A DYNAMIC APPROACH FOR CLUSTERING DATA *

J. A. García,  J. Fdez-Valdivia, F. J. Cortijo and R. Molina
Departamento de Ciencias de la Computación e I.A.
E.T.S. de Ingeniería Informática
Universidad de Granada. 18071 Granada. Spain.

*Corresponding author: J.A. García
Departamento de Ciencias de la Computación e I.A.
E.T.S. de Ingeniería Informática.
Universidad de Granada. 18071 Granada. Spain.

Key word and phrases: dynamic scheme of clustering, non-piecewise linear separable clusters, dissimilarity measure, first derivative.

Number of manuscript pages: 33
Number of figures: 15

*This work has been supported by the Spanish Dirección General de Ciencia y Tecnología (DGCYT)
under grant PB92-0925
Abstract

This paper introduces a new method for clustering data using a dynamic scheme. An appropriate partitioning is obtained based on both a dissimilarity measure between pairs of entities as well as a dynamic procedure of splitting. A dissimilarity function is defined by using the cost of the optimum path from a datum to each entity on a graph, with the cost of a path being defined as the greatest distance between two successive vertices on the path. The procedure of clustering is dynamic in the sense that the initial problem of determining a partition into an unknown number of natural groupings has been reduced to a sequence of only two class splitting stages. Having arisen from any particular application, the proposed approach could be effective for many domains, and it is especially successful to identify clusters if there is lack of prior knowledge about the data set. The usefulness of the dynamic algorithm to deal with elongated or non-piecewise linear separable clusters as well as sparse and dense groupings is demonstrated with several data sets.

1 Introduction

An algorithm of clustering of existing natural groupings in the data set, which was not driven by applications, should be able to deal with several problems: (a) to assure an optimal number of clusters or partitions in a given data to overcome the lack of knowledge about such a number (e.g., Xu et al. (1993)), (b) to circumvent the lack of prior knowledge of size of clusters in the data set (e.g., Milligan et al. (1983)), (c) to avoid the dependence of clustering on the initial cluster distribution (e.g., Zhang (1991)), (d) to find elongated and non-piecewise linear separable clusters, as well as to identify dense and sparse ones (e.g., Yin et al. (1994)), (e) to be robust in many complicated situations (e.g., García et al. (1994b)), and (f) to capable of quickly handling large data rates (e.g., Dubes et al. (1980)).

The objective of clustering is to partition entities into groups isolating any underlying structure, which requires a minimum level of homogeneity within a cluster (e.g., Toet (1991)). There are many approaches to accomplish this objective, categorized into two main types: hierarchical and partitional. Because of computational requirements of typical hierarchical methods (e.g., Jain et al. (1988)), as well as the requirement of prior knowledge of partitional methods about the number of clusters and the minimum level of homogeneity
for cluster (e.g., MacQueen (1967); Hartigan (1975); Gnanadesikan (1977); Kaufman et al. (1987)), both hierarchical and partitional methods have limited applicability.

A hierarchical cluster analysis provides a dendrogram consisting of layers of nodes, each representing a cluster. Cutting such a dendrogram horizontally at any level defines a clustering and identifies clusters. But a very illuminating example given in Jain and Dubes (1988) demonstrates the difficulty in comparing two dendrograms and motivates the development of methods for automatically isolating significant clusters. Also, examples can be found in Jain and Dubes (1988) demonstrating the difficulties inherent in letting the human eye scan over the dendrogram to pick out reliable clusters, as well as that the hierarchical structure can change dramatically with small changes in the rank orders of the proximities. Nevertheless, the main problem with the hierarchical methods remains in the difficulty of deriving quantitative measures of the quality of clusters to help answer such questions as: where is the best cutting level to define a partition ?. To define hierarchical clustering methods can be used properties others than connectedness and completeness (e.g., Hubert (1974)). But certain practical difficulties arise when trying to select a suitable property, and few guidelines exist other than intuition and experience.

