization; Clustering Tendency

1. Introduction

Clustering is the problem of partitioning a set of objects $O=\{o_1,o_2,\cdots,o_n\}$ into c self-similar subsets (clusters) based on available data and some well-defined measure of similarity. The type of clusters found depends strongly on the mathematical model that underlies the clustering algorithm. All clustering algorithms will find any number (up to n) of clusters, even if no meaningful clusters exist. Therefore before choosing a clustering method one has to decide whether there are meaningful clusters, and if so, how many are there. This is called the *assessing of clustering tendency*.

Numerous formal (statistics-based) and informal techniques for such assessment are discussed in Jain and Dubes [1] and Everitt [2]. None of these existing methods are totally satisfactory, nor will they ever be. Visual approaches for assessing clustering tendency have been widely studied in the last few decades; Tukey [3] and Cleveland [4] are standard references for visual approaches in various data analysis problems. Recently the research on the *visual assessment of tendency* (VAT) technique has been quite active; see the original VAT paper by Bezdek and Hathaway [5], also see VATr by Bezdek, Hathaway and Huband [6], sVAT by Hathaway, Bezdek and Huband [7], and reVAT and bigVAT by Huband, Bezdek and Hathaway [8,9].

The object set O is usually represented in the following two ways. When each object o_i is represented by a vector $x_i \in \mathbb{R}^s$, the set $X=\{x_1,x_2,\cdots x_n\}\subset\mathbb{R}^s$ is called an *object data* representation of O. The s components of x_i represent the s features of the object o_i. It is in this feature space that people sometimes seek descriptors of the clusters, *cluster centers* or *prototypes*, as they are called. Alternatively, when each *pair* of objects in O is represented by a relationship, it is called *relational data*. Most of the time, the relationship between o_i and o_j is given by their dissimilarity R_{ij} (a distance or some other measure; see [10,11]). These n^2 data items form a symmetric matrix $R=\left[R_{ij}\right]_{n\times n}$.

Our method, which we call *VATdt*, standing for Visual Assessment of cluster Tendency using diagonal tracing, replaces the visual output of the VAT algorithms (the original one or its variations). VAT applies directly on a dissimilarity matrix. If the original data consist of a (symmetric) matrix of pair-wise similarities $S=\left[S_{ij}\right]_{n\times n}$, then a dissimilarity matrix R can be obtained through a simple transformation such as

$$R_{ij}=S_{\max}-S_{ij},$$

where $S_{\max}$ denotes the largest similarity value. If the original data are represented by object data $X=\{x_1,x_2,\cdots x_n\}$, then R_{ij} can be computed as the distance between x_i and x_j measured by some norm or metric in the feature space $\mathbb{R}^s$. Hence the VAT algorithms can always be applied, and so can our VATdt algorithm. They are applicable even if some components of the original data are missing; see [5] and the references therein. In this paper if the data are given as object data *X*, the dissimilarity matrix *R* will be given by the square root of the Euclidean norm of $x_i - x_j$:

$$R_{ij} = \sqrt{\left\|x_i - x_j\right\|_2} \qquad (1)$$

VAT reorders the points in a data set so that points that are close to one another in the feature space will generally have similar indices (see the example below). Some versions, such as sVAT [7], reduce the size of *R* by choosing a subset of the original set *O*. Their numeric output is an *ordered dissimilarity matrix* (ODM). We will still use the letter *R* for the ODM. This will not cause confusion since it is the only information on the data we are going to use. The ODM satisfies

$$0 \le R_{ij} \le 1, R_{ij} = R_{ji,} \text{ and } R_{ii} = 0.$$

The largest element of *R* is 1 because the VAT algorithms scale the elements of *R*.

VAT displays the ODM on the screen in a straightforward way, as *ordered dissimilarity image* (ODI). In ODI the gray level g_{ij} of pixel (*i*,*j*) is proportional to the value of R_{ij} with $g_{ij} = 0$ (pure black) if $R_{ij} = 0$ and $g_{ij} = 255$ (pure white) if $R_{ij} = 1$. The idea of VAT is shown in the following example.

Example 1. A data set $X \subset \mathbb{R}^2$ of 20 points containing three well-defined clusters is shown in **Figure 1**. As most likely found in applications, the points in each of the clusters are not indexed together. **Figure 2(a)** shows the original (random) order of the points in *X*, with $x_1 \approx (2,10)$ represented by a diamond. The corresponding ODI image in **Figure 2(b)** shows no useful visual information about the structure in *X*. The VAT technique can reorder the points in *X* so that nearby points are (generally) indexed closely. **Figure 3(a)** shows the new order of the data set *X*, with the diamond in the lower left corner representing the first point in the ordered data set. **Figure 3(b)** gives the corresponding ODI. Now the three clusters are represented by the three well-formed black blocks.

The VAT algorithms are certainly very useful, but there is room for improvements. It seems to us that our eyes are not very sensitive to structures in gray level images. One example is given in **Figure 4**. There are three clusters in the data as we will show later. The clusters are not well separated, and the ODI from VAT reveals almost no sign of the existence of the structure.

The approach of this paper is to trace changes in dissimilarities along the diagonal of the ODM, the numeric output of VAT that underlies its visual output ODI. This

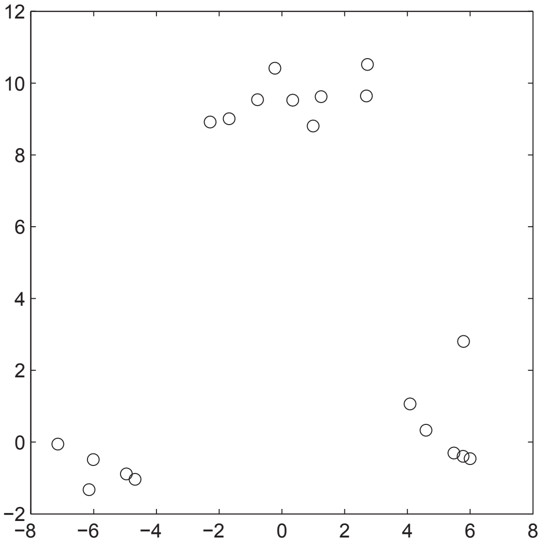
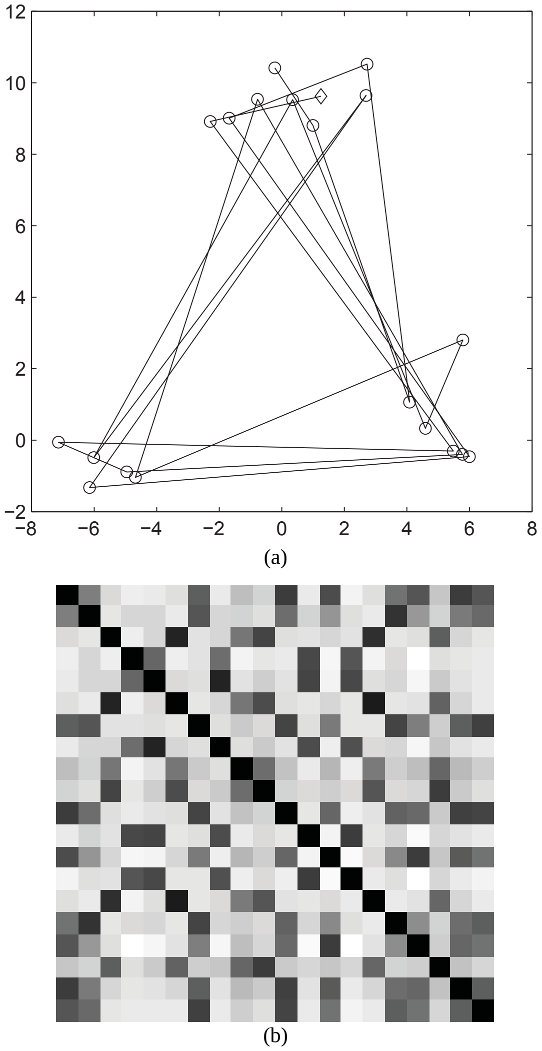


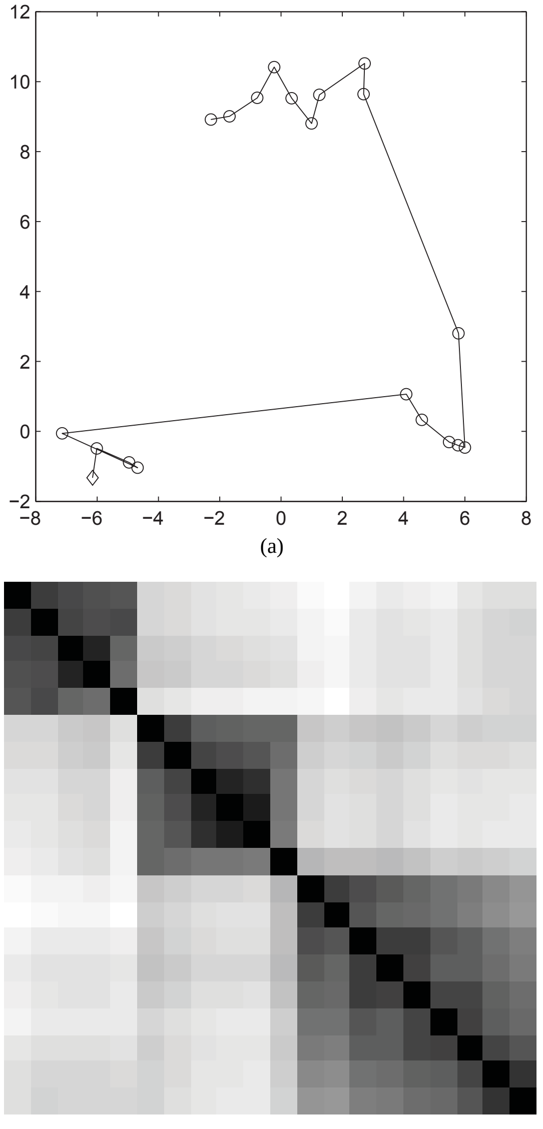
Figure 1. Scatterplot of the data set *X*.



