Pattern Recognition Letters 12 (1991) 657-664
North-Holland

November 1991

Characterization and detection of noise in clustering

Rajesh N. Dave

Department of Mechanical and Industrial Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

Received 20 August 1991

Abstract

Dave, R.N., Characterization and detection of noise in clustering, Pattern Recognition Letters 12 (1991) 657-664.

A concept of 'Noise Cluster' is introduced such that noisy data points may be assigned to the noise class. The approach is developed for ojbective functional type (K-means or fuzzy K-means) algorithms, and its ability to detect 'good' clusters amongst noisy data is demonstrated. The approach presented is applicable to a variety of fuzzy clustering algorithms as well as regression analysis.

Keywords. Clustering, noise cluster, classification amongst noisy data, K-means algorithms, fuzzy K-means algorithms.

1. Introduction

Cluster analysis is an important tool in many scientific disciplines, and many clustering methods are available (see e.g. Everitt (1974) or Jain and Dubes (1988)). A single clustering method or algorithm cannot solve all the possible clustering problems, hence the proliferation of many techniques. Most clustering methods are plagued with the problem of noisy data, i.e., characterization of *good* clusters amongst noisy data. In some cases, even a few noisy points or outliers affect the outcome of the method by severely biasing the algorithm. The noise that is just due to the statistical distribution of the measuring instrument is usually of no concern. On the other hand, the completely arbitrary noise points that just do not belong to the pattern or class being searched for are of real concern. A good example of that is in image processing, where one is searching for certain shapes, for instance, amongst all the edge elements detected. An approach that is frequently recommended (for example, Jain and Dubes (1988)) is where one tries to identify such data and removes it before application of the clustering algorithms. In many cases, however, that may not be possible or it may be extremely difficult.

In this paper, a class of algorithms based on the square-error clustering (a sub-class of partitional clustering) is considered. The performance of the algorithms of this kind is highly susceptible to outliers or noisy points. The K-means type algorithms is one example where each point in the data-set must be assigned to one of the clusters. Because of this requirement, even the noise points have to be allotted to one of the good clusters, and that would deteriorate the performance of the algorithm. One approach to solve this problem is as proposed by Jolion and Rosenfeld (1989), where each data point is given a weight proportional to the density of data points in its vicinity, thus assigning higher weights to the points belonging to the clusters, while assigning lower weights to the noise or background points. Thus the approach results in preprocessing of the data in order to reduce the bias due to noise background. The

technique is based on utilizing the inter-cluster statistics to estimate local density around each point and then the weights are assigned based on either the density or information derived from the density histogram. This approach is shown to work well on examples where there is uniformly distributed noise in the background. Weiss (1988) reports a technique based on separating the data of interest from random noise, utilizing the maximum likelihood principle. This technique is shown to work well for fitting a single line to *good* data amongst noisy background. The approach is not extended to multiple clusters, and has a disadvantage of getting trapped at local minima.

A different approach is presented in this paper, where a concept of a *noise cluster* is introduced, with the hope that all the noisy points can be dumped into a noise cluster. The fuzzy version of the K-means algorithm is an attractive candidate for the new approach considered here, since with that approach, one can also obtain a relative degree of belonging of a point to a noise cluster. The approach is based on the concept of first defining a noise cluster and then defining a similarity (or dissimilarity) measure for the noise cluster. Thus if one is looking for a certain number of *good* clusters, then the formulation of the problem requires defining an additional cluster that will collect the noisy data points.

First, a background on K-means type algorithms is presented, along with the popular fuzzy c-means algorithm. This is followed by the introduction of the concept of noise clusters and the similarity measure for noise. An algorithm based on that concept is presented next, along with several numerical examples illustrating the utility of the algorithm. The paper is concluded by the discussion of the results and comments on the extensions of this approach to families of clustering algorithms.

