A Novel Multiseed Nonhierarchical Data Clustering Technique

D. Chaudhuri and B. B. Chaudhuri

Abstract—**Clustering techniques such as *K*-means and Forgy as well as their improved version ISODATA group data around one seed point for each cluster. It is well known that these methods do not work well if the shape of the cluster is elongated or nonconvex. We argue that for a elongated or nonconvex shaped cluster, more than one seed is needed. In this paper a *multiseed clustering algorithm is proposed*. A density based representative point selection algorithm is used to choose the initial seed points. To assign several seed points to one cluster, a minimal spanning tree guided novel technique is proposed. Also, a border point detection algorithm is proposed for the detection of shape of the cluster. This border in turn signifies whether the cluster is elongated or not. Experimental results show the efficiency of this clustering technique.**

Index Terms—**Classification, minimal spanning tree, multiseed clustering, pattern recognition.**

I. Introduction

Cluster analysis is the formal study of algorithms and methods for grouping or classifying objects and data. Clustering is a useful and important technique in image processing and pattern recognition [2], [9], [11], [15], [16]. There exist two classes of clustering techniques, namely *hierarchical* and *nonhierarchical* techniques. A hierarchical clustering is a nested sequence of partitions, whereas a nonhierarchical clustering is a single partition. K-means [2] and the Forgy [8], [19] algorithms are among the oldest nonhierarchical techniques. Another approach, namely ISODATA [20] clustering is a modification of K-means technique by including additional criteria to obtain better clusters. The K-means class of methods start with some initial *seed* or *representative points* and grow cluster around them. There exist two basic problems related to all seed based techniques. One is the choice of appropriate initial seed points. The other problem is that these techniques are effective for clusters of spherical and ellipsoidal shape. For clusters of more complex and elongated shape, good results may not be obtained. This paper is concerned with both problems.

From the basic idea of K-means algorithm, it is clear that a seed point has the best capability to collect data if the cluster around it is hyperspherical in shape. Any elongated or nonconvex cluster can be considered as the union of a few distinct hyperspherical clusters. To capture the data of an elongated cluster, we should, therefore, consider more than one seed point in the cluster. This is the central idea of the *multiseed clustering* method proposed in this paper.

To get an idea about the shape of the cluster, it is useful to find out the border points. The border points signify whether the cluster is elongated or not. A novel border point detection algorithm is proposed in Section II. According to the shape of the cluster one should be able to find automatically the seed points. The seed point detection algorithm and the splitting technique are proposed in Section III. To assign multiple seeds to a cluster a novel merging technique based on minimal spanning tree of seeds and density at the border region of two initial clusters is also proposed in Section III. The results on synthetic data are presented in Section IV.

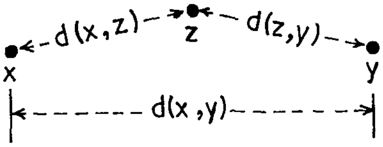


Fig. 1. Nearly opposite points with respect to a fixed point.

II. Border Point Detection

Given a homogeneous set of points in two-dimensional (2-D) and three-dimensional (3-D) space, we have a perceptual notion about the points lying on the border as compared to those of the interior of the data set. Detection of the border points and the interior points of a dot pattern is a difficult job. In our work [17], [18] we described an approach where the low density points are border points. But density alone cannot capture the notion of border points, because if the interior portion of any pattern is sparsely populated then the interior points are also detected as border points. We have a perceptual notion about the points lying on the border as compared to those of the interior of the data set. Border points are not surrounded by other points in all directions while the interior points are. The present approach of border point detection is based on this observation.

Definition 1: A point $\mathbf{x} \in S$ is said to be a nearly *opposite point* of $\mathbf{y} \in S$ with respect to $\mathbf{z} \in S$ if $\mathbf{x}$, $\mathbf{z}$ and $\mathbf{y}$ ($\mathbf{x} \neq \mathbf{z}$ and $\mathbf{y} \neq \mathbf{z}$) almost lie in a straight line, i.e., if

$$I(\mathbf{x},\mathbf{y})_{\mathbf{z}} = \frac{d(\mathbf{x},\mathbf{y})}{d(\mathbf{x},\mathbf{z}) + d(\mathbf{z},\mathbf{y})} \approx 1$$

where $d(\mathbf{x},\mathbf{y})$ means the Euclidean distance between two points $\mathbf{x}$ and $\mathbf{y}$.

Fig. 1 shows that $(\mathbf{x},\mathbf{y})$ are nearly opposite points with respect to $\mathbf{z}$.

Note that if $\mathbf{x}$ is an opposite point of $\mathbf{y}$ then $\mathbf{y}$ is also an opposite point of $\mathbf{x}$ with respect to $\mathbf{z}$. $I(\mathbf{x}, \mathbf{y})_{\mathbf{z}}$ may be called the *degree of oppositeness* of $\mathbf{x}$ and $\mathbf{y}$ with respect to $\mathbf{z}$.

Consider a neighborhood D around $\mathbf{z}$. Let $I_{\mathbf{z}}$ be the average of $I(\mathbf{x}, \mathbf{y})_{\mathbf{z}}$; $\mathbf{x}, \mathbf{y} \in D$. If $I_{\mathbf{z}}$ has a small value then $\mathbf{z}$ should be a border point in the neighborhood. Intuitively, we make a threshold at $\frac{1}{2}$.