The idea presented in this paper is related to the single linkage hierarchical clustering concept in the sense that the distance between two clusters is defined as the smallest distance between two of their points, or phrasing it in a different way, points belong to the same cluster if you can walk from one to the other by sufficiently small steps. The interesting point is of course, what is "sufficiently small". Exactly this point is analyzed in Section 3.1. But the proposed clustering method differs from the single linkage concept in the following sense: the partitioning is obtained using a dynamic procedure of splitting where the initial problem of determining a partition into an unknown number of groupings has been reduced to a sequence of stages of splitting into only two classes. However, what is important and new about our approach is that, at each stage of the dynamic processing, one significant cluster is automatically isolated by using a proper measure of the strength of the cluster formed at that stage. In the section of experimental results and discussion, it will be make clear in what conditions this might be advantageous.

This paper introduces a new algorithm for cluster analysis having arisen from any par-
ticular application. The approach presented could be effective for many domains: character recognition (e.g., Chan et al. (1992)), military surveillance (e.g., Brown et al. (1991)), boundary simplification in Cartography (e.g., García et al. (1994a)), unsupervised estimation of surface and texture models (e.g., Silverman et al. (1988)), and so on. Besides the approach presented could be an useful way to circumvent the problems of clustering described above.

In the proposed clustering scheme, the following very realistic assumption to many applications is made: the data set has several separable clusters, e.g., elongated and non-piecewise linear separable groupings of arbitrary shape, dense and sparse natural clusters, and each of them can be isolated in a different part of the space in which the data set is defined. Hence the problem of detecting groupings which reveal any underlying structure will be solved by exploiting two key features: the first, given an entity \( X_{ij} \) picked randomly from the data, the ordered version of a proper dissimilarity measure from \( X_{ij} \) to all the other entities \( X_k, d_{ij}(k) \), provides us with the overall shape of the data from the point of view of \( X_{ij} \), see Figures 2 and 4; and the second, the entities assigned the same cluster of the point \( X_{ij} \) can be isolated at a single valley (the first one) of the first derivative of \( d_{ij} \), making the assumption that such derivative is computed in such a way that overlooks the inter-cluster differences as well as removes the intra-cluster differences. In allowing the clustering to be reformulated as a sequence of simpler problems, the previous two key features can be employed to define a dynamic scheme of partitioning.

The problem of clustering the given data set \( X = \{X_1, \ldots, X_n\} \) into an unknown number \( N \) of clusters \( C_0, C_1, \ldots, C_{N-1} \), can be reduced to a sequence of stages of simpler partitioning. In each stage, a subset \( X^j \) of \( X \) to be clustered into only two groupings. A natural cluster \( C_j \) containing all the data points in \( X^j \) which are assigned the same class of a given point \( X_{ij} \), with \( X_{ij} \) being picked randomly from \( X^j \), will be one class. The data in the subset \( X^j \) not absorbed in the cluster \( C_j \), will be the other.

The partitioning produced by our clustering algorithm is therefore based on only two central concepts: (a) the dissimilarity function to measure the closeness between the entities to be clustered, as well as (b) the dynamic procedure of splitting. These concepts necessary for understanding our approach are introduced in the following sections. In Section 2, the
ordered dissimilarity measure is defined. In Section 3, the dynamic procedure of clustering is presented, and also the algorithm of clustering by using such dynamic approach is given. Experimental results and discussion are given in Section 4. Conclusions are summarized in Section 5.

2 A convenient measure of dissimilarity

The choice of a dissimilarity function from a point of view to all the other entities determines the type of structure produced by the clustering algorithm. Because of a standard Euclidean distance fails to deal with elongated or non-piecewise linear separable clusters (e.g., Yin et al. (1994)), a dissimilarity function from the viewpoint location $X_j$ to all the other entities $X_k$ in any subset $X^j$ chosen from the data $X$ need to be introduced.

We define a graph $G^j = (X^j, U^j)$ corresponding to the data subset $X^j$ in the given data $X$, with $U^j$ being the set of arcs $u = (k, l)$ from $X_k$ to $X_l$, a pair of points in $X^j$. We associate with each arc $u \in U^j$ a real number $l(u) \geq 0$, and if $u = (k, l)$, we shall equally well use the notation $l_{kl}$ for $l(u)$.

Let $l_{kl}$ be the distance from $X_k$ to $X_l$ defined as

$$ l_{kl} = \sum_{i=1}^{p} (X^j_k - X^j_l)^2 $$

where $X_k$ is the vector $(X^1_k, \ldots, X^p_k)^t$, for all $k$, with a superscript $t$ to a vector denoting vector transpose.