2. K-means algorithms

There are many versions of the K-means type partitional clustering algorithms, see for example an appropriate text book (Anderberg (1973), Everitt (1974), or Jain and Dubes (1988)). A basic version is considered here for the sake of presenting the new approach, keeping in mind that any version can be adapted to the new approach at least in principle. The form of the algorithm is as follows:

2.1. *Iterative K-means algorithm*

Step 1. Select initial location of cluster centers.
Step 2. Generate a (new) partition by assigning each point to its closest cluster center.
Step 3. Calculate new cluster centers as the centroids of the clusters.
Step 4. If the cluster partition is stable, stop; else go to Step 2.

The above algorithm needs specification of user supplied parameters, which are particular to the form of software used. The distance measurement criterion, a major variable, is frequently the Euclidean distance. It can be observed from Step 2 of the algorithm that each data point will be assigned to one of the classes based on its proximity. Thus even an outlier will be assigned to one of the classes. Since that assignment affects the new cluster center computation in Step 3, an outlier may significantly bias the final result. An example is shown in Figure 1a, where there are two apparent clusters, and an outlier point. Application of the above algorithm with the Euclidean norm to find two classes results in the partition where the outlier itself is one cluster, while the rest of the

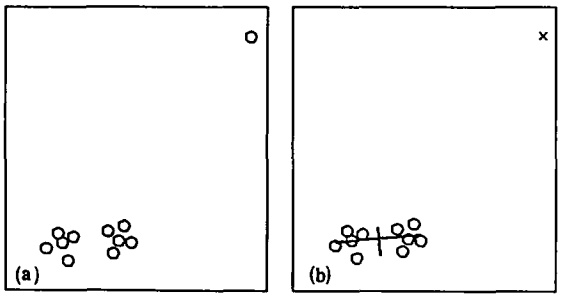

Figure 1. Two clusters and one outlier point; (a) data-set, (b) results of the conventional K-means algorithm. Partition denoted by 'o' and '×', indicating the outlier itself as one cluster.

points form another cluster (see Figure 1b). In Figure 1b, the results are given, where for each cluster, two crossing perpendicular lines depict cluster prototypes, such that their intersection is the center, and the lines represent the eigenvectors. Lengths of the eigenvectors depend on the extent of the clusters, i.e., how far each cluster extends along each principal direction. The partition is shown by using different markers. The points marked by 'O' and '×' denote two different clusters. Although this example is somewhat of an extreme case, it is used here to stress the need for a method that automatically avoids bias due to noise or outliers.

It is necessary to examine the mathematical formulation of the above algorithm in order to gain an understanding of its performance against noise. The problem formulation can be based on mathematical programming, where the weighted sum of squared distances between the points and cluster prototypes is minimized. Following the notations in Bezdek (1981), the minimization functional is

$$J(U,v) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m (d_{ik})^2 \tag{1}$$

where the distances are defined as

$$(d_{ik})^2 = \langle x_k - v_i \rangle^{\mathrm{T}} A_i \langle x_k - v_i \rangle \tag{2}$$

with x_k being the feature vector of point k ($k=1$ through n number of points), and v_i the cluster prototype of class i ($i=1$ through c number of classes). The distances are measured through the norm induced by symmetric, positive definite matrices A_i (for Euclidean norm, they are identity matrices), m is the exponent, $1<m<\infty$, and u_{ik} is the membership of point k in class i. The following restrictions apply to the membership u_{ik}:

$$\sum_{i=1}^{c} (u_{ik}) = 1. \tag{3}$$

If the memberships u_{ik} are hard, i.e. 0 or 1, then the exponent $m>1$ has no meaning, hence $m=1$ is chosen. The membership value is 1 when an object belongs to a class, and 0 when it does not. The memberships can also be fuzzy, that is it can also take the value between 0 and 1. The algorithm mentioned above is based on hard memberships. If the memberships are fuzzy, the following equation for u_{ik} can be obtained using the Lagrange multiplier technique (Bezdek (1981)), and can be used at Step 2 of the algorithm:

$$u_{ik} = \frac{1}{\sum_{j=1}^{n} [(d_{ik})^2/(d_{jk})^2]^{1/(m-1)}}. \tag{4}$$

Careful observation of the formulation shows that the restriction imposed by equation (3) causes the bias due to noise. Since the sum of the memberships of a point k must be equal to unity, means that the point must be assigned to one of the classes. Thus even an outlier must belong to a class. This can create the problem of bias. Even in case of fuzzy memberships, an outlier still must be assigned to one or more classes.