(a)

(b)

Figure 2. (a) The original order of *X*; (b) The corresponding dissimilarity image.



(a)

(b)

Figure 3. (a) The new order of *X*; (b) The corresponding dissimilarity image shows three clusters.

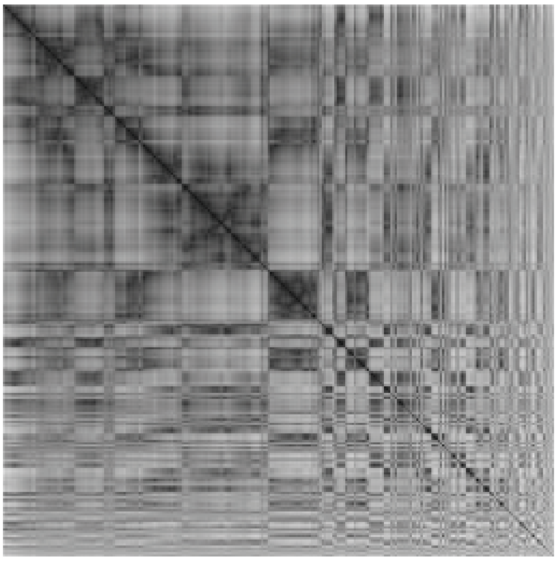


Figure 4. How many clusters can be seen in this ODI?

will result in what we call the *tendency curves*. The borders of clusters in the ODM (or blocks in the ODI) are reflected as certain patterns in peaks and valleys on the tendency curves. To be exact, we will actually use only one of these curves, called the *d*-curve, which is the difference of two other curves. The patterns on the *d*-curve can be caught not only by human eyes but also by the computer. It seems that the computer is more sensitive to these patterns than human eyes are to them, or to the gray level patterns in the ODI. For example, the computer caught three clusters in the data set that produced the virtually useless ODI in **Figure 4**.

Remark: *The patterns on the tendency curves only roughly match the block borders in ODI in positions, and the sizes of these blocks do not closely approximate the sizes of clusters in the data, either. This is because the VAT algorithms tend to index each cluster's most outlying points at the very end, after all the more dense cluster cores are indexed. Whenever we say in this paper* "*catch clusters/blocks*", *we mean the program reveals the existence of clusters/blocks. The sizes and members* (*or memberships*) *will have to be found by a clustering method, not by a tendency algorithm such as ours.*

We will describe our method in detail in §2 below, give numerical examples in §3, and conclude the paper with discussions and future plans in the last section.

2. Visual Assessment of Cluster Tendency Using Diagonal Tracing

We try to catch possible diagonal blocks in the ordered dissimilarity matrix *R*, the numeric output of VAT. We do so by using various averages of dissimilarities, which are stored as vectors and displayed as curves. The goal is to catch the borders of black blocks in an ODI such as **Figure 3(b)**. Imagine that a horizontal line segment running from the left edge of the ODI to the diagonal (exclusive) moves down. The line segment is dark when it is inside the first block, and becomes light once it gets out of the block. If the clusters are well separated, the change in the darkness should be large enough to catch. We use the *row-average*, which is the average of the elements to the left of the diagonal in a row of the ODM, to represent (the darkness of) the line segment. We call its graph (versus the row number) the *r*-curve. The darker the line segment, the smaller the row-average, thus the lower the *r*-curve. When line segment goes across a border, the *r*-curve should first show a peak because the numbers to the left of the diagonal element R_{ii} will suddenly increase. It should drop back down rather quickly when the line moves well inside the next black block. Therefore a border of two blocks should induce a peak on the curve. There is a complication, though, that we can not keep using all the elements from the very left edge to the diagonal. This is because, from the second block down, the beginning part of the line, which is rather light, would drag down its average darkness, and decrease the change in its darkness when the line goes across another possible border. In terms of the graph, the *r*-curve would become flatter and flatter, its peaks lower and lower when moving to the right, thus harder and harder for the program to catch. Our way to solve this problem is to restrict ourselves in a subdiagonal band with a width *w*, called the *w*-band, as shown in **Figure 5**. That is, we cap by *w* the number of elements in the average. To be exact, we define the *i*-th element of the *r*-curve (the *i*-th row-average) as

$$r_1 = 0, r_i = \frac{1}{i - \ell_i} \sum_{j=\ell_i}^{i-1} R_{ij}, \qquad (2)$$

where $\ell_i = \max(1, i - w)$, $2 \le i \le n$. This is the average of the elements of row *i* in the *w*-band shown in **Figure 5** below.

When the situation is less than ideal, there will be noise, sometimes very "loud" noise, on the *r*-curve, which may destroy possible patterns on it. To overcome this, we extend the idea of averaging to more rows, which leads to the *m*-curve, whose *i*-the element is the average of all elements R_{kl} such that

$$\max(1, i - m + 1) \le k \le i$$

and

$$\max(1, i - w) \le l \le i - 1.$$

These are the elements in up to *m* rows above row *i*, inclusive, that fall in the *w*-band, corresponding to the region between the two horizontal line segments in **Figure 5**.

The *m*-curve often reveals the pattern beneath the noisy *r*-curve. Since the ODM is scaled so that $0 \le R_{ij} \le 1$, $\forall i$ and j, the heights of peaks on the *m*-curve remain roughly the same from case to case, that is, when clusters are well formed.

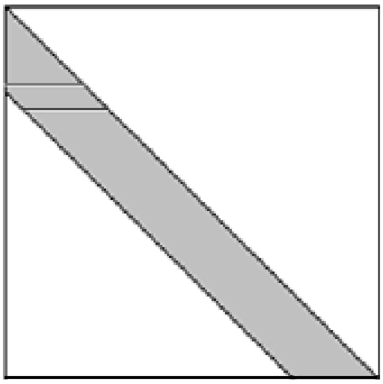


Figure 5. Sub-diagonal band of the ODM.

But again there are less-than-ideal situations, in which there are outliers. The VAT algorithms tend to order outliers near the end, so the *m*-curve tends to move up toward the right, which is fine to human eyes but makes it hard for the program to identify peaks and valleys using thresholds. This is why we introduce the *M*-row moving average, called the *M*-curve. The *M*-curve is defined in the same way as the *m*-curve except with *m* replaced by *M*. The *M*-curve shows long term trends of the *r*-curve. We are, however, NOT interested in the *M*-curve itself. We use the *M*-curve to "correct", or to level up, the *m*-curve, by subtracting the former from the latter. It is the difference of the *m*- and *M*-curves, which we call the *d*-curve, that we are interested in. The *d*-curve retains the shape of the *m*-curve but is more horizontal, basically lying on the horizontal axis. Furthermore, the *M*-curve changes more slowly than the *m*-curve, thus when moving from one block into another block in the ODM, it will tend to be lower than the *m*-curve. As the result, the *d*-curve will show a valley, most likely below the horizontal axis, after a peak. It is the peak-valley, or high-low, patterns on the *d*-curve that signal the existence of cluster structures. This will become clear in our examples in the section that follows.

Although the *d*-curve is the only curve we really need, we will also show other tendency curves, that is, the *r*-, *m*- and *M*-curves, in the first few examples to show the reader how the idea evolved from an intuitive *r*-curve to the final, rather technical, *d*-curve.

Remark: *It may seem much more natural to define the i-th element of the r-curve as the average of all* R_{ij} *such that the object* o_j *is in the same cluster as* o_i *and* $j < i$. *Actually this is what we tried at the very beginning of this work. More precisely, we set* $\ell_i = 0$ *in definition* (2) *at the beginning of the calculation, and once we believed we had found a new cluster, we reset* ℓ_i *to the index of the element we believed to be the first one in the new cluster. There were several problems. First, neither the VAT algorithms nor our program can accurately locate the borders of clusters in terms of the index values. Second, any possible patterns obtained that way were self-fulfilled: once we reset* ℓ_i, *all curves went back to zero, and then it would look like there was indeed a new cluster. It would literally tear the tendency curves apart, and distort all possible high-low patterns.*

3. Numerical Examples

In all the examples in this paper, we will use the values

$$m = \text{ceil}(0.05n), \; M = 5m, w = 3m, \qquad (3)$$

where *n* is the number of objects in the data set. Here the ceiling function is used for *m* so that it is at least 1 even if *n* is very small. And these are the values we recommend to possible users of our algorithm when there is no clear reason to change them. Discussion on how the values of these, and two other, parameters were chosen can be found later in the section.

