CLUSTERING ANALYSIS AS A GESTALT PROBLEM

Shelia Guberman and Wita Wojtkowski

1. Introduction

Cluster analysis is, at present, one of the most popular methods used for processing data that represent set of objects (ANDENBERG 1973; HARTIGAN 1975; MURTAGH and HECK 1987). In very general terms, the purpose of cluster analysis is to group data according to the principle of similarity.

In the context of this paper the notion of similarity embodies a measure in n-dimensional space (a distance between points). This notion is nontrivial, because the difficulty surfaces when one has to choose weights for the attributes representing an object. For example, how to weight between the length of an object and its electrical conductivity? However, for the points of a cluster on a plane this is not a difficult problem to resolve: grouping of points by similarities is one of the traditional themes extensively investigated by the Gestalt psychologists (ANDENBERG 1973; HARTIGAN 1975; MURTAGH and HECK 1987; TOUSSAINT 1980; MATULA and SOKAL 1980). When objects are points on a plane, our psychological perception of these objects seems to agree with the Euclidean measure (where all attributes have the same weight). We note thus, that in the case of clustering on a plane, the object (a visual object drawn on the sheet of paper) and its representation (the point on a plane with a given system of Cartesian coordinates) coincide. In the multidimensional space, the point represents the object under study, and the coordinates are its attributes.

For simple cluster shapes, and for relatively uniform cluster densities, as well as for clusters that are widely separated (as compared to the average distance between points in a given cluster) clustering algorithms provide good results *if the number of clusters is known a priori.* We note that the precise number of clusters is known when the nature of the objects (that is the domain to which the objects belong) *and* the nature of the clusters are known. Clustering problem for geological data may serve as an example. For geological data, the data from a specific geological region may identify lithology of different layers, and may include, for example, limestone, sandstone, and carbonates. Therefore, if we are confronted with a set of objects, and we know to which clusters they might possibly belong, for specific clusters we may identify at least some of the representative objects.

Thus, if we know the number of clusters and their representative objects, clusters become classes - now we have a pattern recognition problem.

2. The Clustering Problem

We note that when the number of clusters is a part of the clustering algorithm, in reality this number is only a guess. Consequently we posit that the precise clustering problem is this: *distinguish between the disparate clusters when the number of clusters is not known a priori.* The work of OSBOURN and MARTINEZ (1995) addresses exactly this issue. A key contribution of this study is the introduction of a visually-empirical region of influence (VERI) shape which can correctly treat a wide variety of clustering problems. The notion of the VERI shape is based on the psychophysical studies which have attempted to quantify human visual judgments of clusters (TOUSSAINT 1980a). The OSBOURN and MARTINEZ method computes clustering based on the local k-dimensional neighbors of each point, and thus handles arbitrary number of clusters and the arbitrary global cluster shapes. The same VERI shape can be used for a variety of examples of clusters. It is important to note, however, that the OSBOURN and MARTINEZ method requires the input of k-dimensional data.

The clustering problem, at least for the 2-dimensional case, is a Gestalt problem: from the psychological point of view the problem requires a holistic approach. What this means is as follows: the decision about each point in a cluster depends (in general) on the distribution of all points in the image of the cluster. We propose to generalize such an approach to *n dimensions*.

3. The DD Algorithm and the Clustering Problem

In a previous article (GUBERMAN 1983) we describe in detail an algorithm (referred to as the DD algorithm) that can be put to use when performing automated image processing, and which employs the Gestalt approach. We propose to apply this algorithm to the cluster analysis problem. In Appendix we offer brief description of the DD algorithm.

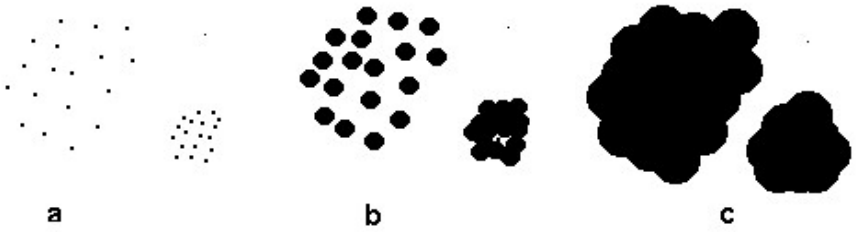


Figure 1. Applying DD algorithm to the set of points on a plane

The key aspect of the DD algorithm is the generation of the function n(r). In brief, the algorithm works as follows: consider any given set of points on a plane (for example, those shown in Figure 1a). Let each point be the center of a circle of a radius **r**. We define $F_r(x,y)$ to be $F_r(x,y)=1$ within each circle of radius **r**, and $F_r(x,y)=0$ outside. In Figure 1 the areas where $F_r(x,y)=1$ are shaded black. If **r** is sufficiently small (Figure 1a), the number of connected areas **n**, where $F_r(x,y)=1$ is equal to N, where N is the number of points on the image. As **r** increases, some circles begin to intersect, and the number of separate regions for which $F_r(x,y)=1$ decreases (Figure 1b). When circles become bigger, the number of these separate regions continues to decrease (Figure 1c). The graph of function n(r) is shown in Figure 2.