It seems that if the restriction imposed by equation (3) can be relaxed, then one may be able to prevent the noise point from adversely affecting the outcome. If a technique can be developed such that the restriction of equation (3) does not hold for the noise points, then their effect can be minimized. Another related problem can be demonstrated by the example in Figure 2a. Here, there are two apparent clusters, with three noise points falling along the bisector. Normally, the fuzzy classification has an advantage in a situation like this where a point in the middle would get equal but low values of memberships, indicating that something is wrong with the point or the classification. In this example, however, since the points fall along the bisector of two cluster centers, the memberships of all points would be the same,

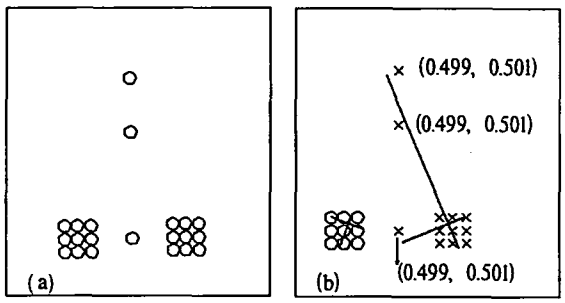

Figure 2. Two clusters and three outlier points along the bisector; (a) data-set, (b) results of the conventional K-means algorithm. Partition denoted by 'O' and '×', showing all three outliers classified in one cluster. Their membership values are printed.

showing no discrimination between closer points and farther points. The results are shown in Figure 2b, where the membership of each of the three points is nearly equal, i.e., 0.499 in the left cluster and 0.501 in the right cluster. Thus although the three points are at different distances, the membership values do not show discrimination. This fact negates the advantage of having fuzzy memberships. Again, this happens because of the restriction imposed by equation (3). Thus there is a need to develop a method to relax that restriction. An approach to do that is presented in the next section.

3. Noise clusters

The bias due to noise is a classical problem affecting all clustering algorithms. A good solution to this problem does not exist, although the field of clustering has been in existence for decades. An ideal solution would be one where the noise points get automatically identified and removed from the data. The concept of having an approach where one can define one cluster as the noise cluster is also promising, provided there is a way in which all the noise points could be dumped into that single cluster. In the examples shown in Figures 1 and 2, if one searched for three clusters instead of one, it is highly probable that the noise points would get dumped into one of the three clusters, while the other two clusters may exhibit the required partition. The problems in real life, however, may get more complex. For example, in the data shown in Figure 3, there are two apparent clusters amongst widely scattered noise. The algorithm of Section 2.1 would fail in a situation like this if asked for a two-partition. The algorithm would fail even for a three-partition of the data, since all the noise points would not get classified into a single cluster. Even the schemes based on minimal spanning tree (MST) such as one by Zahn (1971) may not work well, because the noise points form a trail between two good clusters in this example. What is really needed is some way of characterizing the points that belong to a noise cluster. On the other hand, a scheme that would allow us to define noise as a prototype cluster would also work.

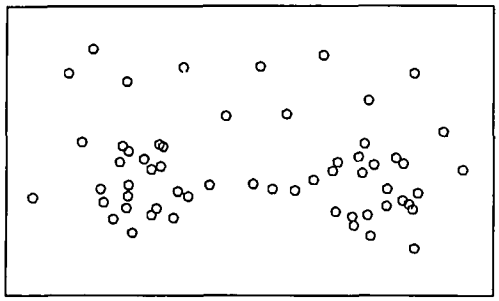

Figure 3. Two clusters among many noise points.