We first give one group of examples in $\mathbb{R}^2$ so that we can use their scatterplots to show how well/poorly the clusters are separated. We also give the visual outputs (ODIs) of VAT for comparison. These sets are generated by choosing $\alpha =$ 8, 4, 3, 2, 1 and 0 in the following settings: 2000 points are generated in three groups from multivariate normal distribution having mean vectors $\mu_1 = \left(0, \alpha\sqrt{6}/2\right)$, $\mu_2 = \left(-\alpha\sqrt{2}/2, 0\right)$ and $\mu_3 = \left(\alpha\sqrt{2}/2, 0\right)$. The probabilities for a point to fall into each of the three groups are 0.35, 0.4 and 0.25, respectively. The covariance matrices for all three groups are I_2. Note that μ_1, μ_2 and μ_3 form an equilateral triangle of side length $\alpha\sqrt{2}$.

The pictures for $\alpha = 8$ (**Figure 6**) show what we should look for on the curves. The clusters are very well sepa- rated, and the ODI has three black blocks on the diagonal with sharp borders. Our *r*-curve (the one with "noise") has two vertical rises and the *m*-curve (the solid curve going through the *r*-curve where it is relatively flat) has two peaks, corresponding to the two block borders in the ODI. The *M*-curve, the smoother, dash-dotted curve, is only interesting in its relative position with respect to the *m*-curve. That is, it is only useful in generating the *d*-curve, the difference of these two curves. The *d*-curve looks almost identical to the *m*-curve, also having two peaks and two valleys. The major difference is that it is in the lower part of the figure, around the horizontal axis.

Figure 7 shows the case $\alpha = 4.$ The clusters are less separated than the case $\alpha = 8$ and the slopes of the tendency curves are smaller. There are still two vertical rises on the *r*-curve, and two peaks followed by two valleys on all other curves where the block borders are in the ODI in part (b). What is really different here from the case $\alpha = 8$ is the wild oscillations near the end of the *r*-curve, bringing up all other three curves. This corresponds to the small region in the lower-right corner of the ODI, where there lacks pattern. Note that no valley follows from the third rise or peak. This is understandable because a valley appears only when the curve index (the horizontal variable of the graphs) runs into a cluster, shown as a block in ODI.

Now we know what we should look for: peaks followed by valleys, or high-low patterns, on the *r* and *d*-curves. Later on we will show that even the *r*-curve is not good enough and only the *d*-curve will do the job.

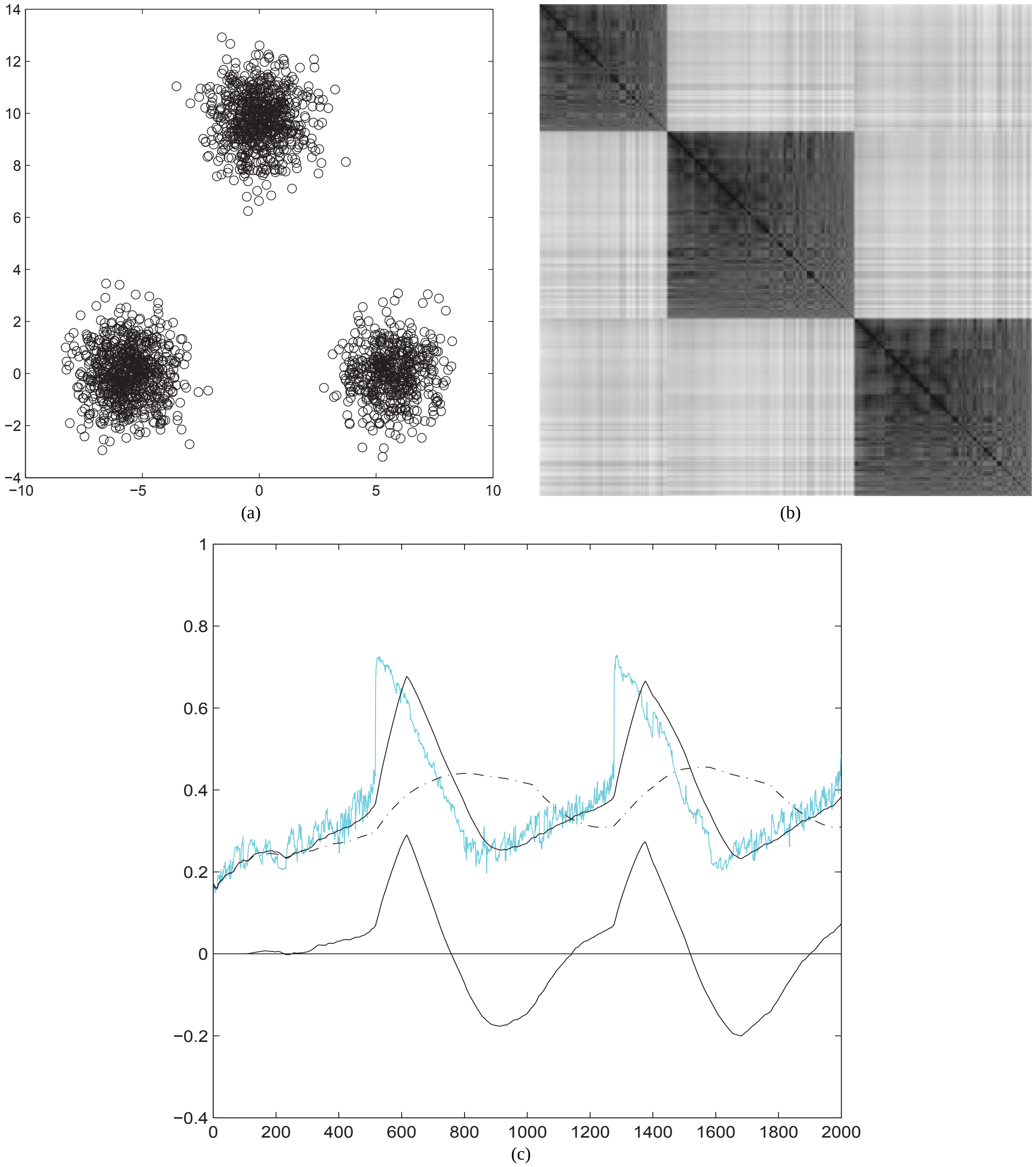
The case $\alpha = 3$ is given in **Figure 8**. One can still easily make out three clusters in the scatterplot, but it is hard to say to which cluster many points in the middle belong. It is expected that every visual method will have difficulties with them, as evidenced by the lower right corner of the ODI, and the oscillations on the last one fifth of the *r*-curve. The oscillations bring up the *m*- and *M*-curves, but not the *d*-curve. The *d*-curve remains almost the same as those in the two previous cases, except the third peak becomes larger and decreases moderately near the end, without forming a valley. The two high-low patterns on the *m*- and *d*-curves show the existence of three clusters. As we have said earlier that it is a valley on the *m*-curve and, especially, the *d*-curve that signals the beginning of a new cluster.

Note that the *m*-curve goes up with wild oscillations so much in **Figure 8(c)** that its right end rises higher than any peak on it, which makes it hard for the computer to catch the high-low patterns, at least hard with thresholds. That is why we introduced the more technical *d*-curve to replace the intuitive *m*-curve, which had earlier replaced the more intuitive but often noisy *r*-curve. The *d*-curve remains mostly level, close to the horizontal axis, thanks to the compensation it gets from the *M*-curve. Also, unlike the other three curves, its values never get too high or too low, which enables us to catch the high-lows easily. As a consequence, our VATdt algorithm only uses the *d*-curve to access the tendency. And *we will only display the d-curve in the remaining part of the paper* to show the reader a cleaner view, although we ourselves often feel the *r*- and *m*-curves visually informative.

We use two thresholds to detect high-lows. When the *d*-curve hits a ceiling, set as 0.04, and then a floor, set as 0, the program reports one new cluster. These ceiling and floor values are satisfied by all cases in our numerical experiments, even those not reported here, where the clusters are reasonably, sometimes only barely, separated.

If we lower the ceiling and raise the floor, we would be able to catch some of the blended clusters we know we have missed, but it would also increase the chance of "catching" false clusters. We are not saying these values are the best. Any values are arguable, as arguable as the number of clusters is when the clusters are blended. We do not like the idea of tuning parameters to particular examples, and will stick to the same ceiling and floor values throughout this paper. In fact, we will stick to the same set of values for all parameters in our program, that is, the values for the ceiling and floor set here, and those for *m*, *M* and *w* given in (3).