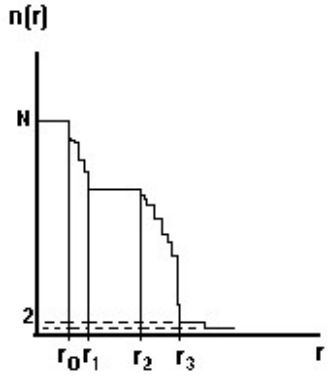


Figure 2. n(r) for the set of points shown in Figure 1a

We note that n(r) declines quickly as r increases from r_0 to r_1: circles around the points of one of the clusters fuse into a coherent set. At $r=r_1$ (see Figure 1b) all points of this cluster fuse into one set and n(r) remains constant until r increases to r_2. For $r>r_2$, n(r) again declines quickly: second cluster of points had fused into one coherent set (see Figure 1c). Thus we note that the appearance of the decline between r_2 and r_3 indicates the existence of a second cluster. At that point ($r=r_3$) $n(r_3)=2$ (there are only two connected areas). With increasing r the two clusters merge and n(r) becomes equal to 1. We also note the following about n(r): the steeper the decline and larger the step, the more reliable the prediction that there are two clusters. What happened here is this: the analysis of the graph n(r) enabled us to separate a given set of points appearing in Figure 1a into two clusters - in agreement with our visual perception of this image.

We also note that function n(r) is completely defined by a finite number of points on r axis. These specific points correspond to the values of r for which sets of circles merge. Between these points function n(r) is constant. To

apply the algorithm one can calculate all distances between initial points, arrange them in increasing, order and calculate n(r) for these points only.

For a set of randomly and uniformly distributed points, like those shown in Figure 3a, corresponding function n(r) is illustrated in Figure 3b. For a set of points as those in Figure 3a it is clear that n(r) will decline gradually and does not contain any steps.

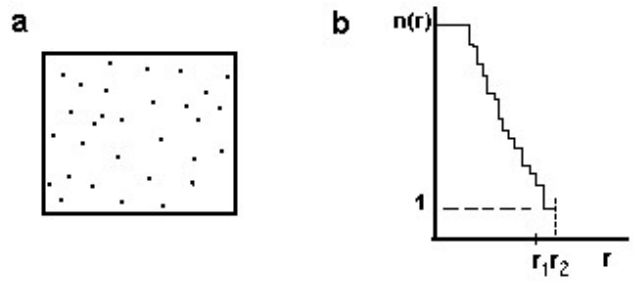
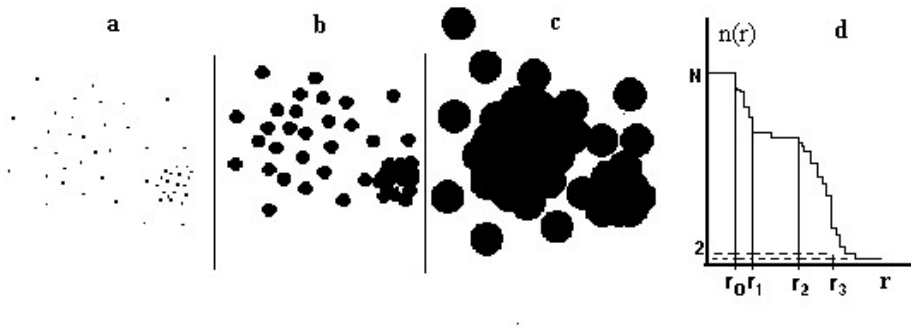


Figure 3. An example of uniformly and randomly distributed set of points (3a), and the corresponding n(r) function (3b)

If we were to add noise to the set of points shown in Figure 3 (see Figure 4a) our visual perception of the cluster structure will not be apparent. To discern the cluster we will go through the steps prescribed by the DD algorithm. Thus for the example depicted in Figure 4, the form of n(r) is like

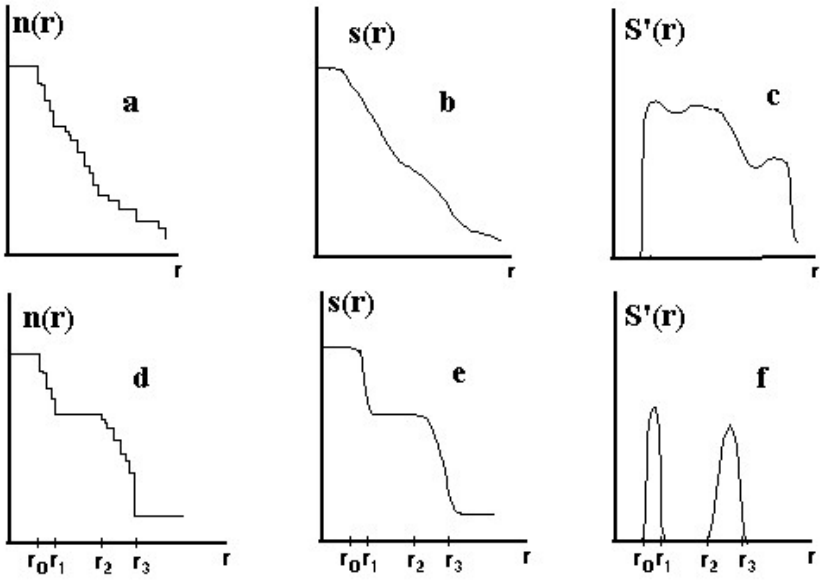


that shown in Figure 4d.