Definition 2: A point $\mathbf{x} \in S$ is said to be *border point* if the average value of the degree of oppositeness, $I_{\mathbf{x}} < \frac{1}{2}$.

Definition 3: A point $\mathbf{x} \in S$ is said to be *interior point* if the average value of the degree of oppositeness, $I_{\mathbf{x}} \geq \frac{1}{2}$.

The degree of oppositeness $I(\mathbf{x}, \mathbf{y})_{\mathbf{z}}$ satisfies the following properties.

1) $I(\mathbf{x},\mathbf{y})_{\mathbf{z}}$ is scale invariant.
2) $I(\mathbf{x},\mathbf{y})_{\mathbf{z}}$ is invariant under rotation and translation.
3) $I(\mathbf{x},\mathbf{y})_{\mathbf{z}} \in [0, 1]$ for all $\mathbf{x}$, $\mathbf{y}$ with respect to $\mathbf{z}$.
4) $I(\mathbf{x},\mathbf{y})_{\mathbf{z}} = 0$ iff $\mathbf{x} = \mathbf{y}$.
5) $I(\mathbf{x},\mathbf{y})_{\mathbf{z}}$ is symmetric. i.e. $I(\mathbf{x},\mathbf{y})_{\mathbf{z}} = I(\mathbf{y},\mathbf{x})_{\mathbf{z}}$ for all $\mathbf{x}$ and $\mathbf{y}$ with respect to $\mathbf{z}$.

Let m be the desired number of border points. The border point detection algorithm of a data set $S = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\} \subseteq \Re^q$ is as follows.

Algorithm BPD:

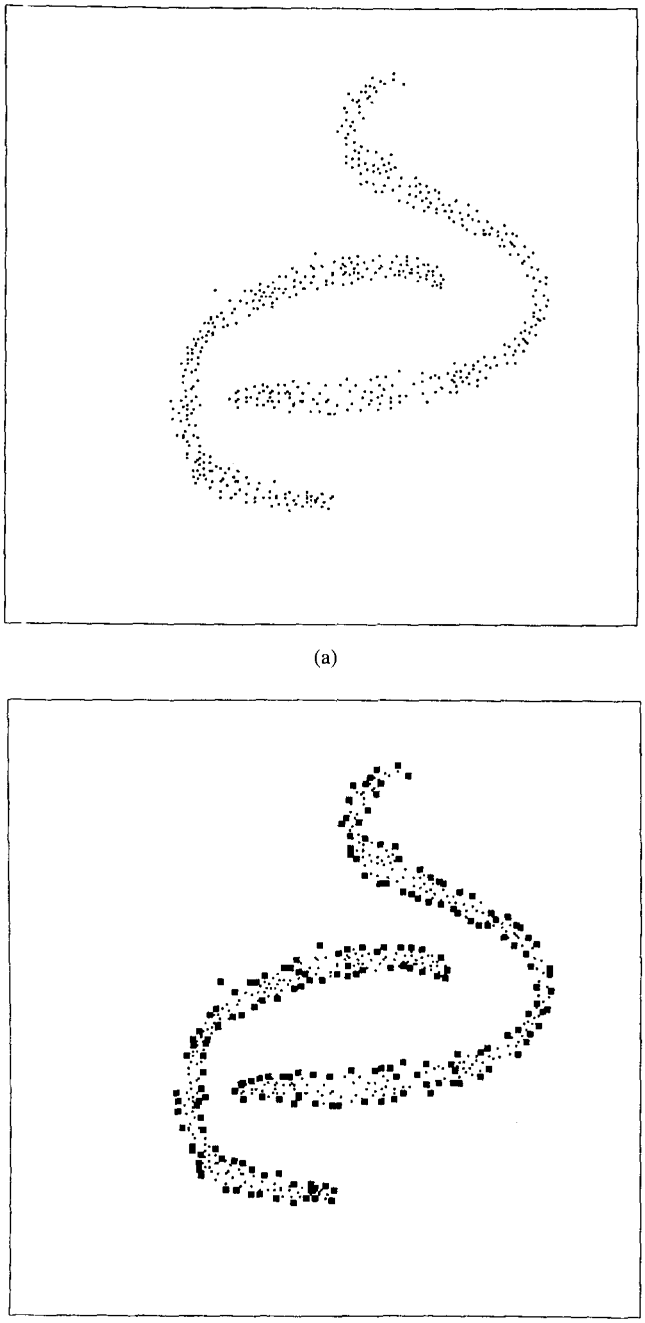
```
Step 1. Find the value of $I_{\mathbf{x}_i}$ for all $i = 1, 2, \ldots, n$.
Step 2. Rearrange the points according to the increasing order of
        their value of $I_{\mathbf{x}}$ provided, $I_{\mathbf{x}} < \frac{1}{2}$.
Step 3. Declare the first $k$ ranking points as $k$ border points, if
        they exist.
```

To test the efficiency of the border point detection (BPD) algorithm, several 2-D data were generated. Fig. 2(a) shows a nonconvex shape data. The border points of this data are marked by dark small squares

Manuscript received September 4, 1995; revised July 20, 1996.