The situation in the case $\alpha = 2,$ shown in **Figure 9**, really deteriorates. One can barely make out the three clusters in part (a) that are supposed to be there; the ODI in part (b) is a mess. In fact, this is the same ODI as the one in **Figure 4**, put here again for side-by-side comparison with the scatterplot and the *d*-curve. The *d*-curve,



(a)

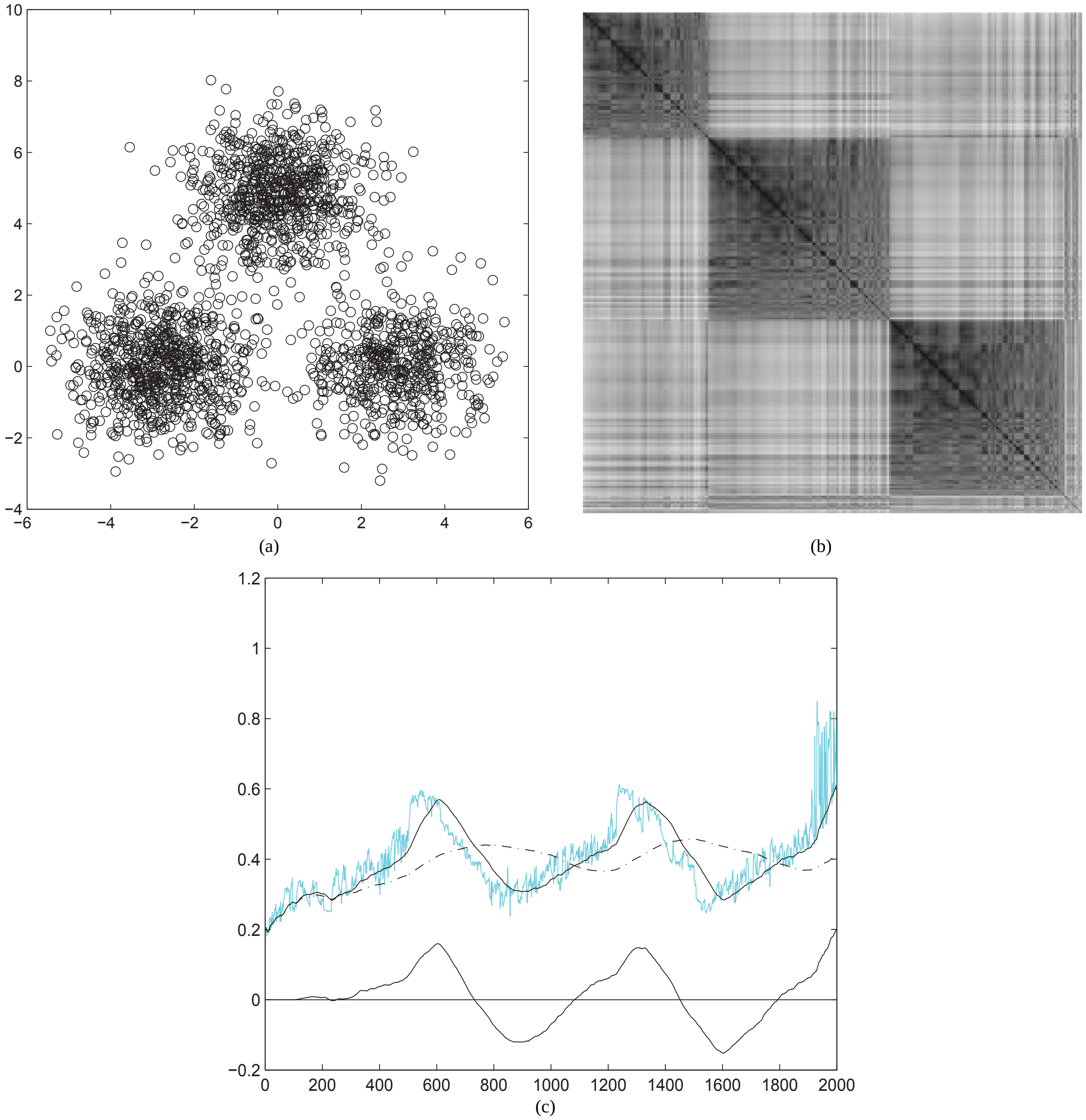
(b)

(c)

Figure 6. Three normally distributed clusters in $\mathbb{R}^2$ with $\alpha = 8$. (a) Scatterplot; (b) ODI from VAT; (c) Tendency curves.

however, picks up cluster structure from the ODM. It has several high-lows, with two of them large enough to hit both the ceiling and floor, whose peaks are near 600 and 1000 marks on the horizontal axis, respectively. This example shows that our tendency curves are more sensitive than the raw block structure in the 2D display ODI. The largest advantage of the tendency curves is probably the quantization of gray level patterns which enables the computer, not only human eyes, to catch possible patterns.

One may question how many clusters this data set truly has, but it then depends on what one means by "truly". This may be subjective. We see three clusters in **Figure 9(a)**; if one sees only one cluster there, one may want to tune down the sensitivity of the program by raising its ceiling value and lowering its floor value. We are



(a)

(b)

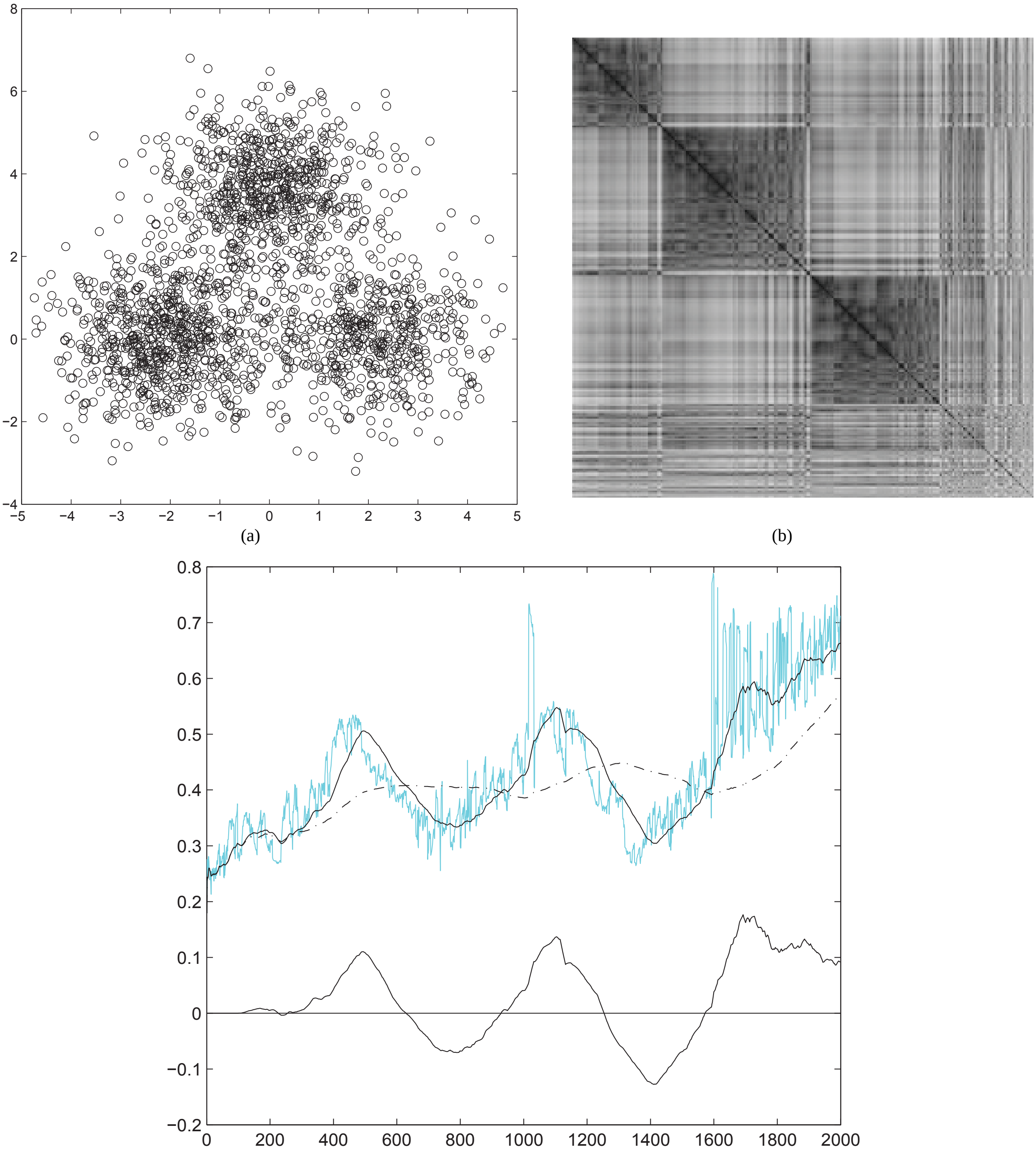
(c)

Figure 7. Three normally distributed clusters in $\mathbb{R}^2$ with $\alpha = 4$. (a) Scatterplot; (b) ODI from VAT; (c) Tendency curves.

only saying that our program can be sensitive enough to "see" three clusters in this case.

When α goes down to zero, the cluster structure disappears. The scatterplots for $\alpha = 0$ (**Figure 10(a)**) and $\alpha = 1$ (not shown) are almost identical, showing a single cluster in the center. The *d*-curves for both cases (**Figures 10(b)** and **(c)**) have no high-lows large enough to hit the ceiling then the floor, which is the way they should be.

