V. Conclusions

The concept of variable string lengths has been utilized in GAs for developing a pattern classifier (called VGA-classifier) that can determine the required number of hyperplanes automatically in order to approximate the class boundaries of any data set in $\mathbb{R}^N$. New genetic operators are defined for handling variable length chromosomes. The fitness function takes care of minimization of the number of misclassified samples as well as the required number of hyperplanes.

Experimental results on different kinds of data (with nonlinear, overlapping class boundaries and dimensions ranging from two to nine) indicate that given a value of $H_{\text{opt}}$, the VGA-classifier is able not only to automatically evolve an appropriate value of $H$ for a given data set, but also to result in improved performance as compared to its fixed length version. The performance of the classifier is also found to be comparable to, sometimes better than, those of the k-NN rule, Bayes maximum likelihood classifier and MLP.

The VGA-classifier is found to determine automatically the architecture and the associated connection weights of MLP. The method guarantees that the architecture will involve at most two layers (excluding the input and output layers), with the neurons in the first and second hidden layers being responsible for hyperplane and region generation, and those in the output for providing a combination of regions for the classes. This investigation will augment the application domain of the VGA-classifier to those areas where MLP has widespread use.

Since the principle of VGA-classifier is used for developing NCA, it becomes mandatory to consider hard limiting neurons in the derived MLP. Although this makes the network rigid and susceptible to noise and corruption in the data, one may use NCA for providing a possible appropriate structure of conventional MLPs.

REFERENCES


Using Cluster Skeleton as Prototype for Data Labeling

Yuhui Yao, Lihui Chen, and Yan Qiu Chen

Abstract—A new approach, designed for clustering data whose underlying distribution shapes are arbitrary, is presented. This study is concerned with the use of the skeleton of a cluster as its prototype, which can represent the cluster more closely than that of using a single data point. The given data set is then partitioned into those skeleton-represented clusters without any prior knowledge nor assumptions of hidden structures. A novel function called cluster characteristic function (CCF) has been constructed and the associated theorems have been proposed and proved that the proper number of clusters can be determined with the approach.

Index Terms—cluster characteristic function, clustering, fuzzy c-means, skeleton clustering, unsupervised learning.

I. INTRODUCTION

Clustering refers to the process of grouping samples so that the samples are similar within each group and different between any two groups. It attempts to discover the inherent structure in a data set. It is unsupervised learning and no priori information on the data distribution can be assumed. In recent years, many schemes have been proposed for solving this intrinsically difficult problem. These methods can be sorted into four major categories:

1) prototype-based central point clustering;
2) learning-network-based central point clustering;
3) shell-clustering;
4) graph clustering.

Prototype-based central point clustering attempts to use the central point of a cluster as the prototype to represent the cluster and then labels all samples with those prototypes. The first published method is hard c-means proposed by Duda and Hart [1]. Later modifications and improvements include:

2) Fuzzy c-means, proposed by Bezdek [2]. Although fuzzy c-means can find a partition of data for a fixed number c of clusters, it needs a good cluster index [18], [19] to automatically determine the optimal number of clusters. It is computationally expensive since it needs to perform the clustering for a range of c values. Furthermore, reliable validity measures are quite difficult to attain, and the number of clusters that optimizes a particular validity may not always be “correct” [4].

3) Possibilistic c-means, a mode-seeking algorithm, proposed by Krishnapuram and Keller [3].

4) Statistical models such as probabilistic mixtures, proposed by Titterington [5].

5) Vector quantization approaches such as the generalized Lloyd algorithm [6].

6) Maxim Clustering Entropy Principle models such as least biased fuzzy clustering method, proposed by Gerardo Beni and Xiaomin Liu [7].