3.1. Noise prototype

Based on the arguments presented above, the noise prototype is defined as follows:

Definition 1 (*Noise prototype*). Noise prototype is a universal entity such that it is always at the same distance from every point in the data-set. Let v_c be the noise prototype, and x_k be the point in feature space, $v_c, x_k \in \mathbb{R}^p$. Then the noise prototype is such that the distance d_{ck}, distance of point x_k from v_c, is

$$d_{ck} = \delta, \quad \forall k. \tag{5}$$

The above definition does not tell us what the distance δ is. It simply says that all the points are at the same distance δ from v_c. Physically, this means that all the points have equal *a priori* probability of belonging to a noise cluster. This makes sense, since given no prior information, all the points should have an equal probability of falling into a noise class. It is hoped, however, that as the algorithm progresses, the *good* points increase their probability of being classified into a *good* cluster.

Let there be $c-1$ *good* clusters in the data-set, and let the cth cluster be the noise cluster. Then the functional J_N including the noise cluster is defined in the same manner as equation (1):

$$J_N(U,v) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m (d_{ik})^2 \tag{6}$$

where the distances are defined as

$$(d_{ik})^2 = \langle x_k - v_i \rangle^T A_i \langle x_k - v_i \rangle \quad \text{for all } k \text{ and } i = 1 \text{ to } c-1, \tag{7a}$$

and,

$$(d_{ik})^2 = \delta^2, \quad \text{for } i = c. \tag{7b}$$

Before the above can be minimized, the distance to the noise cluster δ must be specified. Assuming that the distance δ is specified, the minimization can proceed.

Theorem 1. *Let all A_i be fixed as identity matrices, and m be fixed, $m = 1$ for hard memberships, while $1 < m < \infty$ for fuzzy memberships; let $X = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^p$ have at least $c < n$ distinct points.*

Then J_N is globally minimum only if:

For $u_{ik} \in [0,1]$, i.e., fuzzy memberships,

$$u_{ik} = \frac{1}{\sum_{j=1}^{n} [(d_{ik})^2/(d_{jk})^2]^{1/(m-1)}}, \tag{8a}$$

and for $u_{ik} \in \{0, 1\}$, i.e., hard memberships,

$$\begin{aligned} &u_{ik} = 0 \quad \forall i \neq j, \\ &u_{jk} = 1, \quad j \ni d_{jk} = \min(d_{ik},\ i = 1 \text{ to } c); \end{aligned} \tag{8b}$$

and

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_i}{\sum_{k=1}^{n} (u_{ik})^m}, \quad \text{for } i = 1 \text{ to } c-1. \tag{9}$$

The proof of the above theorem can be obtained using standard optimization techniques, and is not presented here due to lack of space. Note that the noise prototype v_c is as defined by equation (5), and the distances are as defined by equation (7).

The above theorem provides necessary conditions for minimization of J_N, provided the noise cluster distance δ is specified. From equations (8) and (9), the memberships u_{ik} and cluster centers v_i can be calculated.

The question of determining δ needs to be addressed. First, the effect of δ on the equations of the above theorem needs examination. The distance δ appears in the functional (equation (6)), and then it also appears in membership calculation (equation (8)). The minimization of the functional requires the memberships calculated by equation (8). Careful observation of equation (8) shows that for any point x_k, the membership u_{ik} in a cluster i depends not only on the distance from the cluster i, but also on the distances from all other clusters. This equation represents a weighted average of inverse of the distances, thus a point will have the highest membership for the cluster that is closest. Thus if the δ is chosen to be very small, then most of the points will get classified as noise points, while if the δ is large, then most of the points will be classified into clusters other than the noise cluster. A proper selection of δ will result in a classification where the points that are actually close enough to the *good* clusters will get correctly classified into a *good* cluster, while the noise points that are away from the *good* clusters will be closer to the noise cluster and would get classified into the noise cluster.