We now show that, without modifying any parameter values, our VATdt algorithm works on small data sets, too. The data sets in this group of examples are similar to those in **Figures 6-10** of Bezdek and Hathaway [5]. These are data sets in $\mathbb{R}^4$, generated in the same way as the examples in the first group. A total number of 120 observations were generated in four groups from multivariate normal distribution having mean vectors αe_i, $i = 1, \cdots, 4$, where e_i's are the unit axis vectors. A point has equal opportunity to fall into each of the four groups, that is, the probability is 0.25 for every group. The covariance matrices are all I_4. Note the distances between cluster centers are still $\alpha\sqrt{2}$, same as the previous examples. The appearances of the ODIs and the *d*-curves



(a)

(b)

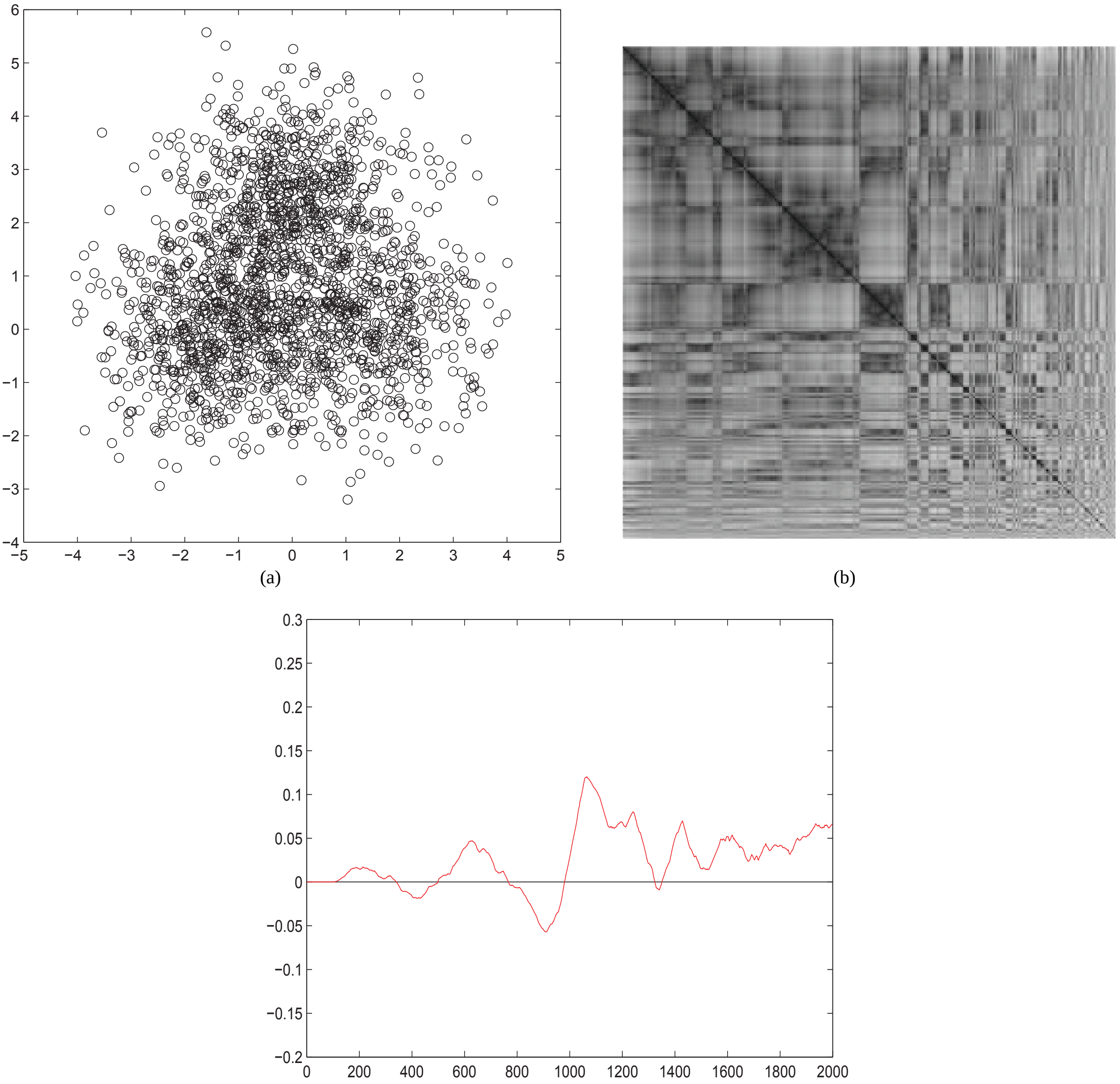
(c)

Figure 8. Three normally distributed clusters in $\mathbb{R}^2$ with $\alpha = 3$. (a) Scatterplot; (b) ODI from VAT; (c) Tendency curves.

are similar to those in $\mathbb{R}^2$, and so is the way they deteriorate as the value of α decreases. The *d*-curves for $\alpha = 8$, 4 and 3 are given in **Figure 11**. There are three clear high-lows on each of them, revealing the existence of four clusters, which we think is appropriate since the α values are relatively large.

Figure 12 gives another example in which the ODI from VAT (in part (a)) fails to show any useful visual information on the structure, but our program identified three high-lows (between the 40 and 80 marks), or four clusters, from the *d*-curve.

Remark: *We remind the reader that the positions of the peaks and valleys do not reflect the sizes of the clusters closely unless the clusters are very well separated.*



(a)

(b)

(c)

Figure 9. Three normally distributed clusters in $\mathbb{R}^2$ with $\alpha = 2$. (a) Scatterplot; (b) ODI from VAT; (c) *d*-curve.

Does our method always say what it should say? Well, there is not, and there will never be, an infallible method to determine the number of clusters. In many cases, there are no right or wrong answers; it all depends on what one means by "should". The data set used in **Figure 13** was generated the same way as that in **Figures 10(a)** and **(b)** where $\alpha = 0$, except that it contains only 100 observations. So there "should" be only a single cluster. But both the ODI from VAT and our *d*-curve show some structure, and our program "caught" three clusters. If one compares the scatterplots in **Figures 10** and **13**, one can find that there is a single well-shaped cluster in **Figure 10** while there are only scattered points in **Figure 13**. If the number of points is small, there is a difference between what a random generator is intended to generate and what it actually generates.

We now give two examples where the points are regularly arranged, on a rectangular grid, and along a pair of concentric circles, respectively. These are similar to the data sets in **Figures 12** and **13** of Bezdek and Hathaway [5]. These examples show that the *d*-curve works better on these rows and rings than the ODI, which does not contain black blocks anymore. In **Figure 14**, 32 points are equally arranged on each of the 8 lines, with the distance between two consecutive points on the same line equal to 0.05, and that between two lines 0.4. The ODI in