Learning-network-based central point clustering uses learning networks to achieve data labeling. During the process of training, the network updates all the vector weights and lets them to be
close to the inherent pattern central points. Each output node of the network associates a particular cluster central point. There are many approaches of this category reported in the literature such as: learning vector quantization \cite{8}, whose disadvantage is sequence dependent. In order to cluster data sequence-independently, some improvements have been proposed such as the fuzzy learning-vector-quantization family models called fuzzy Kohonen clustering network \cite{9}, fuzzy learning vector quantization \cite{10}, and fuzzy competitive/cooperative Kohonen model \cite{11}. Those techniques are the integration of a family of competitive learning algorithms with some fuzzy learning vector quantization algorithms.

Both prototype-based central point clustering and learning-network-based central point clustering can achieve good performance when they are used to cluster data whose subgroup distributions exhibit regular spherical shapes. Unfortunately, if we apply those methods to cluster data whose distributions exhibit arbitrary shapes such as circles, long lines or spirals, they will take inappropriate assumption of the distribution shape being spherical and produce improper clustering results. This is due to their intrinsic spherically clustering process.

Shell-clustering techniques are proposed to detect clusters with shapes of circles, ellipses, curves, and curved surfaces \cite{12}–\cite{14}. Those methods need to assume the shape of a cluster distribution before learning, and use the shape assumption to build the cost function that is applied for cluster detection. One of such methods is based on the Hough transform that is computationally expensive, and requires large storage \cite{15}, \cite{16}. Others introduce fuzzy c-means \cite{17}, \cite{13}, \cite{14} to build their cost functions. In strict unsupervised learning, we do not have any shape information before clustering. Therefore, shell-clustering is not an unsupervised learning process with the strict definition of unsupervised learning. Furthermore, if the distribution shapes of the clusters cannot be described by a mathematical function, shell-clustering technique will be unsuitable.

Several graph theoretic methods have been used for clustering. The commonly used methods are:

2) Single-link and complete-link hierarchical algorithms formulated and implemented using a threshold graph \cite{20}.
3) Forming clusters by breaking inconsistent arcs in minimum spanning tree of the proximity graph \cite{21} or graph constructed based on limited neighborhood sets \cite{22}.
4) Detecting clusters using directed trees \cite{23}.
5) Minimum cuts in an undirected adjacency graph \cite{24}.

The advantage of graph clustering is that they can cluster data with any irregular underlying shapes. However, those methods may be noise sensitive or may consume a great amount of computations.

We propose in this correspondence a new technique for data clustering that uses cluster skeleton as prototype to overcome the disadvantage of the existing methods. In a multidimensional feature space, each cluster corresponds to an underlying class region. Sampled data are distributed among those underlying regions. In order to represent the hidden structure of a cluster more closely, a skeleton of it is used. A class region, its reference points, and its cluster skeleton are shown in Fig. 1 as an example.

The new clustering technique uses the skeleton of a cluster to represent its underlying structure and partitions data into those skeleton-represented clusters. We name it skeleton clustering (SC). A cluster skeleton is a data set that consists of some reference points of a class. Each reference point represents a local data distribution of the cluster. Therefore the cluster skeleton can provide more information about the underlying structure of the data than that of using a single point. SC is an unsupervised learning process, and the partition of data points is accomplished without any priori knowledge nor assumptions of the cluster such as the number of clusters, and the underlying shape of each cluster.

The SC procedure consists of two phases. In the first phase, we employ fuzzy c-means to generate reference points. Each reference point corresponds to the center of a local data distribution. After generating the reference points, SC uses a process of skeleton generation to find out the number of clusters, form the skeleton of each cluster and assign data to those cluster skeletons. This generation of cluster skeleton is guided by a novel cluster characteristic function (CCF).

The rest of this correspondence is organized as follows. In Section II, we explain in detail about data clustering, the structure of SC and the novel CCF. In Section III, we study the performance of SC with three experiment sets. All clustering results have demonstrated that SC can perform well in finding the number of clusters, detecting the underlying shape of each cluster and labeling the data quickly and correctly. We have also compared the clustering results of SC with those of hard c-means, which is a typical point-prototype-based clustering technique, and with those of two popular graph clustering methods. Finally, conclusions are presented in Section IV.