The authors are with Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Calcutta 700 035, India (e-mail: bbc@isical.ernet.in).

Publisher Item Identifier S 1083-4419(97)05209-6.



(a)

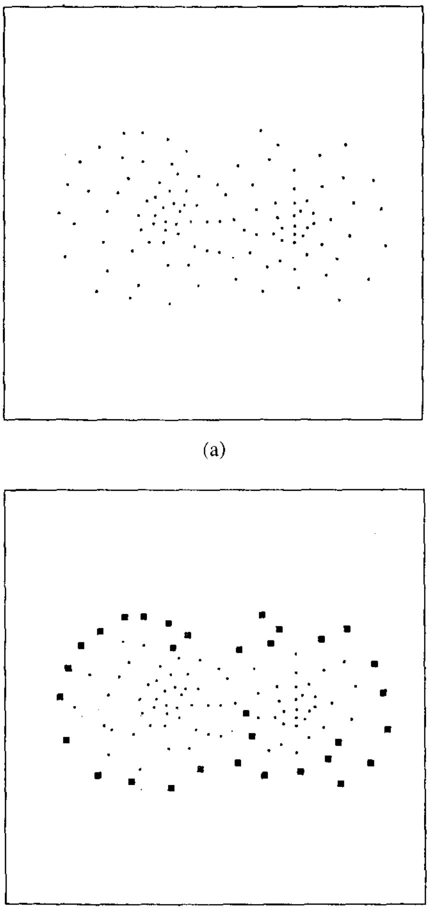
(b)

Fig. 2. Nonconvex shaped data. (a) A nonconvex shaped data of size 570. (b) Border points obtained by applying BPD algorithm.

[Fig. 2(b)]. Here k is nearest integer to 48% of the total number of data.

Another data, Fig. 3(a) shows an overlapping Gaussian cluster data of size 107. The border points are shown as Fig. 3(b). Here k is nearest integer to 30% of the total number of data.

The algorithm BPD is also useful for the basic idea about the shape of the dot pattern. If the dot pattern is of hyperspherical shape then the maximum and minimum of pairwise distances between the border points are almost equal. If the dot pattern is elongated or nonconvex then the difference between the maximum and minimum of pairwise distances is usually greater than some threshold value. So, for capturing any elongated or nonconvex cluster, we should have more than one seed point.



(a)

(b)

Fig. 3. Overlapping Gaussian cluster data. (a) An overlapping Gaussian cluster of size 107. (b) Border points obtained by applying BPD algorithm.

III. Multiseed Clustering Technique

Usually a clustering method has three main aspects namely, *initialization, cluster updating iterations* and *stopping criteria.* At the initialization stage, an initial partition is created and the (one or more) *seed* points corresponding to each cluster are defined. During each of the cluster updating iterations, the partitions as well as the positions of the seed points are modified. The stopping criteria determine when the execution of the method should end.

A. Initial Partitioning and Multiseed Assignment

The choice of the seed points depends on the shape of the clusters. Usually, the nonconvex or elongated cluster needs large number of seed points, while for convex and compact clusters one seed per cluster is sufficient to capture the data. The seed point selection problems may be understood as follows.

Consider a set of objects represented as point data in $\Re^q$ feature space. Given a set S of n data, we address the problem of selecting a small subset $V^\star \subset S$ of $n_0 \ll n$ data that *faithfully* represents the spatial organization of original data. The solution to this problem can find applications in data compression, data clustering [2], [9], [22], [23], pattern classification as well as statistical parameter estimation [7], [9]. For example, in clustering, many algorithms start with a few *seed* points, where each seed point represents the core of one cluster. The minimum distance classifier or the k-nearest neighbor classifier considers the seed points as the best patterns representing the classes. Quite often, a single *best representative point* is assumed to be the *mode* of the pattern set. *Density* and *mode* estimation are two classical problems in statistics related to the estimation of seed points. In many situations the problem of finding best representative points may be considered as a generalization of mode estimation and seed point detection problem.

In the literature of cluster analysis there exist several approaches of *seed* point estimation [2]. Macqueen [12] chooses the first K data units in data set as the initial seed points. Forgy [8] takes any desired partition of the data units into K mutually exclusive groups and compute the group centroids as seed points. Astrahan [1] computes the density for each data unit as the number of other data units within some specified distance, order the data units by density and choose the one with the highest density as the first seed point. The subsequent seed points are chosen in order of decreasing density, subject to the stipulation that each new seed point is at least a minimum distance away from all other previously chosen seed points. In another approach Ball and Hall [4] have suggested that the overall mean vector of data set is considered as the first seed point. The subsequent seed points are selected by examining the data units in their input sequence and accepting any data unit which is at least some specified distance, say d, from all previously chosen seed points. This process is continued until K seed points are accumulated or the data set is exhausted. Ling [11] suggests that (K, d)-cluster has the property that its elements are within a distance d of at least K other elements of the same cluster and the entire set can be marked by a chain of links each of length less than or equal to d. But there are no guidelines about how to choose K and d.