Figure 4. An example of a set of points with noise (4a), the outcome of the application of the algorithm (4b and 4c) and the resultant n(r) (4d)

We posit that the presence of a series of steps on n(r) establishes formal representation of the fact that dividing the points of the image into separate clusters is meaningful. The more step-like n(r), the more separate are the clusters (that is, the more organized the initial image). It should be noted that the degree of organization of a given function is understood in the GELFAND-TSETLIN sense: the organization of a function with a large

number of variables is describable by a small number of parameters (GELFAND and TSETLIN 1966). Normalized derivative of the function n(r) represents distribution of distances between “neighboring” points. The



entropy of that distribution can serve as a measure of organization of the initial image. To calculate the normalized derivative the function n(r) first has to be smoothed (for example, by averaging). Then the derivative of the smoothed function s(r) can be calculated. In the case of randomly distributed points (Figure 3a) the function n(r) is represented in the Figure 5a (see also Figure 3b). s(r) is shown in Figure 5b, and its derivative S’(r) is shown in Figure 5c. To normalize S’(r) one has to change the scale of that function so that the area under the curve S’(r) will be equal to 1. Now, for a case of two clusters (see also Figure 1), n(r) is shown in Figure 5d. s(r) is in Figure 5e, and its derivative S’(r) in Figure 5f .

Figure 5. An example of n(r), s(r) and S'(r) for the randomly and uniformly distributed points (4a through 4c), and for two clusters (4d through 4f).

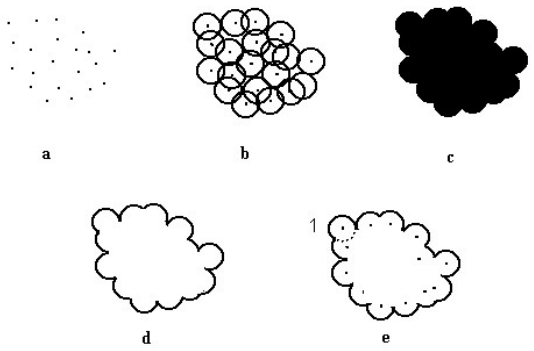
The entropy of S’(r) is $E=\int\{S'(r)\ln S'(r)dr\}$. That function reaches its maximum when S’(r) is a constant. It reaches its minimum when $S'(r)=\Delta(r^0)$, where $\Delta(r)$ is a delta-function (which means that S’(r) is concentrated in one point, r^0). Correspondingly, when there are no clusters (see Figure 3a), n(r) and s(r) will slowly decline (Figure 5a and Figure 5b). S’(r) will vary little (Figure 5c), and the entropy E will be large (close to 1). In the case of two apparent clusters (see Figure 1a), S’(r) will contract to two separate short intervals (see Figure 5f), and the entropy E will be small (close to 0). For clusters, as shown in Figure 4a, E will be between 0 and 1. Therefore

entropy E can serve as a measure of clustering of a given data. We thus posit that the DD algorithm can be applied to a weakly expressed (fuzzy) cluster. Moreover we are also able to resolve questions such as these: Which points of the cluster are border points? How to separate connected clusters?

4. Border Points and Separation of Connected Clusters

Now, let us ponder situation shown in Figure 6. We have established that the cluster can be described by our algorithm for which r=r' (recall that r' is the radius of the circle constructed so that the last remaining separate point of the cluster becomes part of the cluster). All circles in Figure 6b have radius equal to r'. Figure 6c depicts the area occupied by the cluster. This area is defined as a sum of all circles of radius r' build around all points that belong to the cluster (plus all interior regions).

The border of the area occupied by the cluster consist of arcs of radius r' (Figure 6d). Each of these (border) arcs belongs to a circle surrounding a specific point of the cluster. Such a point we will call a *border point*. It is clear that border points exhibit what we may call a different "degree" of belonging to the borderline". We may call this a degree of "borderiness". Some border points might be located on the very edge of the cluster (as illustrated by identified as point 1 in Figure 6d); a big part of the circle for this point belongs to the borderline. Some border points might be located a bit deeper in the body of the cluster; a small part of the circle for this point belongs to the borderline. Thus, for a given border point the "degree of borderiness" can be measured by the value of the arc exposed to the border: Arc/2*pi. For example, the "degree of borderiness" for the point identified as point 1 in Figure 6e is 2/3 for the part of the circle included in border (solid part of the



circle in Figure 6e). 1/3 of the circle belongs to the inner part of the cluster area (dashed line).