Another observation that can be made from equation (8) is that it allows one to rewrite the constraint of equation (3) in the following way:

$$\sum_{i=1}^{c-1} (u_{ik}) = 1 - u_{ck} \quad \text{or} \quad 0 \leqslant \sum_{i=1}^{c-1} (u_{ik}) \leqslant 1. \tag{10}$$

In the above equation, the constraint on sum of memberships for $c-1$ good clusters is relaxed through having the $(1 - u_{ck})$ term. Thus depending on the value of u_{ck}, the sum will be anywhere from 0 to 1. This means that the method has a potential of setting the membership sum of a noise point over $(c-1)$ *good* clusters equal to almost zero, thus effectively eliminating the noise point from the picture.

The algorithm for the above theorem can be constructed based on the fixed point iteration scheme and is presented below.

3.2. Noise clustering algorithm

The algorithm is based on the standard K-means type algorithms, using the results of the above theorem.

Step 1. Fix the number of clusters c, and fix the exponent m. For hard memberships, $m = 1$. Select initial locations of cluster centers v_i. Specify noise cluster distance δ.

Step 2. Generate a (new) partition using equation (8).

Step 3. Calculate new cluster centers using equation (9).

Step 4. If the cluster partition is stable, stop; else go to Step 2.

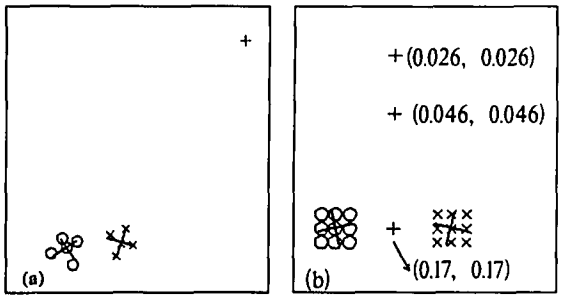

Figure 4. Results of Noise Clustering algorithm; (a) data of Figure 1a, the outlier is classified as a noise point, denoted by '+', (b) data of Figure 2a, the outliers are classified as noise points, denoted by '+'. The membership values are printed, showing the variation due to distance from the good clusters.

The only problem is the specification of noise cluster distance δ. From a practical viewpoint, prespecification of δ is not easy, because in most cases, information to decide the value of δ is not available. The value of δ would be different for different problems, and would be based on the statistical parameters of the data-set. A scheme based on the average interpoint distances is proposed for prediction of δ. Interpoint distances reflect structural relationship among the feature points. Therefore, the value of δ can be based on the statistics of interpoint distances. Based on this argument, a simplified statistical average is used to calculate δ,

$$\delta^2 = \lambda \left[\frac{\sum_{i=1}^{c-1} \sum_{k=1}^{n} (d_{ik})^2}{n(c-1)} \right] \tag{11}$$

where λ is the value of the multiplier used to obtain δ from the average of distances. Based on equation (11), δ can be calculated at each iteration of the sequence. Equation (11) has to be used along with equation (9) at Step 3 of the algorithm. This new algorithm was coded and tested on several data-sets, including the data-sets shown in Figures 1–3. The results are presented in the next section.

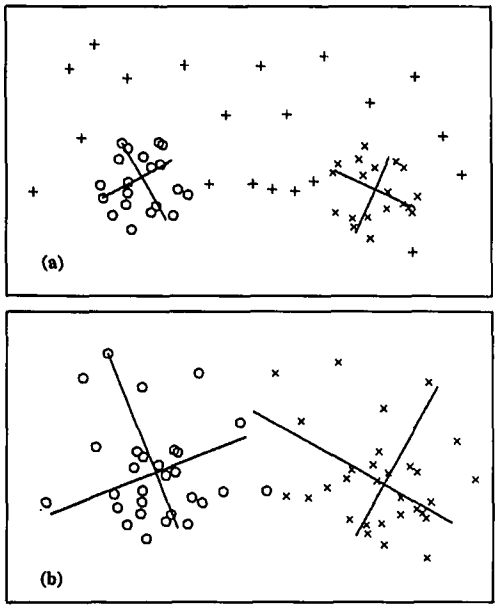

Figure 5. (a) Result of Noise Clustering algorithm on data of Figure 3, the noise points are correctly classified, denoted by '+', (b) result of the conventional K-means algorithm, cluster centers are affected by noise.