Chaudhuri *et al.* [6] suggested an approach for finding the seed points in plane that is dependent upon the local densities of the points. In this algorithm some outlier rejection was accomplished and the ordering was done in terms of cumulative density. Also, there are some guidelines about how to choose K and the distance d. But one of the drawbacks of the algorithm is that the chosen seeds are close to one another if the data is densely populated.

In this paper, a new parameter-free seed point detection approach is proposed where it is necessary to estimate the density at the data points. For this purpose we developed a kernel-based data driven density estimation procedure [17] which is briefly described below.

Consider the given set of points $S = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\} \subseteq \Re^q$. Suppose $F(\mathbf{y})$ is a Borel scalar function on $\Re^q$ such that

$$\sup_{y\in\Re^q} \mid F(\mathbf{y}) < \infty \qquad \int_{\Re^q} \mid F(\mathbf{y})d\mathbf{y} < \infty$$

$$\lim_{|y|\to\infty} |\mathbf{y}|^q F(\mathbf{y}) = 0 \qquad \int_{\Re^q} F(\mathbf{y})d\mathbf{y} = 1$$

where $|\mathbf{y}|$ denotes the length of the vector $\mathbf{y}$ on $\Re^q$. $F(\mathbf{y})$ is termed the *kernel* of the density estimator. Let $f(\mathbf{y})$ be the actual density function on $\Re^q$. The estimated density is defined as

$$f_n(\mathbf{x}) = \frac{1}{nh_n^q}\sum_{i=1}^{n} F\left(\frac{\mathbf{x}-\mathbf{x_i}}{h_n}\right) \qquad (1)$$

where $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}$ are independent and identically distributed random vectors following the density f and $\{h_n\}$ is a sequence of positive constants satisfying $h_n \to 0$ and $nh_n^q \to \infty$. We take the value of h_n be equal to

$$h_n = \left(\frac{\ell_n}{n}\right)^{\frac{1}{q}}$$

where q is the dimension ℓ_n is the sum of the edge weights of the minimal spanning tree (MST) [13], [14], [21] of the data set S; edge weight being the Euclidean inter-point distance. We take the kernel $F(\mathbf{x})$ as

$$F(\mathbf{x}) = \begin{cases} \frac{1}{2^q} & \text{if } |x_j - x_{ij}| \le 1 \quad \forall j = 1, 2, \ldots, q \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{x_i} = (x_{i1}, x_{i2}, \ldots, x_{iq})'$ and $\mathbf{x} = (x_1, x_2, \ldots, x_q)'$ and $'$ denotes the transpose.

So at the point $\mathbf{x}_i; i = 1, 2, \ldots, n$, let $A_i = \{\mathbf{y} : \|\mathbf{x}_i - \mathbf{y}\| \le h_n, \mathbf{y} \in S\}, i = 1, 2, \ldots, n$, the density of (1) can be expressed as

$$m_i = \frac{1}{2^q nh_n^q} \times \#A_i, \quad i = 1, 2, \ldots, n \qquad (2)$$

where $\#A$ means the number of points of the set A.

It has been proved in [17] that the estimated density is consistent and asymptotically unbiased.

Seed Point Detection:

Step 1: Find the radius h_n as discussed before.

Step 2: Compute the *density* for each datum $\mathbf{x}$ from (2).

Step 3: Find the border points of the data by applying BPD algorithm.

Step 4: Find the point where the maximum density occurs, say $\mathbf{x_1}^\star$. Count the number of points of the present cluster, say n.

Step 5: Let $\max_b$ and $\min_b$ be the maximum and minimum distances of the border points from the point $\mathbf{x_1}^\star$. Compute $\Delta f_b = \max_b - \min_b$. If $\Delta f_b \le \theta$ then go to Step 7. If $\Delta f_b > \theta$ then go to Step 6. (θ is the predefined threshold value.)

Step 6: Remove the nearest neighbor points of $\mathbf{x_1}^\star$ whose distances from $\mathbf{x_1}^\star$ are less than or equal to $\min_b$ from the present cluster. Let $S_1 = \{\mathbf{y} : \|\mathbf{x_1}^\star - \mathbf{y}\| \le \min_b\}$ and $m_0 = \#S_1$. Remove these m_0 points of S_1 from the present cluster. If $n - m_0$ is very small then go to Step 7 otherwise $n \leftarrow n - m_0$ and go to Step 4.

Step 7: Stop.

Depending on θ the algorithm SPD can automatically decide the number of seed points in the cluster. The parameter θ is a measure of circularity. If θ is zero the data is perfectly circular which is practically impossible. If θ is large, then one seed should take care of an elongated set of data. Thus, θ should be a small quantity but larger than the minimum *interpoint* distance in the data set. Let n_0 be the number of seed points in the cluster. The core (seed point) of the cluster is the mode, i.e., the highest density point. $\max_b$ and $\min_b$ are the maximum and minimum distances of the border points from the core (seed point). So, Step 5 of the algorithm SPD will decide whether the cluster is elongated or not. If the cluster is hyperspherical in shape then Δf_b will be almost equal to zero.

K-means type algorithm can capture the data of hyperspherical shape and any elongated or nonconvex cluster can be considered as the union of a few distinct hyperspherical clusters. So, to capture the data of an elongated cluster, our next task is to define the initial clustering and to label the seeds of each cluster. The following stages are proposed for the purpose.