Figure 6. Illustration of the steps taken when finding border points. The set of points (6a); circles of radius r' around the points, when all circles merge (6b); the characteristic function (shown in black) (6c); border of the area shown in 6c (6d); border points (6e)

We recall that in discrete geometry, resolved on a regular grid, operators of erosion and expansion are useful when processing black-and-white images. For example, the erosion operator on 2-dimensional grid deletes all border points. We point out that the introduction of the notion of “border points,” as defined above, allows us to expand the use of these operators to irregular grids.

Now we are ready to explain how we identify clusters linked by a bridge. Such a situation is illustrated in Figure 7.

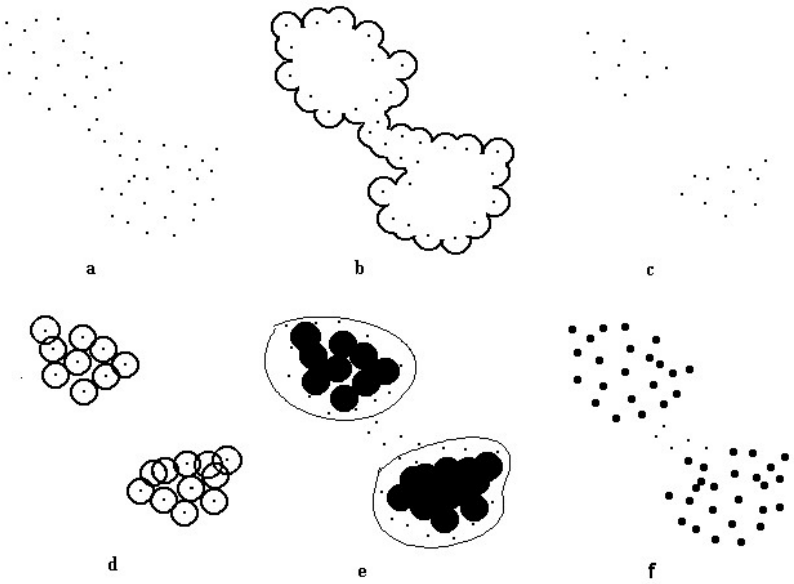


Figure 7. The initial set- two clusters linked by a bridge (7a); the border line and border points (7b); set of inner points - initial set after deleting the border points (7c); inner points surrounded by circles of radius r' (7d); expanded area - includes some previously deleted points, located at distance <=r' from circles shown in d (7e); final solution: two found clusters (7d)

The steps we will take are these:

1. Apply DD algorithm for cluster analysis. Determine from n(r) the value of r for which the growth of the cluster is completed ($r=r_1$). Thus an explicit cluster is found.
2. Mark all border points (as in Figure 7b).

3. Apply the erosion operator. That is, delete all border points (as in Figure 7c).
4. Again, apply DD algorithm.
5. If applying DD algorithm reveals two clusters (as in Figure 7d) proceed to step 6, if not, return to step 1.
6. Determine r=r' so that n(r')=2, i.e. all points are cleaved into two clusters. Resolve each cluster's area and its border (as in Figure 7e).
7. Apply expansion operator. That is, add to the identified clusters only these border points that were eliminated by the erosion operator in step 3. These border points are located closest to the clusters, that is, at a distance less then r' (as in Figure 7e). We have the final solution (as in Figure 7f).
8. If at least one of the border points of the initial set of points (marked at step 2) is now included in one of the clusters- we are done. If not – we return to step 7.

Thus, we divided the set of points into two clusters and a bridge (see Figure 7f). Now, to further illustrate the performance of DD algorithm, we use cluster patterns introduced by OSBOURNE and MARTINEZ (1995). This set of patterns illustrates a variety of visually "obvious" clusters that, as a group, are hard to define in simple terms (see Figures 8 and 10). For all cases represented in Figure 8, function n(r) is similar in appearance (see Figure 9).

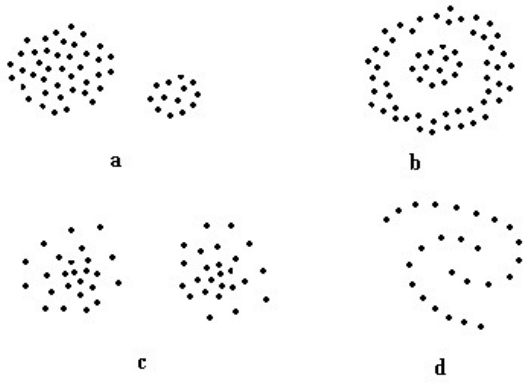


Figure 8. Examples of possible kinds of clusters (based on TOUSSAINT 1980a)