Let $\mu(i, j, k)$ be a set of arcs constituting a path between two vertices $X_i$ and $X_k$, and $l(\mu)$ be the cost of $\mu(i, j, k)$ from $X_i$ to $X_k$ defined as follows:

$$ l(\mu) = \max\{l(u) \mid u \in \mu(i, j, k)\} $$

The optimum path between two data points $X_i$ and $X_k$ is a path $\mu^*(i, j, k)$ from $X_i$ to $X_k$ whose maximum cost $l(\mu^*)$ is minimum:

$$ \mu^*(i, j, k) = \text{Arg}_\mu \min \{\max\{l(u) \mid u \in \mu(i, j, k)\} \} $$

Hence, the dissimilarity from the viewpoint $X_i$ to all $X_k$, is defined by using the cost of the optimum path $\mu^*(i, j, k)$ from $X_i$ to each $X_k$:

$$ d^{ij}(X_i, X_k) = l(\mu^*) = \max\{l(u) \mid u \in \mu^*(i, j, k)\} $$
with $\mu^*$ being the optimum path between $X_{i_j}$ and $X_k$.

Since our interest is to consider only an ordered sequence of dissimilarity measures, the set

$$\{d^{G_i}(X_{i_j}, X_i) \mid X_i \in X^j\}$$

is ordered to obtain a new function $d_j(k) = d^{G_j}(X_{i_j}, X_{i_k})$ where

$$d_j(k) \leq d_j(k + 1), \quad \text{with } X_{i_k}, X_{i_{k+1}} \in X^j.$$ 

Figures 2 and 4 show the plot of $d_j(k)$ against $k$, on the two data sets given in Figures 1 and 3 respectively.

2.1 An algorithm for obtaining the optimum path from a viewpoint to all the data points

The problem of the optimum path between two data points $X_1$ and $X_i$, for all $i \in \{1, \cdots, n\}$, is that of finding a path $\mu(1, i)$ from $X_1$ to $X_i$ whose maximum cost

$$l(\mu) = \max\{l(u) \mid u \in \mu(1, i)\}$$

is minimum. In our problem, because of $l(u)$, with $u = (X_k, X_l)$, it is interpreted as defined in (1), the characteristics of the graph are as follows:

$$l(u) \geq 0, \quad \text{for all } u \in U.$$ 

The problem considered, is therefore to find the optimum path from one vertex to all the others. A modified version of the algorithm of Moore and Dijkstra (e.g., Moore (1957); Dijkstra (1959)) can be used to compute such a path from a vertex (numbered 1) to all the others.

Set $X = \{1, \cdots, n\}$. Let $l_{ij}$ be the distance from $X_i$ to $X_j$, if $(i, j) \in U$, as defined in (1).

Algorithm 1:

(a) Start:

$$\overline{S} = \{2, \cdots, n\}; \quad \pi(1) = 0, \quad \pi(i) = l_{ii}, \text{ for all } i \in \overline{S}.$$
(b) Find \( j \in \mathcal{S} \) such that

\[
\pi(j) = \min_{i \in \mathcal{S}} \pi(i)
\]

Set \( \mathcal{S} \leftarrow \mathcal{S} - \{j\} \).

If \( |\mathcal{S}| = 0 \), End; otherwise go to step (c).

(c) For all \( i \in \mathcal{S} \) set

\[
\pi(i) = \min\{\pi(i); \max(\pi(j), l_{ji})\}
\]

go to step (b).

If \( j \in \mathcal{S} \), then \( \pi(j) \) is the cost of an optimum path from 1 to \( j \). Stage (c) means adjusting the value of \( \pi(i) \), for all \( i \in \mathcal{S} \), to take into account the fact that \( j \) is now in \( \mathcal{S} \).

The algorithm works in \( O(M) + O(n^2) \) time, with \( M \) being the number of arcs, \( M = |U| \). In our graph, where \( M \) is of order \( n^2 \), the second term \( O(n^2) \) is acceptable, and therefore we conclude that the algorithm requires \( O(n^2) \) time.

3 A dynamic procedure of clustering

Our scheme for clustering data, it is dynamic in the sense that the initial problem of determining a partition into an unknown number of clusters will be reduced to a sequence of two class splitting stages.