1) Assign the data to the seed points by nearest neighbor rule, thus forming n_0 clusters, each cluster containing only one seed point.
2) Merge two or more clusters until K clusters are obtained.

The second stage contains two steps. In the first step a few candidate pairs out of ${}^{n_0}C_2$ cluster pairs are chosen. In the second step, only a few of these candidate pairs are merged. To find the candidate pairs, we generate a minimum spanning tree of n_0 seed points. A pair of clusters is considered for merging only if their seed points form an edge in the MST. The merging algorithm is as follows.

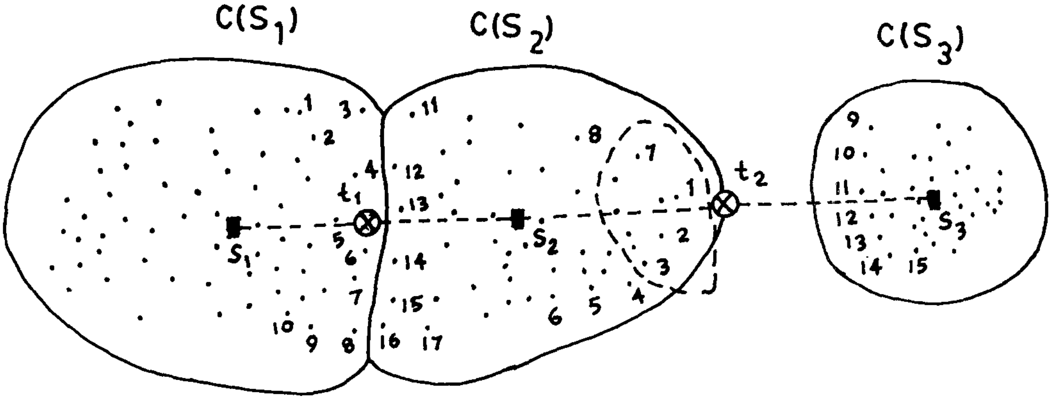


Fig. 4. Seed point assignment procedure.

Merging Technique [MT]:

Step 1: Find the number of points, N_0 of the two clusters which are to be merged. Let $C(\mathbf{x})$ and $C(\mathbf{y})$ be two clusters of the two seed points $\mathbf{x}$ and $\mathbf{y}$ which are to be merged and $N_0 = \#C(\mathbf{x}) \cup C(\mathbf{y})$.

Step 2: Find the midpoint, $\mathbf{z}$ of the two points $\mathbf{x}$ and $\mathbf{y}$.

Step 3: Find the border points by applying BPD algorithm of each $C(\mathbf{x})$ and $C(\mathbf{y})$ whose distances from $\mathbf{z}$ less than the distance between $\mathbf{z}$ and $\mathbf{x}$ or $\mathbf{y}$. Let $B(C(\mathbf{x}), C(\mathbf{y}))$ be the border points set such that

$$B(C(\mathbf{x}), C(\mathbf{y})) = \{\mathbf{X} : d(\mathbf{X}, \mathbf{z}) \leq d(\mathbf{z}, \mathbf{x})\}$$

where $d(\mathbf{x}, \mathbf{y})$ is the euclidean distance. Let $m = \#B(C(\mathbf{x}). C(\mathbf{y}))$.

Step 4: Find the border points of these m border points which belong to $C(\mathbf{x})$ and $C(\mathbf{y})$, respectively. Let i border points called $a_1, a_2, \ldots, a_i$ belong to $C(\mathbf{x})$ and remaining $j = m - i$ border points called $b_1, b_2, \ldots, b_j$ belong to $C(\mathbf{y})$.

Step 5: Find $m_0 = [p\% N_0]$. $[a]$ means the largest integer $\leq a$.

Step 6: For each $a_k, k = 1, 2, \ldots, i$ find m_0 nearest neighbors in $C(\mathbf{x}) \cup C(\mathbf{y})$. Let $n_k(\mathbf{x})$ and $n_k(\mathbf{y})$ be the neighbors of a_k coming from $C(\mathbf{x})$ and $C(\mathbf{y})$, respectively. Similarly for each $b_k, k = 1, 2, \ldots, j$, let $n'_k(\mathbf{x})$ and $n'_k(\mathbf{y})$ be the neighbors of b_k from $C(\mathbf{x})$ and $C(\mathbf{y})$, respectively.

Step 7: Find the value

$$Q(\mathbf{x}, \mathbf{y}) = \left| \sum_{k=1}^{i} n_k(\mathbf{y}) \Big/ \sum_{k=1}^{i} n_k(\mathbf{x}) + \sum_{k=1}^{j} n'_k(\mathbf{x}) \Big/ \sum_{k=1}^{j} n'_k(\mathbf{y}) - 2 \right|.$$

Step 8: Go to Step 1 until all candidate pairs, which are from an edge in the MST of the n_0 seed points, are considered.

Step 9: Order the edges in the MST in decreasing magnitude of the Q values. Go on deleting the edges corresponding to the top of the order list until K subtrees (a subtree can be a single node as well) are obtained.