II. SKELETON CLUSTERING

A. Clustering

Given \( N \) unlabeled samples in a feature space of dimension \( d \). Let \( X = \{ x_1, x_2, \ldots, x_N \} \subset \mathbb{R}^d \) denote the unlabeled data set. The \( i \)th sample has vector \( x_i \), as its numerical representation, and \( x_{ij} \) is the \( j \)th characteristic (or feature) associated with sample \( i \). In data clustering,
we use a label set $L = \{l_1, l_2, \cdots, l_c\}$ to label the set $X$. Each element of $L$ corresponds to a label, and $c$ is the number of labels. Clustering can be considered as a course of using a labeling function for $c$-labeling data set $X$, it can be written as

$$L^X = \Theta(X, L)$$

(1)

where $\Theta$ is a labeling function, and $L^X$ is the $c$-labels of $X$ by $\Theta$ with $L$. With a $\Theta$, $L^X$ are sets of $(N \times c)$ values which can be conveniently arranged as a $(N \times c)$ matrix

$$U = \{u_{ij}\}, \quad i = 1, \cdots, N; \quad j = 1, \cdots, c$$

(2)

where $u_{ij}$ is the label value of sample $x_i$ labeled with $l_j$. There are three categories of $c$-labeling matrices: possibilistic $U^{pc}$, constrained fuzzy or probabilistic $U^{fc}$, and crisp (hard) $U^{hc}$. They are defined as follows:

$$U^{pc} = \left\{ U \in \mathbb{R}^{N \times c} | 0 \leq u_{ij} \leq 1 \quad \forall i, j \right\}$$

(3a)

$$U^{fc} = \left\{ U \in U^{pc} \left| \sum_{j=1}^{c} u_{ij} = 1 \right. \forall i \right\}$$

(3b)

$$U^{hc} = \left\{ U \in U^{fc} | u_{ij} \in \{0, 1\} \quad \forall i, j \right\}$$

(3c)

If the labeling matrix $U$ is crisp or fuzzy, $u_{ij}$ is considered as the membership of data $x_i$ with $l_j$. If $U$ is probabilistic, $u_{ij}$ is usually the (posterior) probability $p(l_j|x_i)$ that, given $x_i$, it came from $l_j$. And if $U$ is possibilistic, $u_{ij}$ is taken as the possibility that $x_i$ belongs to $l_j$.

B. Skeleton Clustering Procedure

Since using the skeleton of a cluster can closely represent the cluster distribution with arbitrary shape, SC applies its skeleton as the cluster label. A cluster skeleton is a data set that consists of the reference points of the cluster. Each reference point represents a part of the cluster distribution. The process of SC is: Given an unlabeled data set $X$ that has $c$ intrinsic clusters, after detecting the number $c$ and the skeleton of each cluster, we label all elements of $X$ with those $c$ cluster skeletons. That is Given $X = \{x_i\}, x_i \in \mathbb{R}^d, i = 1, \cdots, N$

$$\Psi = \Gamma^{SC}(X)$$

$$\Psi = \{\psi_j\}, \quad j = 1, \cdots, c$$

(4a)

$$U^{sc} = \left\{ U \in U^{hc} | U \in \Theta^{SC}(X, \Psi) \right\}$$

$$U^{sc} = \{ u_{ij} \}, \quad i = 1, \cdots, N; \quad j = 1, \cdots, c$$

(4b)

where $\Gamma^{SC}$ is a function to obtain cluster skeletons, $\psi_j$ is the number of underlying clusters, $\Psi$ is the set of cluster skeletons, $\psi_j$ is the skeleton of the $j$th cluster, $\Theta^{SC}$ is a crisply labeling function, and $u_{ij}$ is considered as the membership of data $x_i$ with cluster $j$.

1) The structure of skeleton clustering: The procedure of SC has two stages as depicted in Fig. 2. In the first stage, it generates the multiple reference points of each cluster, and lets each reference point correspond to the center of a local data distribution area. This approach applies fuzzy c-means to build up the multiple reference points of the set $X$. After generating the reference points, SC performs the hidden structure extraction of the set $X$ by a cluster skeleton generating and data labeling process. It generates cluster skeletons and crisply labels the data with those skeleton-represented clusters. The generation of cluster skeletons is guided by a novel CCF that will be discussed in Section II-B3. The outputs of SC are the labels of data.