Let \( X^j \) be a subset of data not absorbed in any of the existing clusters, \( C_0, C_1, \ldots, C_{j-1} \), at the stage \( j \) of the dynamic processing; with \( X^0 \) being the given data set, \( X^0 = X \).

At stage \( j \), \( X^j \) is divided into only two classes: \( C_j \) and \( X^j - C_j \). The dynamic process of clustering is stopped at stage \( j \), if the class \( X^j - C_j \) is the empty set, \( X^j - C_j = \emptyset \). Otherwise, the process progresses, and the subset \( X^{j+1} \) to be partitioned at the stage \( j + 1 \), it will be the one defined as \( X^{j+1} = X^j - C_j \). The subset \( X^{j+1} \) to be processed is therefore the data still not placed in any existing cluster at stage \( j + 1 \).

The process of placing all the entities from data subset \( X^j \) into two classes is accomplished as follows.
3.1 A simpler problem of splitting into two classes

Let $X_{ij}$ be a datum picked randomly from $X^j$, which can be thought of as a seed point for a new cluster $C_j$ to be formed at the stage $j$. The cluster $C_j$ grows because of absorbance of patterns from $X^j$. It may be understood that a datum is absorbed if it is very ‘close’ to $X_{ij}$, where the meaning of ‘closeness’ need to be defined.

Let $G_j = (X^j, U^j)$ be a graph corresponding to subset $X^j$, and $d_j$ be the ordered dissimilarity measure defined as described in section 2. The problem of determining a new natural cluster, $C_j$, in which the point $X_{ij}$ is placed, is performed taking account of the two key features: (a) the first, the ordered version of a proper dissimilarity measure from $X_{ij}$ to all the other entities $X_k$ from $X^j$, $d_j$, provides us with the overall shape of the data subset $X^j$ from the point of view of $X_{ij}$; (b) the second, the entities assigned the same cluster of the point $X_{ij}$ can be isolated as a single valley (the first one) of the first derivative of $d_j$, where the first derivative $d'_j$ is computed by using a Gaussian filter with a proper scale parameter $\sigma$, overlooking the peaks from the inter-cluster dissimilarities, as well as removing the detail from the intra-cluster dissimilarities.

To locate the derivative’s peaks that the inter-cluster differences produce, the unwanted detail from intra-cluster differences must be removed by smoothing, but how much smoothing should be performed? The derivative should be processed at the scale which best describes each shape of interest, the peaks that the inter-cluster differences produce, while eliminating noise and many spurious peak of derivative which the intra-cluster differences produce. If the function $d'_j$ is processed at such a scale then the right peaks can be extracted without any confusion from unwanted detail.

3.1.1 A proper scale to process the derivatives

We will use Gaussian smoothing to obtain a representation of the function $d'_j$ at a specific scale. To smooth such a function, it must be convolved with an one-dimensional Gaussian Kernel $G_\sigma(t)$ of standard deviation $\sigma$, using the technique described in Mokhtarian et al. (1986).

The function $d'_j$ is smoothed over a range of six separated scales, with $\sigma$ starting at 4, to guarantee a certain amount of smoothing, and increasing by a factor of $\sqrt{2}$. 


We will choose a normalized measure of the zeros of curvature of $d'_j$ at a scale $\sigma$

$$Z_{\sigma}(C_i) = \sigma \sum_{l \in \mathcal{D} \cap y(x_l, y_l) \in C_i} z(k_l),$$

with

$$z(k_l) = \begin{cases} 1 & \text{if } k_l = 0 \\ 0 & \text{otherwise} \end{cases},$$

as a significance measure describing fundamental features of the function’s shape (Rosin (1992)), with $k_l$ being any curvature value on the function at such a scale.

To find out a single scale for the derivative, we choose the scale minimizing this measure. Minima of this significance measure over different scales correspond to structure in the function having been isolated at its natural scale. A discussion about this measure can be found in Rosin (1992).