In this algorithm MT, Step 3 to Step 7 are used to see whether the data of the border region of the two clusters are densely populated or not. In particular, for Step 6 that the ratios of $n_k(\mathbf{x})$ and $n_k(\mathbf{y})$, i.e., $\sum_{k=1}^{i} n_k(\mathbf{y}) / \sum_{k=1}^{i} n_k(\mathbf{x})$ and $n'_k(\mathbf{x})$ and $n'_k(\mathbf{y})$ i.e. $\sum_{k=1}^{j} n'_k(\mathbf{x}) / \sum_{k=1}^{j} n'_k(\mathbf{y})$ should be close to 1 if the two clusters $C(\mathbf{x})$ and $C(\mathbf{y})$ are to be merged. Thus, the merging criteria can be decided on the smallness of the quantity $Q(\mathbf{x}, \mathbf{y})$ (defined in Step 7). This Q value is assigned to the edge of the MST between the nodes $\mathbf{x}$ and $\mathbf{y}$.

Note that only p should be specified in this algorithm. Our experience is that $p = 10$ is a good choice. The choice of p depends more on the number of data than its dimensionality.

The procedure is explained through Fig. 4. Here the number of seeds $n_0 = 3$ and these three seeds are denoted as s_1, s_2 and s_3. Let the initial clusters corresponding to the seeds s_1, s_2 and s_3 be called $C(s_1), C(s_2)$ and $C(s_3)$, respectively. In Fig. 4 these clusters are enclosed by continuous lines. The MST of these seeds is given by the dashed lines. Now, the cluster pairs $(C(s_1), C(s_2))$ as well as $(C(s_2), C(s_3))$ are considered for merging because there are edges between s_1 and s_2 as well as between s_2 and s_3.

At first, consider $C(s_2)$ and $C(s_3)$ for possible merging. Let t_2 be the midpoint of the line $\overline{s_2 s_3}$. According to Step 3 of MT algorithm the border points are chosen from $C(s_2)$ and $C(s_3)$. Thus, eight border points numbered by $1, 2, \ldots, 8$ comes from the cluster $C(s_2)$ and seven border points numbered by $9, 10, \ldots, 15$ comes from $C(s_3)$. Here $m = 15$, $i = 8$ and $j = 7$. Now, the number of points of the union of two clusters is 83, i.e., $83 = \#C(s_2) \cup C(s_3)$. Consider 10% of the total, i.e., eight neighbors of each of the border points numbered $1, 2, \ldots, 15$ and check which cluster they are coming from (Step 5 and Step 6 of algorithm MT). For example, take the data numbered 1. We have to compute $n_1(s_3)$ and $n_1(s_2)$ where $n_1(s_k)$ is the number of its nearest neighbors (out of a total of eight) coming from $C(s_k)$, $k = 2, 3$. The nearest neighbors are enclosed by broken lines and it is seen that $n_1(s_3) = 0$ and $n_1(s_2) = 8$. In this way, contributions for all 15 borders are collected to get the value of $Q(s_2, s_3)$ as defined in the algorithm MT in Step 7.

The same procedure should be repeated for the pair (s_1, s_2). Here out of 17 border points of $C(s_1) \cup C(s_2)$, ten border points numbered by $1, 2, \ldots, 10$ and seven border points numbered by $11, 12, \ldots, 17$ comes from $C(s_1)$ and $C(s_2)$, respectively. Here $\#C(s_1) \cup C(s_2) = 110$. So $[10\%$ of $110] = 11$. Now for each of the border points, again, 11 nearest neighbors are used to compute $Q(s_1, s_2)$. It can be found that $Q(s_1, s_2) > Q(s_2, s_3)$. So, the edge between s_2 and s_3 is deleted and two clusters are obtained. In one cluster there are two seeds while in the other there is only one seed. This cluster is subject to cluster updating iteration described below.

B. Cluster Updating

Now we consider the cluster updating iterations. Each of the iterations i) redistributes the data and ii) updates the positions of seed points of each cluster. Suppose that the number of seed points in the ith cluster C_i is n_i. The distance of $\mathbf{x}$ from the cluster C_i

is defined as the minimum of the Euclidean distances of $\mathbf{x}$ from the n_i seeds of C_i. During the redistribution phase, $\mathbf{x}$ is assigned to C_i if its distance from C_i is smaller than its distance from any other cluster.

To update the seed point position, the data in the cluster C_i is distributed among its n_i seeds so as to generate n_i subclusters. The centroid of each of these subclusters denotes a new seed.

C. Stopping Criterion

The cluster updating process is stopped if either a prespecified number of iterations are completed, or the clustering results between two consecutive iterations do not change significantly.

IV. Experimental Results

To test the efficiency of the algorithm, several multidimensional random data clusters were generated. We demonstrate the results on 2-D data only. Fig. 5 contains two clusters, one is elongated and another is compact. Seven seed points are automatically found by our SPD algorithm. The seed points (marked by dark squares) and clusters around them numbered $1, 2, \ldots, 7$ (marked by dashed circular arc) are shown in Fig. 5(a). The MST generated by the seed points are also shown in Fig. 5(a). Some edges of the MST are deleted using MT algorithm. The Q value for clusters pair numbered 4 and 7 is greater than the Q values for all other cluster pairs defined by the MST. So, the edge between the seeds of clusters numbered 4 and 7 is disconnected. Clusters 1–4 are merged together to form one cluster whereas clusters 5–7 are merged together to form another cluster. Note that one cluster now contains four seeds and the other contains three seeds. Fig. 5(b) shows the final clusters of the data after a few iterations of multiseed clustering.