2) Reference points generation: The generation of reference points is achieved through fuzzy c-means. The learning process is to minimize the fuzzy c-means objective function

$$Q(V, R, X) = \frac{1}{k} \sum_{j=1}^{k} \sum_{i=1}^{N} v_{ij}^m \|x_i - r_j\|^2$$

(5)

where $R = \{r_1, \cdots, r_k\}$ are reference points; $V$ is a fuzzy $k$-partition matrix of data set $X = \{x_i\}$. The parameter $m \in [1, \infty)$ represents the amount of increased sharing of points among all clusters. $m = 1$ corresponds to the crisp case and $m \rightarrow \infty$ corresponds to the maximally fuzzy case. It is normally chosen $m = 2$ which is known to give good results for a physical interpretation of fuzzy c-means [25], [26]. The function $\| \cdot \|_{1/2}$ is a distance measure. The aim of this learning process is to let the $k$ reference points disperse to $k$ local distribution areas, and each reference point represents the data points within its distribution area. We select $k$ to be the integer closest to $\sqrt{N}$, which is $k = \text{round}(\sqrt{N})$, to balance the numbers of the representatives and of the points each representative is accountable to, where $N$ is the number of the unlabeled data points. The fuzzy c-means algorithm can be highlighted as follows. Unlabeled data are cycled in the network, during the $t$th cycle the vectors in the output layer are updated as

$$r_{j,t} = \frac{\sum_{i=1}^{N} (v_{ij})^m x_i}{\sum_{i=1}^{N} (v_{ij})^m}$$

for $1 \leq j \leq k$ (6)

$$v_{ij,t} = \frac{1}{\sum_{k=1}^{k} \left( \|x_i - r_{j,t}\|^2 \right)^{(1/(m-1))}} \quad \text{for } 1 \leq j \leq k$$

(7)

The iteration is terminated when

$$E_t = \sum_{j=1}^{k} \left\| r_j - r_{j,t-1} \right\|^2 \leq \varepsilon.$$  (8)

3) Cluster characteristic function: The measure of the relationship between any two pattern vectors $(v^i, v^j)$ is very important in a clustering process. In our method we use

$$P(v^i, v^j) = e^{-\alpha \|v^i - v^j\|^2}.$$  (9)

The coefficient $\alpha$ reflects the Mean-Square-Distance. It determines the speed of the decay of the relationship value

$$\alpha = \frac{1}{\sum_{j=1}^{k} \left\| x_j - \bar{x} \right\|^2}.$$  (10)

where $x_i$ is $i$th point of the unlabeled data set $X$ and $\bar{x} = (1/N) \sum_{i=1}^{N} x_i$ is the mean of $X$.

After applying fuzzy c-means for the selection of the reference points, the $k$ reference points indicate $k$ possible clusters being allocated. To decide whether some of those reference points can be grouped and hence the number of clusters is determined, a similarity-threshold is used to guide the combination. If the similarity of two reference points is larger than the similarity-threshold, those two reference points will be combined into the same cluster. In this study, we propose the use of a voting degree as a measure to reflect the similarity between two reference points. The use of such collective information helps to improve the robustness of the clustering. The novel similarity measure can be intuited as a voting process depicted
in Fig. 3. In this case it is assumed that there are five reference points (A,B,C,D,E). For a given sample $x_i$, if the relationship measurements $P(x_i, A)$, $P(x_i, B)$, $P(x_i, C)$ are greater than permission-threshold $\xi$, and $P(x_i, D)$, $P(x_i, E)$ are less than permission-threshold $\xi$. Then the sample $x_i$ has the right to vote for reference points A, B, and C belonging to the same family and the similarity degrees $s_{A\xi}, s_{B\xi}, s_{C\xi}$ are all increased by one. But the sample doesn’t have any right to vote for D and E. If more and more samples vote for A, B, C, it will be more certain that those three reference points belong to the same cluster. For $\forall r_i \in R$, since the reference point that has the largest relationship measurement with $r_i$ should have higher probability than the other reference points to be of the same class, we choose the permission-threshold $\xi$ as