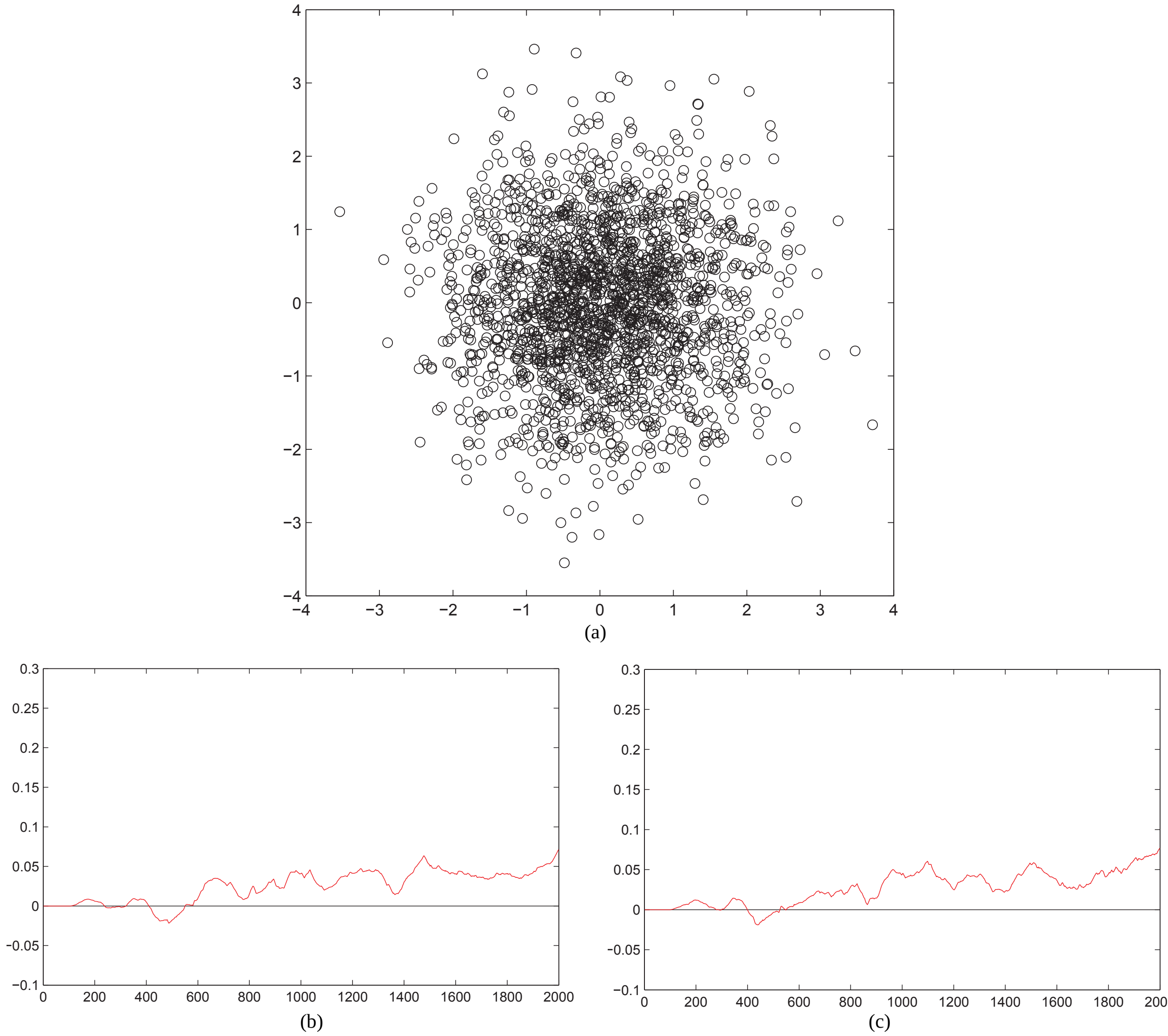


Figure 10. Three normally distributed clusters in $\mathbb{R}^2$ merged into a single one. (a) Scatterplot for $\alpha = 0$; (b) *d*-curve for $\alpha = 0$; (c) *d*-curve for $\alpha = 1$.

part (b) has a periodic nature, but no blocks. Bezdek and Hathaway [5] conclude from the ODI generated from a similar data set that “it is reasonable to conjecture that the underlying data fall into 8 very regular clusters”. The *d*-curve shown in part (c) is almost sinusoidal, with the highs and lows far beyond the ceiling and floor, strongly indicating the existence of 8 clusters (which are rows).

Remark: *Our program works on the original example in* [5] *just as fine. We made the changes here so that it looks more like* 8 *clusters instead of a single rectangular cluster.*

In **Figure 15**, 64 points are equally distributed on a circle of radius 0.45 centered at (0.5, 0.5), and another 64 on a concentric circle of radius 0.25. This ODI does not contain black blocks either, only a block *form*; see **Figure 15(b)**. Based on this 2×2 block form Bezdek and Hathaway [5] infer that the data consist of two similar, regular structures. Our *d*-curve in **Figure 15(c)** has a single high-low pattern on it, and the program reported the existence of two clusters.

It is almost a sacred ritual that everybody tries the Iris data in a paper on clustering, so we also tried our program on it. It is well-known that the data consist of values of four features of each of 150 irises (150 points in a four-dimensional feature space). These irises are of three different physical types, 50 from each type, thus the data have three physically labeled classes. But two of the three flower types yield data points that largely overlap in this particular feature space, so many argue that the unlabeled data are naturally clustered into two geometrically well-defined clusters; see [5]. The *d*-curve our program produced is given in **Figure 16(b)**, and the com-

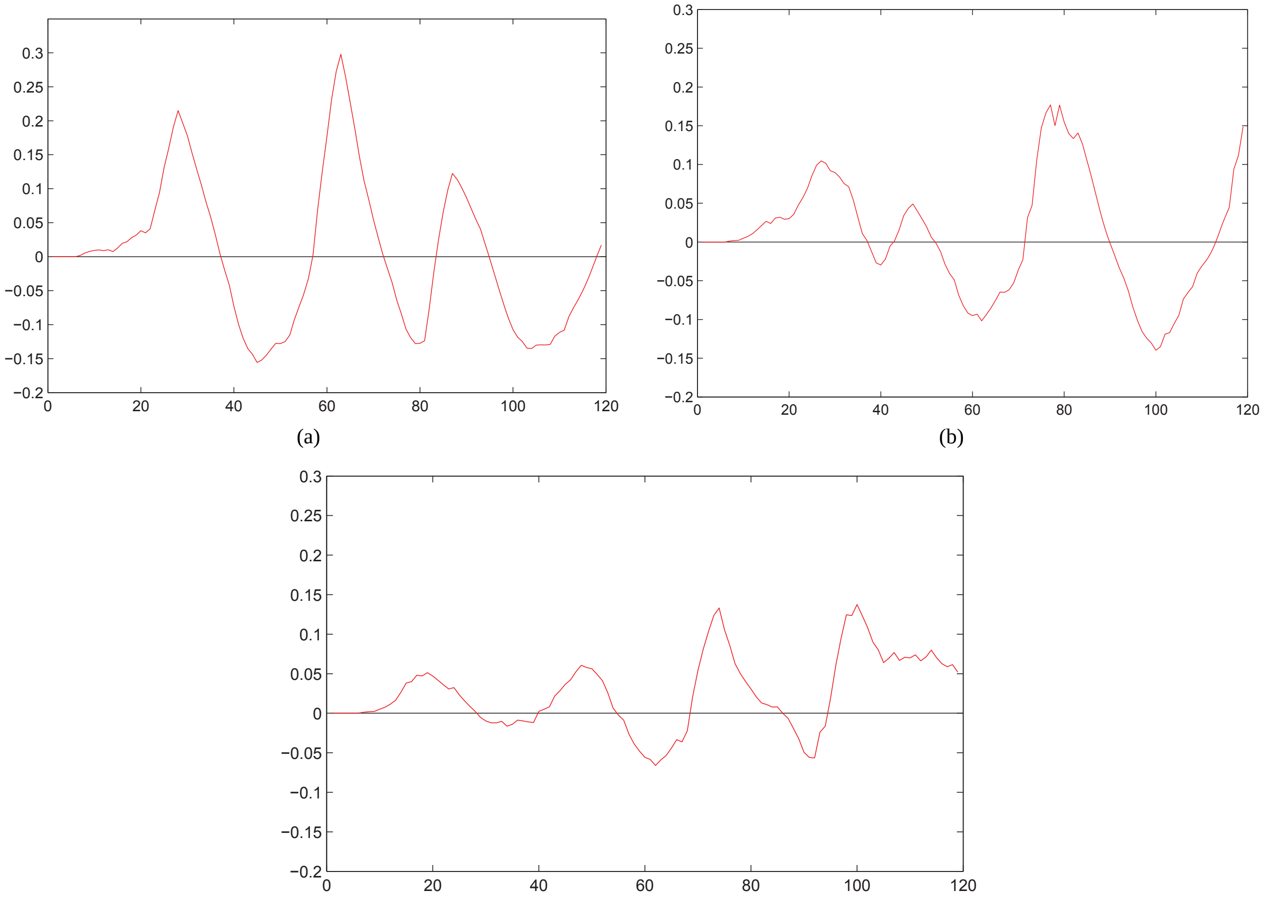


Figure 11. The d-curves for four normally distributed clusters in $\mathbb{R}^4$ (a) $\alpha = 8$; (b) $\alpha = 4$; (b) $\alpha = 3$.