### 3.1.2 A criterium for automatically isolating significant clusters

A function $equiv_j(k)$ is constructed for deciding whether a point $X_k$ from $X^j$ is in the same class of $X_{i_j}$ (that is, it is close to $X_{i_j}$)

$$X_k \in C_j$$

or in a different class

$$X_k \in X^j - C_j.$$

Let $equiv_j$ be defined as:

$$equiv_j(k) = \begin{cases} 1, & \text{if } d^{G_j}(X_{i_j}, X_k) \leq d_j(firstpeak) \\ 0, & \text{otherwise} \end{cases}$$

with $d^{G_j}(X_{i_j}, X_k)$ being the cost of the optimum path from $X_{i_j}$ to $X_k$, and where $firstpeak$ denotes the first value in which the derivative $d'_j$ reach a peak. That extreme is chosen according to the simple rule of selecting the first point such that it is the extreme in a neighborhood centered upon it. Hence, the function $equiv_j(k)$ returns a nonzero value if the values of the first derivative $d'_j$ at $X_k$ and $X_{i_j}$ are not separated by a peak, or equally well if the value of the derivative $d'_j$ at $X_k$ is into the first valley of such a derivative.
If the function $equiv_j(k)$ returns a nonzero (true) value, the points $X_{ij}$ and $X_k$ are assigned the same cluster $C_j$; otherwise they are not.

Since one cluster $C_j$ is obtained after each run, the proposed scheme need to be reapply to the data subset $X^j - C_j = X - \bigcup_{i=0}^{j} C_i$ in which the other clusters must be implicit. As declared above, the clustering procedure progresses until no further clusters can be found, what happens if at a stage $N - 1$ the ordered dissimilarity measure $d_{N-1}$ shows no peak in its first derivative. That means only one cluster, $C_{N-1}$, being implicit in the subset $X^{N-1}$. Hence, $X^N = \emptyset$. The stopping rule suggested here, it is objective in the sense that it assumes no prior knowledge about the given data set.

The dynamic clustering algorithm is given in the following.

**Algorithm 2:**

(a) Start:
\[
\text{Set } j = 0; \quad X^0 = X
\]

(b) Choose randomly a point $X_{ij}$ from $X^j$.

(c) For all $X_k \in X^j$, If $equiv_j(k)$ is true (nonzero), Set $C_j = C_j + \{X_k\}$.

(d) Set $X^{j+1} = X^j - C_j$;
\[
\begin{align*}
\text{If } X^{j+1} = \emptyset, \text{End; otherwise go to step (e).}
\end{align*}
\]

(e) Set $j = j + 1$;
\[
\text{go to step (b).}
\]

4 Experimental results and discussion

In this section, to examine the effectiveness of the method proposed here, its performance has been evaluated for a number of pattern sets showing the differences with existing hierarchical and partitional techniques.

First, two comparative results of the K-means algorithm and the proposed dynamic scheme are presented on elongated and non-piecewise linear separable cluster as well as sparse and dense ones. More precisely, the results of clustering by using the proposed
algorithm are shown in Figures 1 and 3. To examine the usefulness of the stopping criterion in the dynamic scheme in order to assure an optimal number of cluster in a given data, its performance has been evaluate for a number of data sets, even though only two-dimensional results are presented. On the other hand, the performance of the K-means algorithm is shown in Figures 5 and 6 on the same data sets.

In Figure 1(A), there are two elongated and non-piecewise linear separable clusters. The proposed method automatically identified these two clusters, see Figures 1(B) and 1(C). On the contrary, since each data point is assigned to the grouping with the nearest center, the K-means algorithm fails to separate the two clusters, see Figures 5(B) and 5(C).

In Figure 3(A), there are four different collections of objects: a sparse cluster and a dense grouping in addition to a pair of non-globular clusters. The proposed dynamic algorithm successfully separated the four different clusters, see Figures 3(B), 3(C), 3(D), and 3(E), showing the robustness of our method in these complicated cases. Figure 6, shows that our dynamic approach is also superior to the K-means method in this example: the K-means algorithm fails to identify these four natural clusters.

Although the computation of the proposed algorithm is more expensive than that of the K-means method, the application of the dynamic scheme reveals the intrinsic structure of the input data, having excellent results to find natural clusters if there is lack of knowledge about the data.