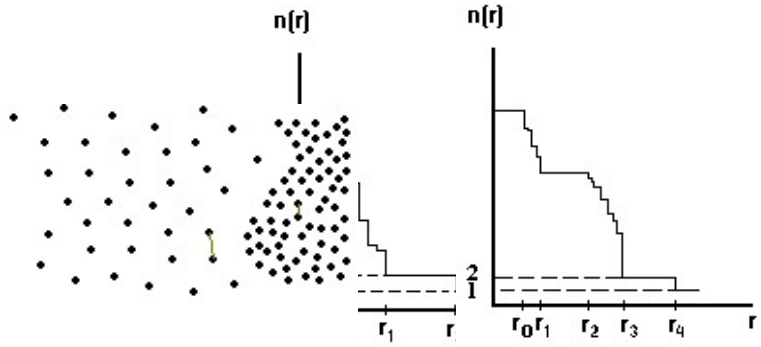


Figure 9. n(r) for examples shown in Figure 8

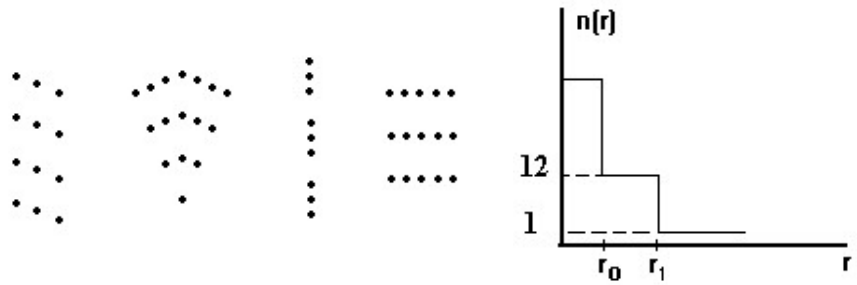
However, pattern illustrated in Figure 10, yields function n(r) of another type (see Figure 11). For n(r) in Figure 11 two steps are clearly in evidence. The curve drops quickly as r increases from r_0 to r_1. This is due to the fact that the circles around the points of the cluster on the right fuse into one set. The existence of the second step on the curve (from r_1 to r_2) reflects the fact that the smallest distance between the points of the cluster on the left is about two times larger than the largest one for the cluster on the right (r_1). Thus, the presence of this step on the curve is an evidence of the existence of at least one cluster.

Figure 10. A special case of two clusters

Figure 11. n(r) for example in Figure 10

The decline of n(r) between r_3 and r_4 occurs because the circles around the points of the cluster on the left are merging. At $r=r_3$ all circles for the points of the cluster on the left merge into one set. The last step of n(r) terminates at $r=r_4$, when two clusters merge, and n(r) drops from 2 to 1. Therefore, the existence of a step on n(r) between r_2 and r_3 is an evidence of the existence of the second cluster. We posit that the longer the steps and sharper the decline, the more certain the conclusion that two clusters exist. We recall that the longer the steps and sharper the decline of the function n(r), the higher the value of entropy E (measure of clustering), analogous to our psychological impression of the existence of clusters.

We now note that for repeating images as those shown in Figure 12, function n(r) is extremely simple. For these types of images the distances between all points in all clusters are equal. The distances between clusters are equal as well. Thus function n(r) drops first from N to the number of clusters and the second time to n=1 when all clusters merge (see Figure



13).

Figure 12. Some examples of multiple clusters

Figure 13. n(r) for multiple clusters shown in Figure 12

5. Cluster's Environment and the Case of a Single Cluster

Now, let us examine the case of a single cluster. We note that each example of clustering problem can be represented in certain environment. Most often this representation is rendered on a sheet of paper, or in a frame, or in a free space between the text. What is most important here is this: *depending on the spatial depiction (environment)*, the same set of points can be considered a cluster (see Figure 14) or not a cluster (see Figure 15) and that difference in perception reflects a fundamental point of Gestalt psychology. We wish to emphasize here that this difference is manifested in the form our function n(r) takes.

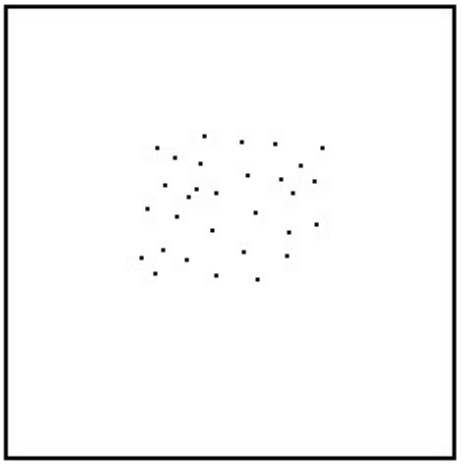


Figure 14. An example of set of points in a frame

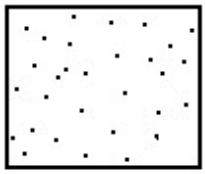


Figure 15. The same set of points as in Figure 14 but in a different frame

The reason is this: in reality expanding circles of radius r have a natural limit when they fill up allotted space (the sheet of paper, or the space inside the frame). Thus function n(r) for set of points shown in Figure 14 is the one illustrated in Figure 16. For the set of points shown in Figure 15, n(r) is shown in Figure 17. Graph of n(r) in Figure 16 has a large drop before $r=r_1$ and a step appears as well. This bespeaks the presence of at least one

cluster. In Figure 17 there is a drop in n(r), but no discernible step – this indicates the absence of a cluster in Figure 15, in agreement with our visual perception.