$$\xi = \frac{1}{k} \sum_{j=1}^{k} \max(P(r_i, r_j)) \quad j = 1, \cdots, k \text{ and } j \neq i \quad (11)$$

where $P$ is the relationship measurement as (9). If sample $x_i$ has more than one reference points whose relationships with it are all larger than the permission-threshold $\xi$, then $x_i$ has the right to ascertain each similarity degree of any two reference points of them. Then the similarity between any two reference point $r_p$ and $r_q$ can be formulated as

$$s_{pq} = s_{qp} = \sum_{j=1}^{N} \left( \text{sgn} \left( P(x_i, r_p) - \xi \right) \times \text{sgn} \left( P(x_i, r_q) - \xi \right) \right)$$

$$p, q = 1, \cdots, k, \quad p \neq q \quad (12)$$

where $P$ is the relationship measurement, $\xi$ is the permission-threshold, $k$ is the number of reference points, $N$ is the number of data points, and $\text{sgn}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$.

With one cycle of the data voting, the similarities of any two reference points have been evaluated. Corresponding to the above criteria, if the similarity of two reference points is larger than the similarity-threshold $\xi$, the two reference points will be combined into the same cluster. All reference points that belong to the same cluster form the skeleton of this cluster. Thus the selection of a similarity-threshold $\xi$ to guide the combination plays a crucial role in forming cluster skeletons. Unsuitable similarity thresholds may generate incorrect number of clusters. In order to select an appropriate similarity-threshold and hence find out the proper number of clusters, we propose a novel function (CCF) to describe the relation between the number of clusters and the choice of similarity-threshold.

In order to construct the CCF for a given data set, the following terminologies and theorems are proposed. Let $G = (R, S)$ be an undirected graph with vertex set $R = \{r_1, \cdots, r_k\}$ and arc set $S = \{s_{pq} | p, q = 1, \cdots, k\}$, where the vertices denote the reference points and the value of arc between the vertex $r_p$ and $r_q$ represents the similarity $s_{pq}$. Associated with every arc $s_{pq}$, another parameter $d_{pq}$ is used to describe the affinity between $r_p$ and $r_q$, which is defined as follows. Let $\zeta$ be all the paths from source vertex $r_p$ to $r_q$ and $\hat{S}$ be the set of all the arc values in the path $\hat{P}$, where $\hat{P}$ is one path of $\zeta$. Then

$$a_{pq} = \max_{P \subseteq \zeta} \left( \min_{S \subseteq \hat{S}} \left( \sum_{i=1}^{k} s_{ij} \right) \right) \quad (13)$$

For a similarity-threshold $\delta$, we build a matrix $Z(\delta) = \{z_{pq}\}$ named cluster characteristic matrix that satisfies

$$z_{pq} = \begin{cases} \text{sgn}(a_{pq} - \delta), \quad p \neq q \\ 1, \quad p = q \end{cases} \quad (14)$$

Then CCF can be formulated as

$$n_{cluster} = \text{rank}(Z(\delta)) \quad (15)$$

Lemma 1: Given a similarity-threshold $\delta$, if the similarity of two reference points is larger than $\delta$, we combine them into the same cluster. Then for all vertices within the same cluster, their line vectors in the cluster characteristic matrix have the same value.