In Figures 2 and 4, the ordered dissimilarity measure proposed as well as its first derivative to several stages of splitting are shown on the two considered examples. Thus, Figure 2(A) shows the plot of such ordered dissimilarity measure at a stage 0, \( d_0 \), and Figure 2(B) its first derivative, on the data given in Figure 1(A). On the other hand, Figures 4(A), 4(C), and 4(E) show the plot of \( d_j \) at the stages \( j = 0, 1, 2 \), respectively, on the data set in Figure 3(A). Figures 4(B), 4(D), and 4(F) show the plot of the first derivative of \( d_j \) at the stages \( j = 0, 1, 2 \), respectively.

Second, the comparative performances of the agglomerative hierarchical clustering and the proposed method are presented. The major steps in agglomerative clustering are con-
tained in the following procedure (Duda and Hart (1973)):

Algorithm: Basic Agglomerative Clustering

(a) Start:

Set \( c^* \leftarrow n \); \( C_i \leftarrow \{X_i\}, i = 1, \ldots, n \).

(b) If \( c^* \leq n \), stop.

(c) Find the nearest pair of distinct clusters, say \( C_i \) and \( C_j \).

(d) Merge \( C_i \) and \( C_j \), delete \( C_j \), and decrement \( c^* \) by one.

(e) Go to step (b).

To measure the distance between two clusters, we restrict our attention to the following distance measures:

\[
d_{\text{max}}(C_i, C_j) = \max_{x \in C_i, x' \in C_j} \|x - x'\| \tag{7}
\]

and

\[
d_{\text{avg}}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{x' \in C_j} \|x - x'\| \tag{8}
\]

We shall use the two-dimensional point sets showed in Figure 1 and 3 to illustrate some of the differences with our technique. Results of the basic agglomerative clustering using \( d_{\text{max}} \) (the furthest-neighbor algorithm) and \( d_{\text{avg}} \) are given in Figure 7 and Figure 8, respectively. As Figure 7 and Figure 8 illustrate, when \( d_{\text{max}} \) or \( d_{\text{avg}} \) is used to measure the distance between subsets, the growth of elongated clusters is discouraged, and the resulting groupings can be meaningless. This is another example of imposing structure on data rather than finding structure in it.

Third, comparative results of the stepwise-optimal hierarchical clustering with our method are presented. The pattern sets in Figure 1 and 3 are used to illustrate some of the differences. In the stepwise-optimal procedure (e.g., Duda and Hart (1973)) the pair of distinct clusters \( C_i \) and \( C_j \) whose merger would increase the criterion function as little as possible can be selected as the pair for which the distance

\[
d_c(C_i, C_j) = \sqrt{\frac{n_i n_j}{n_i + n_j} \|m_i - m_j\|} \tag{9}
\]
is minimum, with \( m_i = \frac{1}{n_i} \sum_{x \in C_i} x \) and \( n_i = |C_i| \). This assures that at each iteration we have done the best possible thing, but as Figure 9 illustrates, it does not guarantee that the final partition is optimal.

Results of the agglomerative hierarchical clustering as well as the stepwise-optimal procedure for the pattern set given in Figure 1 are showed in Figure 10. More precisely, Figures 10.A and 10.D are results of the stepwise-optimal procedure; results of the basic agglomerative hierarchical clustering using \( d_{avg} \) are showed in Figures 10.B and 10.E; as well as Figures 10.C and 10.F are results of the basic agglomerative hierarchical clustering using \( d_{max} \).

Also, for the purpose of experimentation with the proposed technique, we have used an image of one artificial machine part taken under controlled lighting conditions in a laboratory. Some of the detected edge points are showed in Figure 11 (in the top left). As illustrates Figure 11, the proposed dynamic clustering tends to group edge points into meaningful or coherent clusters without relying on prior knowledge of the scene contents. Such a result may yield to the fast identification of distinct low-level perceptual structures entities which can be processed independently and used to speed up object recognition.

Comparative results of the k-means, stepwise-optimal hierarchical clustering, and the basic agglomerative clustering using both \( d_{max} \) as well as \( d_{avg} \), with our method are presented in Figs. 12,13,14 and 15, respectively. As can be noted, all of them fail to find the natural groupings that the scene contains.