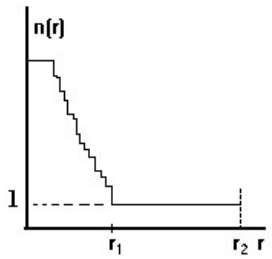


Figure 16. n(r) for the set shown in Figure 14

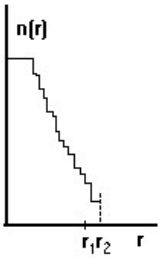


Figure 17. n(r) for the set shown in Figure 15

In summary, the entire cluster analysis procedure takes these two steps:

1. Apply the DD algorithm;

2. If you succeed in finding clusters, you are done. If you do not succeed, apply the 'erosion' operator (delete the border points) and return to step 1. When the set of points becomes empty, you are done.

Appendix

In this Appendix we describe an algorithm that attempts to model the ability of our visual perception to neglect the details of a picture. For this reasons the algorithm has been named "Damn-the-Details" (DD) method. Let us start with one-dimensional image – a curve on a plane. The curve y[x] (see Figure Ap1a) intersects the axis X at points x_0, x_1, …, x_N.

This set of points (nil points of the function y[x]) is a rough description of the curve that contains information about points at which the sign of the function y[x] changes, but disregards the value of the divergence of y[x] from 0. This rough description may be represented as function $y_0[x]=sign(y[x])$ (see Figure Ap1b), which changes its sign at points x_0, x_1, …, x_N and assumes constant values +1 or –1. The number of intervals between nil points for the function y_0 [x] is equal to N (as well as for the initial function y[x]). Now, let us find the shortest of these intervals. Let this be the interval $[x_k, x_{k+1}]$ with the length l_1. Let us now perform the operation of eliminating the given interval. For this we exclude the boundaries of this interval (x_k and x_{k+1}) from the set of nil points. Now in the place of three intervals $[x_{k-1}, x_k]$, $[x_k, x_{k+1}]$, and $[x_{k+1}, x_{k+2}]$ one interval is formed $[x_{k-1}, x_{k+2}]$ with the constant sign equal to the sign of y_0 [x] at the first of these three intervals $[x_{k-1}, x_k]$.

Therefore we now have function $y_1[x]$ (see Figure Ap1c). Now, let us apply the operation of eliminating the shortest interval to the new function $y_1[x]$. The length of that interval will be l_2 (l_2 >= l_1). Let us continue in the same way until all the intervals between the nil points have been eliminated.

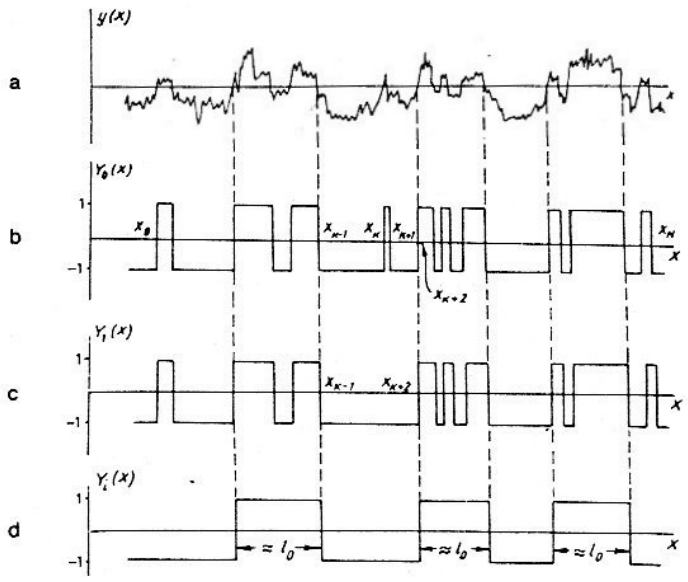


Figure Ap1. Calculating n(l) for y[x].
a: initial curve y[x]; b: quantized curve y_0 [x]; c: y_0 [x] after deleting the smallest interval;
d: y_0 [x] after deleting all intervals <l_0

Now let us construct function n(*l*), where n(*l*) is the number of intervals remaining after eliminating, step by step, all intervals with the length less or equal to *l.* Function n(*l*) is equal to N at *l*=0, it decreases discontinuously for all values of *l* equal to the length of the minimal interval at a given step of the elimination process. It maintains a constant value between these steps. In other words n(*l*) is a monotonically decreasing piecewise constant function.