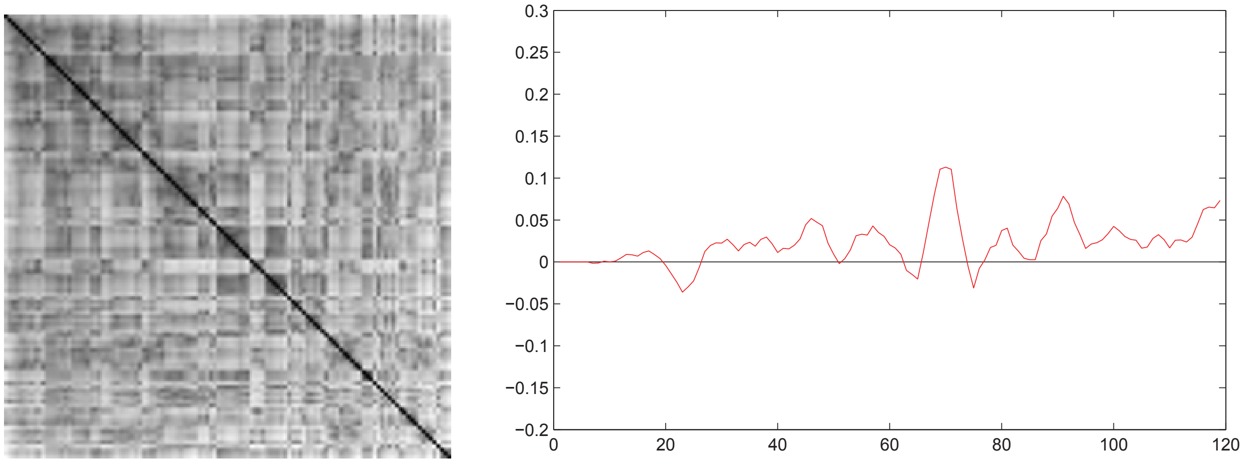
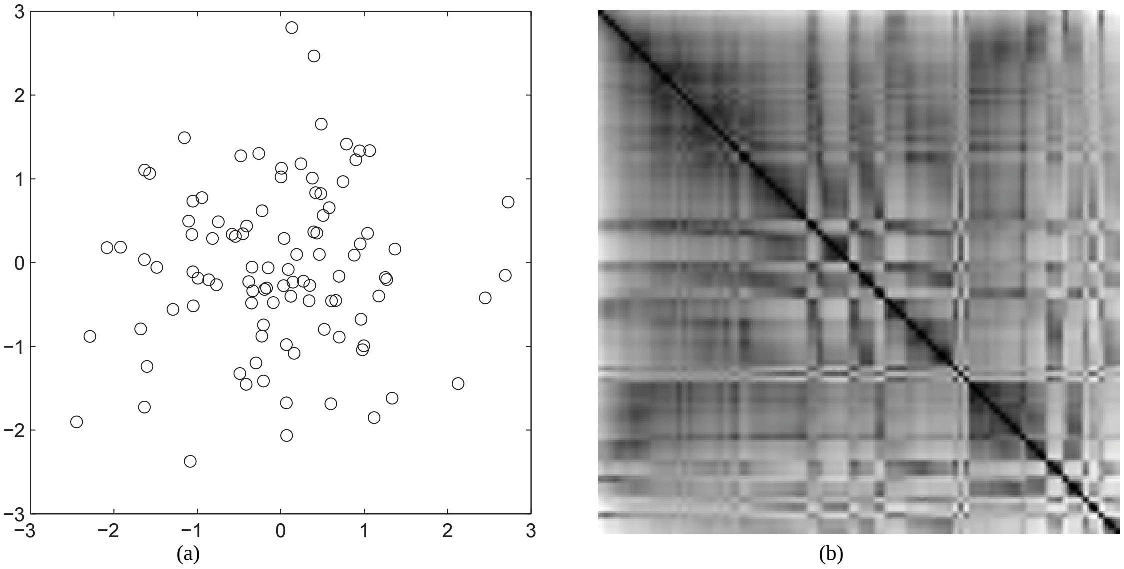


Figure 12. Four normally distributed clusters in $\mathbb{R}^4$ for $\alpha = 2$. (a) ODI from VAT; (b) *d*-curve.

puter caught the large high-low on the left and ignored the small one on the right, and reported the existence of two clusters. Once again one may argue on the correctness of the program ignoring the smaller high-low (thus the choice of the ceiling value), just as one can argue on the "correct" number of clusters in the Iris data.

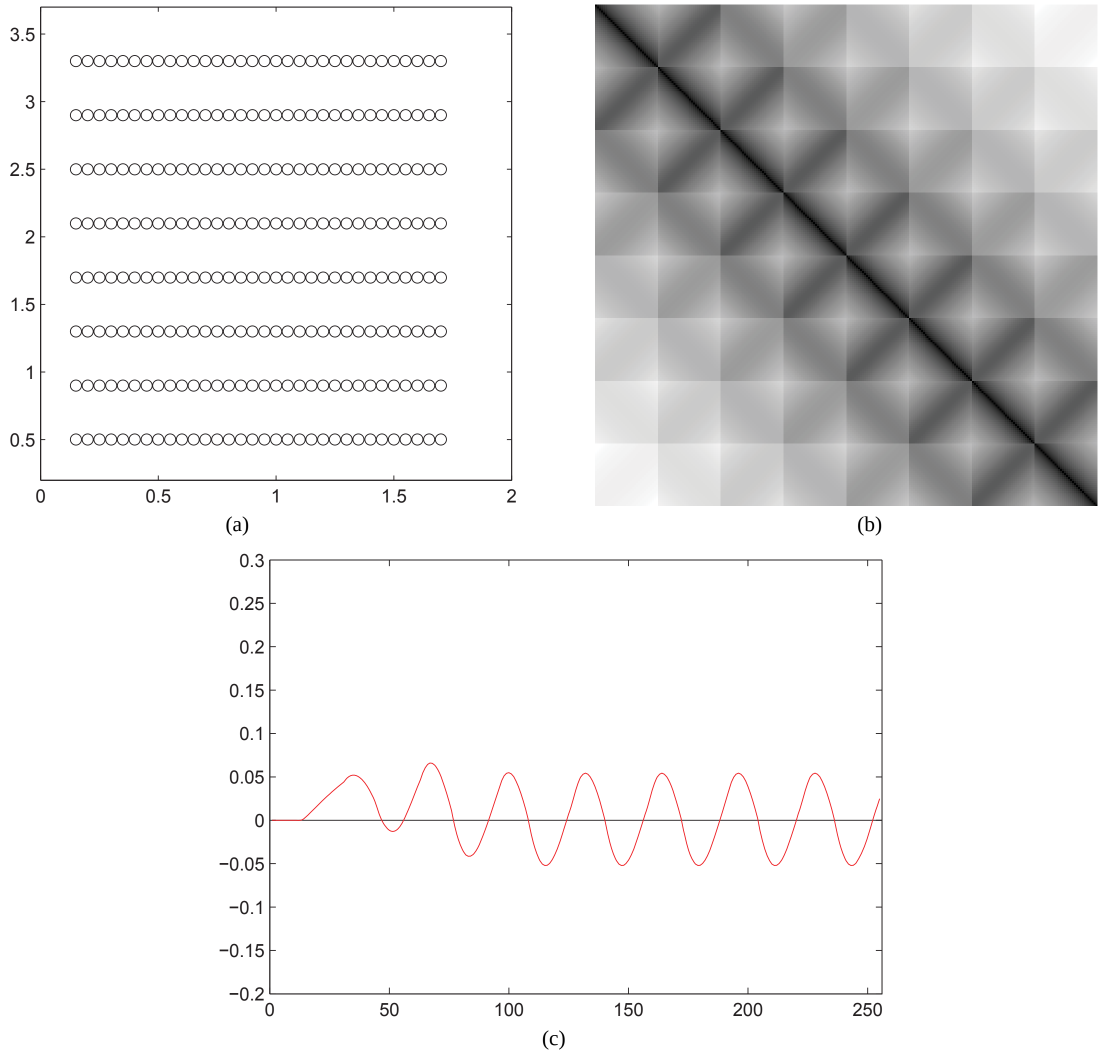
We conclude this section with some comments on the choice of the parameter values of the program. Since enough has been said about the floor and ceiling values, here we only discuss the values of *m*, *M* and *w*. We



(a)

(b)

Figure 13. One hundred normally distributed points in $\mathbb{R}^2$. (a) Scatterplot; (b) ODI from VAT; (c) *d*-curve.

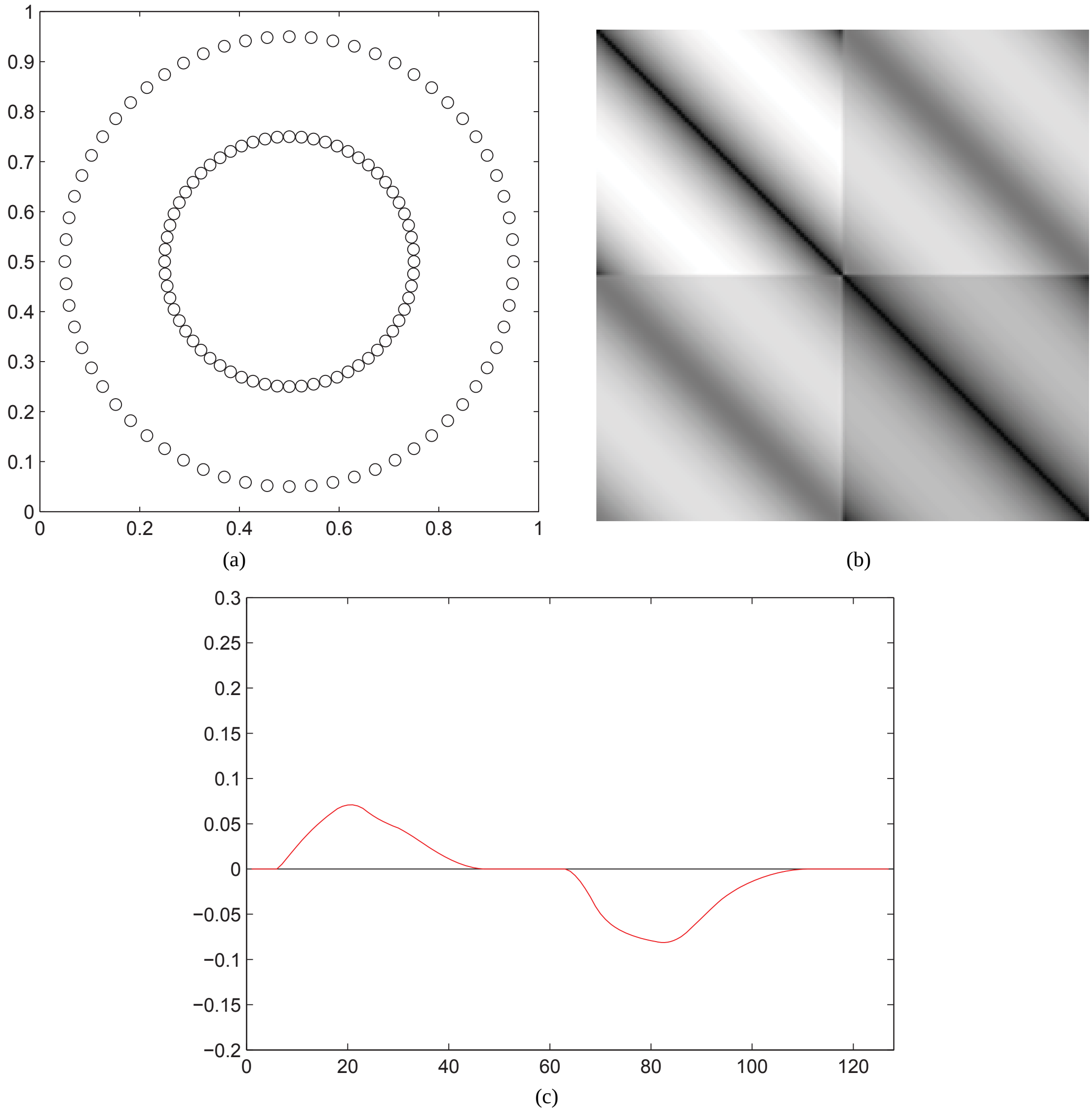


(a)