The proposed scheme has several advantages. It can automatically determine the number of clusters. It is also capable of detecting dense and sparse clusters, as well as elongated or non-piecewise linear separable groupings. There is no necessity of any initial cluster distribution. It is capable of dealing with large and small clusters. The computational cost of the proposed algorithm is bounded by \( O(Nn^2) \), with \( N \) being the number of the clusters, and with \( n \) being the number of given data points. Its main computational burden is the computation of the function \( equiv \). Thus, although K-means algorithm is more efficient, if there is lack of prior knowledge about the data set, our dynamic approach is preferable. The proposed algorithm is especially successful for clustering large data set having a small number of clusters.
Several geometric structures or graphs (e.g., minimum spanning trees, relative neighborhood graphs, and Gabriel graphs) for analyzing patterns have provided useful algorithms which can identify irregularly shaped or nonglobular clusters. A graph can be constructed whose nodes represent the patterns to be clustered and whose edges represent relations between the nodes. Zahn (1971) utilized a minimum spanning tree to detect clusters, but the greatest deficiency of this approach is that prior knowledge of the shapes of the clusters is needed to select the proper heuristic to identify inconsistent edges. In the proposed approach, a cluster can have any shape. Thus, clusters can be linelike or curvelike; they need not be convex or even simply-connected (e.g., they can be annular). However, what is advantageous of our method is that no knowledge is needed about the cluster’s shape, and hence the natural groupings will be obtained in the data even though there is no prior knowledge of the shapes of the clusters. Also, the proposed clustering method might be advantageous when the very reason for applying cluster is to estimate those shapes and their structure.

5 Conclusions

In this paper, a new method for clustering data using a dynamic scheme was introduced. It was shown how an appropriate partitioning can be obtained based on both a dissimilarity measure as well as a dynamic procedure of splitting, making the only assumption that the data set has several separable clusters, even though dense or sparse non-piecewise linear separable groupings of arbitrary shape. A dissimilarity function by using the cost of the optimum path from a datum to each entity on a graph was defined. It was also proven the usefulness of the dynamic algorithm to deal with elongated and non-piecewise linear separable clusters as well as sparse and dense ones. Experimental results showed that complicated collections of objects could be successfully separated if there is lack of prior knowledge about the given data set. It was also shown that according to the experimental results, the proposed algorithm is robust than the K-means algorithm in many complicated cases but the computation of the dynamic scheme is more expensive than that of the K-
means method.

Acknowledgement

The authors would like to thank the anonymous referees for their many helpful comments.

References


Figure 1.- (A) the given data set; (B), (C) the two natural groupings obtained by using the proposed dynamic algorithm of clustering.

Figure 2.- (A) shows the plot of the ordered dissimilarity measure at a stage 0, $d_0$, and (B) its first derivative, on the data given in Figure 1(A).

Figure 3.- (A) the data set; (B), (C), (D), (E) the four natural clusters obtained by using the proposed dynamic algorithm of clustering.

Figure 4.- (A), (C), and (E) show the plot of $d_j$ at the stages $j = 0, 1, 2$, respectively, on the data set in Figure 3(A); (B), (D), and (F) show the plot of the first derivative of $d_j$ at the same stages $j = 0, 1, 2$, respectively.

Figure 5.- (A) the given data set; (B), (C) the two clusters obtained by using the K-means algorithm.

Figure 6.- (A) the data set; (B), (C), (D), (E) the four clusters obtained by the K-means algorithm.

Figure 7.- Results of the basic agglomerative hierarchical clustering using $d_{\text{max}}$.

Figure 8.- Results of the basic agglomerative hierarchical clustering using $d_{\text{avg}}$.

Figure 9.- Results of the stepwise-optimal procedure.

Figure 10.- (A),(D) Results of the stepwise-optimal procedure; (B),(E) results of the basic agglomerative hierarchical clustering using $d_{\text{avg}}$; (C),(F) results of the basic agglomerative hierarchical clustering using $d_{\text{max}}$.

Figure 11.- The data set, in the top left, and the three meaningful groupings obtained by the proposed dynamic algorithm.

Figure 12.- Results of the K-means clustering method for the pattern set showed in Fig. 11.

Figure 13.- Results of the stepwise-optimal procedure.

Figure 14.- Results of the basic agglomerative hierarchical clustering using $d_{\text{max}}$.

Figure 15.- Results of the basic agglomerative hierarchical clustering using $d_{\text{avg}}$.  

18