4. Numerical examples

Several examples of artificially generated data are considered. The focus of this paper is to demonstrate the new concept, thus the use of such examples is justified. The Noise clustering algorithm with fuzzy memberships is applied to the examples shown in Figures 1 and 2. In Figure 4a, the results of the example shown in Figure 1a are shown where for each cluster, two crossing perpendicular lines depict cluster prototype, such that their intersection is the center, and the lines represent the eigenvectors. The partition is shown by using different markers. The points marked by 'O' and '×' denote two *good* clusters, while the points marked by '+' denote the *noise* cluster. The outlier is correctly partitioned into the *noise* cluster. In Figure 4b, the results of the example shown in Figure 2a are shown in the similar manner. The partition, shown by the use of different markers, is excellent. Three outliers are classified into the *noise* cluster and their memberships (equal for each *good* cluster) are printed. The nearer point has a higher membership than the farther points. The lowest membership of a *good* point into a *good* cluster was 0.648, indicating that the membership value of 0.17, which is much lower than 0.648 must be a noise point. For both these examples, the value of λ was chosen as 0.1. The

algorithm was also applied to the example shown in Figure 3, and the results are shown in Figure 5a. Partition is very good, and the cluster prototypes are also characterized very well. The trail of the points in the middle as well as the scattered points are classified into the *noise* cluster. The same example was tried with a conventional algorithm of Section 2.1, and the results were poor as shown in Figure 5b.

The examples considered so far are such that the *good* clusters are well separated and are essentially round. The effect of noise on a clustering algorithm for such cases is not as bad as in the cases where the separation is less and the shapes are not round. For such a case (Figure 6a), the K-means algorithm would not work even without the noise, because the clusters are very elongated. An adaptive fuzzy version by Gustafson and Kessel (1979) may be used. That algorithm, called the GK algorithm was applied to the data in Figure 6a, and the results are shown in Figure 6b. Notice the poor partition because of noise. The GK algorithm was modified to form a noise clustering algorithm, and was applied to the data. The results are shown in Figure 6c. The partition as well as the characterization of the linear clusters is excellent. Although two *good* clusters are not too close in the context of Euclidean distance, they are indeed close because the line of the right cluster passes through the left cluster.

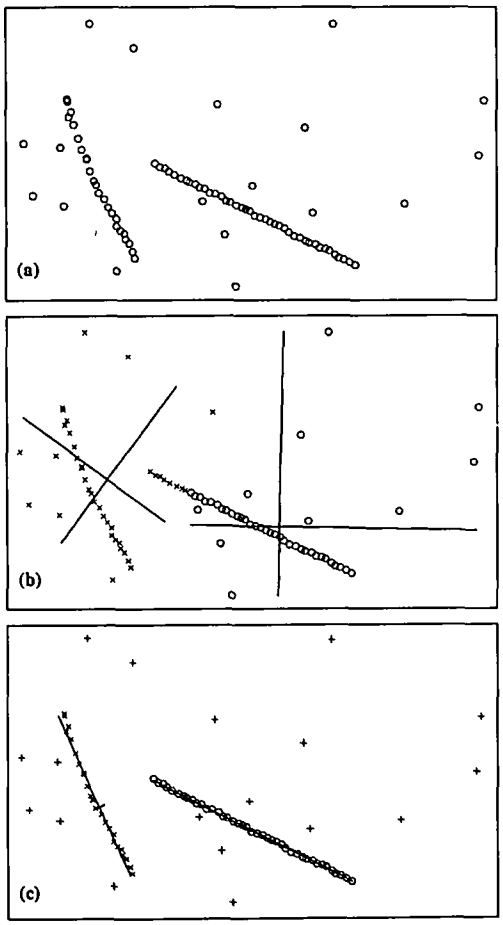

Figure 6. Two linear clusters among noise points; (a) the dataset, (b) results of the conventional GK algorithm, which failed to characterize two linear clusters, (c) results of Noise GK algorithm, showing correct partition as well as characterization.