Fig. 6(a) shows another data where the number of clusters is 2. The seed points and clusters around them are numbered $1, 2, \ldots, 16$. The MST of the seeds are also drawn in Fig. 6(b). The edge between seeds of cluster 1 and 8 is deleted, leading to two clustering. Fig. 6(c) shows the final clusters of the data after a few iterations of multiseed clustering.

Another pattern is shown in Fig. 7(a). Fig. 7(b) shows the corresponding seed points and clusters around them numbered by $1, 2, \ldots, 9$. The MST generated by the seeds are also shown in Fig. 7(b). Here edge between the seeds of cluster numbered 1 and 6 is disconnected. Out of two clusters, one is formed by merging clusters numbered 2–7 and another by merging clusters numbered 1, 8, and 9. The final results are shown in Fig. 7(c).

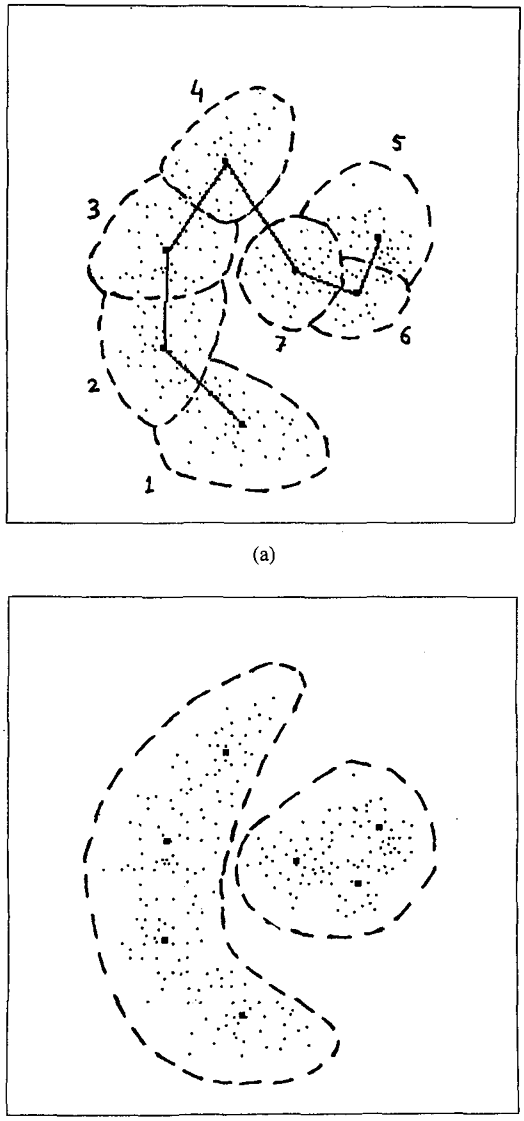
V. Discussion

The problem of multiseed clustering technique for a nonconvex and elongated pattern is considered in this paper. K-means type of clustering techniques are the techniques of one seed one cluster and a single seed point cannot correctly reflect the nature of the data of an elongated and nonconvex shape. To capture the data belonging to an elongated or nonconvex cluster, we have considered more than one seed point in the cluster.

The computational complexity of the algorithm is as follows. The maximum computational burden is on pairwise distance computation which is a $O(n^2)$ algorithm in the worst case. Detection of border points is a $O(n)$ while density estimation is a $O(n \log n)$ procedure. Each seed can be computed in $O(n)$ time. The clustering algorithm complexity is similar to that of standard K-means algorithm. All computations are linear on dimensionality, increasing as the computation of Euclidean distance increases with dimensionality.

The method can be extended to the case when the number of clusters K is unknown. To do so, the MST of the seed points is generated and the Q value for any edge of the MST is computed by Step 7 of MT algorithm. If for two seed points s_1 and s_2 the Q value exceeds a predefined threshold T then s_1 and s_2 should belong to two different clusters and the edge between s_1 and s_2 in the MST is deleted. The number of subtrees obtained in this way is the desired number of clusters. The split and merge criterion proposed in this method can be applied to other clustering techniques as well.

It is possible to generalize the approach so that splitting and merging is activated in each iteration. To do so, at each iteration the MST of the seed points should be computed after the new positions of the seed points are found. Next, using the Q values of the edges of the MST, the seed points are reallocated among the K clusters. Then the iteration is completed and a new iteration starts.



(a)

(b)

Fig. 5. The combination of an elongated and a compact cluster. (a) The initial seed points, individual clusters, and the MST generated by the seed points. (b) The final output.

Acknowledgment

The authors wish to thank S. Chakraborty for his drawings.