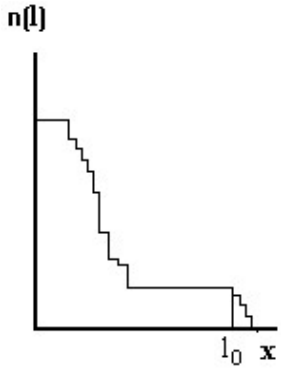


Figure Ap2. n(l) for y [x]

Function n(*l*) constructed for function y[x] is shown in Figure Ap2. The rough description of that function is this: steep slope - long step - another steep slope. That kind of function reflects the particular structure of the function y[x]. Roughly, this function can be described as a sequence of alternating intervals of positive and negative values with similar lengths close to l_0 (see Figure Ap1d). These intervals are complicated by a number of small perturbations. When considering our visual perception, these perturbations are just details that can be eliminated in order that we can give a rough description of the shape of the function y[x]. That is exactly what the DD algorithm is accomplishing. The first steep part of n(*l*) reflects the fact that there are a number of perturbations of similar length. The existence of the step reflects the fact that the rest of the objects on the curve (the intervals) are three to four times larger than the eliminated perturbations. This significant difference in the size of the objects is the reason why, in our visual perception, we recognize them as noticeable objects.

We point to the reader that it is easy to generalize the DD algorithm to higher dimensions. For two dimensions the initial function z[x, y] (for example, a gray image) can be transformed to a black-and-white image by subtracting the moving average and assuming z=+1, when z[x, y]>0, and z=-1, when z[x, y]<=0. The process of eliminating the smallest interval will be replaced by eliminating the spot of smallest area.

In all dimensions, the existence of long steps and short steep parts in a graph of function n(l) reflects the fact that function z is well organized. The shape of the graph of n(l) yields meaningful albeit rough description, eliminating the elements that our vision interprets as details.

Afterthoughts[1]

[1] By Shelia Guberman

A number of important issues were clarified during the discussions with our colleagues, which I would like to represent to the readers.

1. Generalization of the 2-dimensional case to the higher dimensions. The paper describes a formal procedure (algorithm) that simulates the human visual perception of images of a set of dots (i.e. in 2-dimensional space). In more precise terms, the algorithm simulates human **description** of the perception. We test this algorithm on a variety of images and verify the results. Next we postulate that the principles of description of dot sets in higher-dimensional spaces are the same as those for the 2 dimensions. This particular postulate cannot be proved by any experimental data for there is no human perception in the spaces of dimensions higher than 3. We point out that no modifications are required to generalize our algorithm to n dimensions. As a matter of fact, the algorithm was constructed in terms that are equally applicable to any dimensions. Specifically, the algorithm involves only three notions: point, radius and sphere, completely defined for any dimension. The function n(r) is a function of one variable. **The shape** of n(r) does not depend on the dimension of the space in which the clusters arise. For example, two clusters existing in the 5-dimentional space will generate the function n(r) similar to the one shown in fig. 2 that discerns two clusters in 2-dimentional space.

2. Our algorithm is in principle **different from all other clustering algorithms**: it does not use the number of clusters as an input and it does not return the number of clusters in a given data. It ***finds*** the clusters but it does not ***count*** them.

Of course, as soon as the clusters are found one can count them. It seems to us that this feature is in a good agreement with our cognition. Let us suppose that we have a page on which 13 well-separated clusters of points are drawn. In a glance we discern that on this page there is a number of well-defined clusters, but we do not know how many. To answer this question we have to count them, and that is a **different task**.

3. Concerning practical examples in the higher-dimensional space. This is both trivial and tricky. We need to distinguish between a set of objects and a set of characteristics (parameters) describing each object. When attempting a depiction we can ponder this in two ways: change the set of objects (keeping intact the description of the objects), or change the set of characteristics (keeping intact the set of objects). We wish to point out that in our case we can be more specific: the problem of practical clustering is a problem of finding an appropriate set of characteristics, which is a problem of a specific domain. For example, let us suppose that we have to cluster medical data in order to differentiate between two specific diseases. Will we use data sets that are not relevant to these diseases (for example, eye color)? No, it will make no sense. The data sets are relevant when efficacious in differentiating between particular diseases. Thus for specific diseases it might be reasonable to use relevant data, for example blood pressure, body temperature, EKG, behavioral symptoms etc. That means

that each of these parameters is **already clustered**. Consequently, the points representing two different diseases in an n-dimensional space are clustered as well.

What we are stating here is this: the tricky problem is the discovery of the set of appropriate parameters. By appropriate, we mean such a set of parameters that assures that objects belong to clusters. It turns out that this is not an algorithmic puzzle, but mainly a domain problem. That is, the parameters problem depends on the domain. The domain can be medical, geological, and so on.