5. Concluding remarks

A concept of characterizing and detecting noise clusters is introduced. This approach is shown to eliminate the bias due to noise for squared-error type clustering algorithms. The *Noise Clustering Theorem* shows that by defining the noise cluster as in Definition 1, the restriction imposed by equation (3) is relaxed, and the result is the ability to remove the noisy points from the domain of *good* clusters. The resulting algorithm is very simple and yet accomplishes the task of separating noise from the data. It also creates excellent partitioning of the data, besides giving good results for the cluster locations. The resulting algorithm is also much simpler to implement than the algorithms in Weiss (1988), and Jolion and Rosenfeld (1989). The concept presented here can be easily applied to regression analysis (to be reported elsewhere), where a set of data with noise has to be fitted by a single curve.

The examples included show the ability of this approach to come up with good partitions in noisy data. Two important aspects of the noise algorithm are brought out through the example shown in Figure 6. The first aspect is that the concept of noise algorithm can be adapted for all the different squared-error type algorithms. The second aspect is that this example shows that this concept has a high potential in a variety of clustering applications. This example is similar to noisy edge data

in digital images, indicating a potential application in image processing.

It is emphasized that when the cluster prototypes in square-error clustering are other than points, then the effect of noise is more severe. The examples of such prototypes include lines, planes and other curved surfaces such as hyper-spherical and hyper-ellipsoidal (see Dave (1990)). Application of the *noise* cluster concept to these cases will be the topic of a future paper. For prototypes other than points, the technique presented in Jolion and Rosenfeld (1989) would be more difficult to implement, since defining local density may become complex, if not impossible.

Another area of further investigation is the selection of δ. The preliminary suggestion is based on the statistical parameter of estimated interpoint distances. This approach seems to work well, and the results are not very sensitive to the range of the values of the multiplier λ. For example, with round clusters (using the K-means type algorithm), the value of λ may range from 0.5 to 0.05. For elongated clusters (using the GK algorithm), the value of λ may range from 0.05 to 0.005. In both cases, the variation in λ can be up to an order of magnitude. Nonetheless, more investigation is required with regards to selecting δ. One possibility is to use the standard deviation of interpoint distances to estimate the value of λ.

In summary, a powerful new approach for characterization and detection of noise data is presented. The approach can be extended to a variety of clustering algorithms.

Acknowledgements

The author is grateful to Professor Azriel Rosenfeld for pointing out important references related to the paper.

References

Anderberg, M.R. (1973). *Cluster Analysis for Applications*. Academic Press, New York.

Bezdek, J.C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York.

Dave, R.N. (1990). Fuzzy shell-clustering and applications to circle detection in digital images. *Int. J. General Systems* 16(4), 343–355.

Everitt, B.S. (1974). *Cluster Analysis*. Wiley, New York.

Gustafson, E.E. and W.C. Kessel (1979). Fuzzy clustering with a fuzzy covariance matrix. *Proc. IEEE CDC*, San Diego, CA, 761–766.

Jain, A.K. and R.C. Dubes (1988). *Algorithms for Clustering Data*. Prentice-Hall, Englewood Cliffs, NJ.

Jolion, J. and A. Rosenfeld (1989). Cluster detection in background noise. *Pattern Recognition* 22(5), 603–607.

Weiss, I. (1988). Straight line fitting in a noisy image. *Proc. Computer Vision and Pattern Recognition*, 647–652.

Zahn, C.T. (1971). Graph-theoretical methods for detecting and describing Gestalt clusters. *IEEE Trans. Computers* 20, 68–86.