(b)

(c)

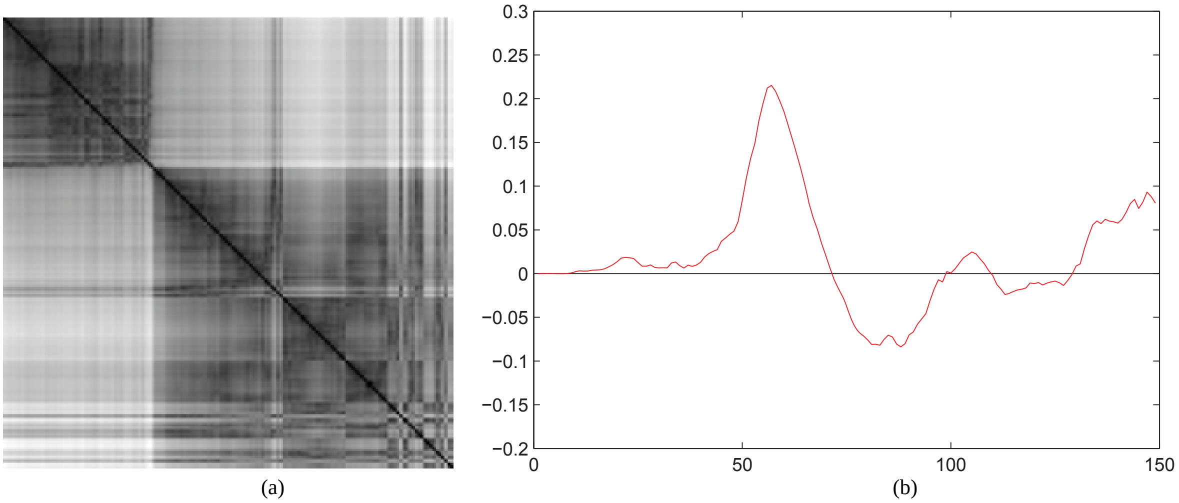
Figure 14. Points regularly distributed along parallel lines (a) Scatterplot; (b) ODI from VAT; (c) *d*-curve.



(a) (b)

(c)

Figure 15. Points regularly distributed along a pair of concentric circles (a) Scatterplot; (b) ODI from VAT; (c) *d*-curve.



(a) (b)

Figure 16. (a) ODI for the Iris data from VAT; (b) *d*-curve.

ended up with the values in (3) from experiments. First, we want as small a value for m as possible so that relatively small clusters will not get lost in the averaging process. But if it gets too small, the m-curve would get noisier and noisier and eventually fall back to the r-curve. We also want it as a percentage of n so that we do not have to change it to suit data sets of different sizes. Five percent is the smallest we dare go, (n often gets below 100, and then we are only looking at the average of a few rows), and it works very well. The performance of the program is not sensitive at all to the changes in M. As long as it is several times larger than m, we did not see much difference. The value of w makes a difference only occasionally, and, when it does, only marginally. We tried values from $w = 2n$ to $5n$, and all of them worked fine. All in all, the value $w = 3m$ worked best, but the difference was insignificant. Thus we decided that a single set of parameter values could successfully be used for all cases, which is a rare situation for clustering procedures involving user-selected parameter values.

One scenario in which we foresee the need of changing parameters is when the ratios of the cluster sizes in a data set are so large that (relatively) small clusters get lost in the averaging, causing the d-curve valleys to be too shallow to hit the floor. One will then need to decrease the values of m and w, which may help form larger high-low patterns on the d-curve. We would feel comfortable adjusting the values of the ceiling and floor if there are "clean" high-low patterns on the d-curve, that is, if there are not many zigzags when the curve goes up and down. When changing parameter values, we recommend the user to look at the r-curve, too. One should feel more confident if the r-curve does not show too much noise.

4. Conclusions

Our VATdt algorithm is meant to replace the straightforward visual displaying part of the VAT algorithms mentioned in the second paragraph of §1. Or, for that matter, it can start from an ordered dissimilarity matrix from any algorithm of that kind. Instead of displaying the matrix as a 2-dimensional gray-level image ODI for human interpretation, VATdt analyzes the matrix by taking averages of various kinds along its diagonal and produces the tendency curves, with the most useful of them being the d-curve. This changes 2D data (a matrix) into a 1D array, which is certainly easier to both human eyes and the computer since the concentration is now only on one variable—the height.

Possible cluster structure is reflected as high-low patterns on the d-curve with a relatively uniform range that enables the computer to catch them with thresholds. The values of thresholds may be arguable, but no more so than the "right" number of clusters that exist in a given data set. For example, some see only one single cluster in **Figure 9(a)** while we see three. Our experiments show that the computer is more sensitive to high-low patterns on the d-curve than human eyes to patterns in 2D gray-level images.

We are truly encouraged by the two examples in **Figures 14** and **15** where the ODI images do not have blocks but the d-curve still did the job nicely. An ODI shows blocks only if the data set contains (elliptical) disk-shaped clusters in 2-dimensional feature space, or ellipsoid- or ball-shaped clusters in feature spaces of higher dimensions. Clusters of other shapes show different patterns in the ODI, whose meaning one can only guess. Our d-curve, however, clearly shows the cluster structures in both cases, where chain-shaped clusters exist.

We plan to further investigate and improve the VATdt algorithm, experimenting with it on clusters of different shapes, even with mixed shapes in the same data set. It also interests us to use different metrics, even dissimilarities that are not metrics. We are mainly interested in cases where structures in the ODM exist but the ODI does not show them clearly, at least not in black blocks.

5. Acknowledgements

The author would like to thank Professor Richard J. Hathaway for his numerous helpful thoughts and suggestions, including the final title of this paper.

REFERENCES

[1] A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data," Prentice-Hall, Englewood Cliffs, 1988.

[2] B. S. Everitt, "Graphical Techniques for Multivariate Data," Elsevier, New York, 1978.

[3] J. W. Tukey, "Exploratory Data Analysis," Addison-Wesley, Reading, 1977.

[4] W. S. Cleveland, "Visualizing Data," Hobart Press, Summit, 1993.

[5] J. C. Bezdek and R. J. Hathaway, "VAT: A Tool for Visual Assessment of (Cluster) Tendency," *Proceedings of the* 2002 *International Joint Conference on Neural Networks*, Honolulu, 12-17 May 2002, pp. 2225-2230.

[6] J. C. Bezdek, R. J. Hathaway and J. M. Huband, "Visual Assessment of Clustering Tendency for Rectangular Dissimilarity Matrices," *IEEE Transactions on Fuzzy Systems*, Vol. 15, No. 5, 2007, pp. 890-903. doi:10.1109/TFUZZ.2006.889956

[7] R. J. Hathaway, J. C. Bezdek and J. M. Huband, "Scalable Visual Assessment of Cluster Tendency for Large Data Sets," *Pattern Recognition*, Vol. 39, No. 7, 2006, pp. 1315-1324. doi:10.1016/j.patcog.2006.02.011

[8] J. M. Huband, J. C. Bezdek and R. J. Hathaway, "Revised Visual Assessment of (Cluster) Tendency (reVAT)," *Proceedings of the North American Fuzzy Information Processing Society* (*NAFIPS*), Banff, 27-30 June 2004, pp.

101-104.

[9] J. M. Huband, J. C. Bezdek and R. J. Hathaway, "Big-VAT: Visual Assessment of Cluster Tendency for Large Data Set," *Pattern Recognition*, Vol. 38, No. 11, 2005, pp. 1875-1886. doi:10.1016/j.patcog.2005.03.018

[10] I. Borg and J. Lingoes, "Multidimensional Similarity Structure Analysis," Springer-Verlag, New York, 1987. doi:10.1007/978-1-4612-4768-5

[11] M. Kendall and J. D. Gibbons, "Rank Correlation Methods," Oxford University Press, New York, 1990.