4. "Proximity" and "similarity" in n-dimensional space. The notions of "proximity" and "similarity" in psychology are well defined and refer to the objects in the real (human) 3-dimensional world (points, lines, cars etc.) as observed by the human eye. In the n-dimensional spaces (n>3) human (real) geometry does not apply and the notion of "proximity" loses meaning. In the n-dimensional space objects are supplanted by representation as the abstract points. Since humans do not function in the n-dimensional space the notion of "similarity" is expressed by measuring the distance between points in an abstract n-dimensional space - the more general sense of proximity. For example, the similarity in size will be represented by the proximity of objects along the "size" axis; similarity in shape, by proximity of objects along axes describing the shape of the objects. Therefore, proximity (in more general sense), which is measured by the distance between points, covers both notions of the real (physical) world - "proximity" and "similarity". This is true not only for n>3, but for the lower dimensions as well, when the 1- or 2- or 3-dimensional space is an abstract space (temperature, pressure etc.).

Zusammenfassung

In diesem Artikel erörtern wir das Problem der Bildung von Cluster und fassen es als ein gestalttheoretisches (holistisches) Problem auf. In diesem Zusammenhang werden wir

1. den DD-Algorithmus auf den Prozeß anwenden, der eine holistische bzw. gestalthafte Wahrnehmung der Clusterbildung simuliert; in Appendix II wird der DD-Algorithmus in Kürze dargestellt, eine genaue Beschreibung findet man in GUBERMAN (1983);

2. zeigen, daß die algorithmische Simulation des Gestalt-Ansatzes Lösungen für eine große Vielfalt an Problemen der Clusterbildung bietet, wobei die Anzahl der Cluster nicht ‚a-priori' als bekannt vorausgesetzt zu werden braucht;

3. ein ‚Clustering'-Maß einführen, das die Anwendung des DD-Algorithmus auf Fuzzy-Cluster erlaubt;

4. den Begriff von ‚Grenzpunkten' (der für regelmäßige Gitter wohldefiniert ist) verallgemeinern und so eine Lösung des Clusterbildungs-Problems für verbundene Cluster ermöglichen;

5. den Begriff der Cluster-Umgebung einführen und so die Anwendung auf das Clusterbildungs-Problem für singuläre Cluster erweitern.

Summary

In this paper we discuss the clustering analysis problem and identify it as a Gestalt (holistic) problem. In this context we:

1. Apply the DD algorithm to the situation that simulates the holistic that is Gestalt perception of clustering; DD algorithm is described in detail in GUBERMAN (1983), and in brief, in Appendix II;

2. Show that the algorithmic simulation of Gestalt approach offers a solution to a broad variety of clustering problems, where an 'a priori' knowledge of the number of clusters is not required;

3. Introduce a measure of 'clustering' that permits the application of the DD algorithm to fuzzy clusters;

4. Generalize the notion of 'border points' (well defined on regular grids) thus resolving the clustering problem for connected clusters;

5. Introduce the notion of cluster's environment and expand the solution of clustering problem to the case of a single cluster.

References

ANDENBERG, R. (1973): *Cluster Analysis for Applications.* New York: Academic Press.

GELFAND, I. M., TSETLIN, M. L. (1966): Mathematical modeling of central nervous system mechanism. In NAUKA, *Models of Structural Functional Organization of Some Biological Systems*. Moscow.

GLASS, L., PEREZ, R. (1973): Perception of random dots interference pattern. *Nature*, *246*, 360-362.

GUBERMAN, S. (1983): Computer vision and Gestalt theory. *Soviet Psychology*, *22*, 89-106.

HARTIGAN, A. (1975): *Clustering Algorithms.* New York: Wiley.

MATULA, D. W., SOKAL, R. R. (1980): Properties of Gabriel graphs relevant to geographic variation research and the clustering points in the plane. *Geographic Anal., 12*, 204-222.

McKENNA, F. P. (1985): Modifying the Gestalt factor of proximity: theories compared. *Perception*, *14*, 359-366.

MURTAGH, F., HECK A. (1987): *Multivariate Data Analysis*, D. Redell Norwell.

OSBOURN, G. C., MARTINEZ, F. R. (1995): Empirically defined regions of influence for clustering analyses. *Pattern Recognition*, *28*, 1793-1806.

TOUSSAINT, G. T. (1980): Pattern recognition and geometrical complexity. *Proc. 5th. Int. Conf. Patt. Recog.*, 1324-1347.

TOUSSAINT, G. T. (1980a): The relative neighborhood graph of a finite planar set. *Pattern Recognition*, *12*, 261-268.

WERTHEIMER, M. (1938): Laws of Organization in Perceptual Form. In W. ELLIS (Ed.): *A Source Book of Gestalt Psychology*, London: Routledge and Kagan.

Addresses of the authors

Shelia Guberman

Barnhill Technologies

PO Box 2411

Cupertino, California 95015, USA

Phone and voice mail: +408-255-4648

e-mail: guboil@hotmail.com

Wita Wojtkowski
Networking, Operations, and Information Systems
Boise State University, 1910 University Drive,
Boise, Idaho 83725, USA
Phone and voice mail: +208-426-1372
e-mail: wwojtkow@boisestate.edu