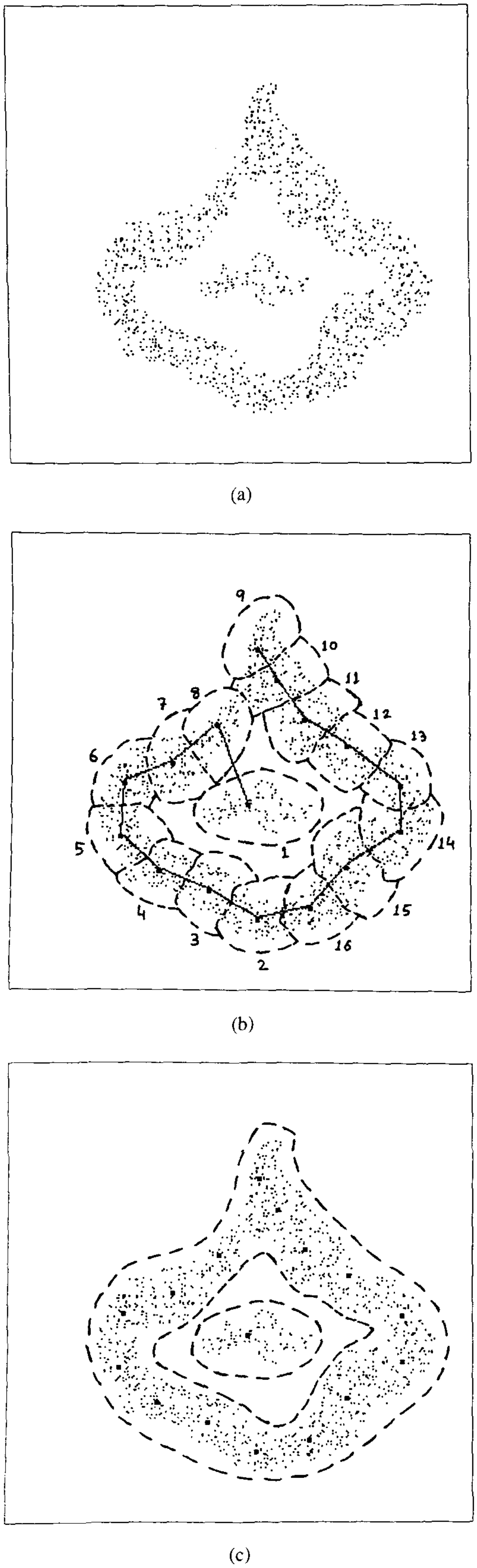


Fig. 6. Two clusters of a complex shape. (a) Original data. (b) The initial seed points, individual clusters and the MST generated by the seed points. (c) The final output.

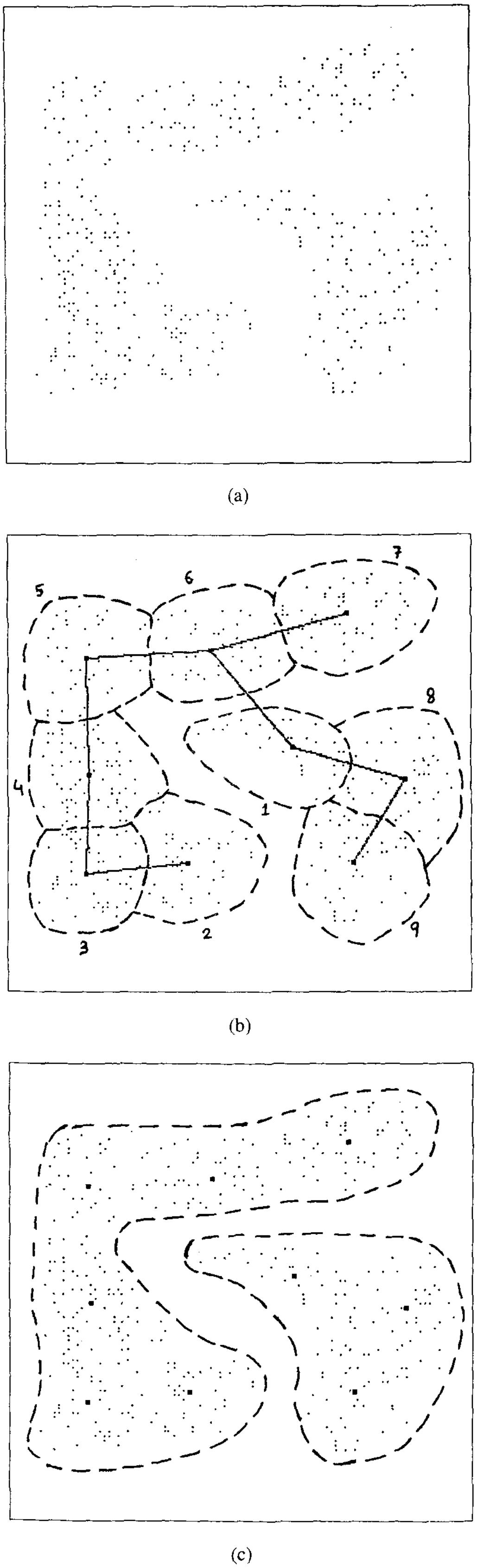


Fig. 7. Two uniform distribution data of complex shape. (a) Original data. (b) The initial seed points, individual clusters and the MST generated by the seed points. (c) The final output.