Proof: For $\forall r_p, r_q \in R$, if $p = q$, $z_{pp} = 1$. If $p \neq q$, and it exists a path $\hat{P}$ in graph $G$ from vertex $r_p$ to $r_q$, so the minimal arc value in $\hat{P}$ is larger than $\delta$. According to (13) and (14), all vertices along $\hat{P}$ can be grouped to the same cluster, then $z_{pp} = z_{pq} = 1$. If there is no path $\hat{P}$ in graph $G$ from vertex $r_p$ to $r_q$, so the minimal similarity value in $\hat{P}$ is larger than $\delta$, $z_{pp} = z_{pq} = 0$. Hence for all vertices that can be grouped within the same cluster, the values of the associated elements in the cluster characteristic matrix are totally 1, otherwise the elements are 0. Then for all vertices within the same cluster, their line vectors in cluster characteristic matrix are same. It is named cluster characteristic vector.

Lemma 2: For any two vertices $r_p, r_q$, if they belong to different clusters, then

$$\sum_{i=1}^{k} z_{pi}z_{qi} = 0$$

where

$$Z_{zp} = [z_{p1}, z_{p2}, \cdots, z_{pk}], \quad Z_{rq} = [z_{q1}, z_{q2}, \cdots, z_{qk}], \quad \forall z \in \{0, 1\}.$$

Proof: By contraposition. If $\Sigma_{i=1}^{k} z_{pi}z_{qi} \neq 0$, since $\forall z \in \{0, 1\}$, then $\Sigma_{i=1}^{k} z_{pi}z_{qi} > 0$. Therefore, $\exists z$ that satisfies $z_{pi} = 1$ and $z_{qi} = 1$. Therefore, vertex $r_p, r_q$ can be grouped to be the same cluster by vertex $r_z$. It conflicts with that $r_p, r_q$ belong to different clusters. Therefore, $\Sigma_{i=1}^{k} z_{pi}z_{qi} = 0$.

Theorem 1: Given a similarity-threshold $\delta$, if the similarity of two reference points is larger than $\delta$, we combine them into the same cluster. Then the number of clusters generated by $\delta$ is the rank of the cluster characteristic matrix. That is $n_{cluster} = \text{rank}(Z(\delta))$.

Proof: With Lemmas 1 and 2, all vertices within the same cluster have the same cluster characteristic vectors in the matrix. For any two vertices $r_p, r_q$, if they belong to different cluster, $\Sigma_{i=1}^{k} z_{pi}z_{qi} = 0$. Therefore, cluster characteristic vectors are linear independent. Therefore it is $n_{cluster} = \text{rank}(Z(\delta))$.

Theorem 2: Assume that the similarities of any two points within the same cluster and between two different clusters are identical respectively. If $\min_{z \in S} (\hat{S}_i) > \max_{z \in S} (\hat{S}_i)$, and

$$\left\{ \begin{array}{l} \min_{z \in S} (\hat{S}_i) > \max_{z \in S} (\hat{S}_i) \\ \hat{S}_i \subseteq S_{in} \end{array} \right\} \quad (16)$$

where $S_{in}$ denotes the similarity set within each class of the data set (Internal), and $S_{ex}$ denotes the similarity set between different classes (External). Then excluding the first stay $s_{t0} = [0, \min_{z \in S} (\hat{S}_i)]$, the longest stay in CCF can be used to determine the proper number of clusters of the data, where the stay denotes the range of similarity thresholds corresponding to the same value of a cluster number.

Proof: It assumes that there are $c$ classes in the data set. With the definition of CCF, it has the following.
The similarity between each two of ten prototypes of set 1

<table>
<thead>
<tr>
<th>Similarity</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(-0.2865, 0.7846)</td>
<td>(-0.3015, 0.2011)</td>
<td>(-0.3049, 0.0197)</td>
<td>(-0.4807, 0.2979)</td>
<td>(-0.0407, 0.0195)</td>
<td>(-0.5660, 0.0037)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10 0 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0 10 0 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0 0 10 0 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0 0 0 10 0 0 0 0 0 0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where $\beta$ is the level of selected threshold. Hence

$$\lambda(CCF) = O(N^2).$$

For example, if $N = 100$, $d = 2$, $\beta = 10$, $\lambda(CCF) = 11552$ (FPO) the computation will take less than 1 second on Pentium PC.

4) Data labeling: The data labeling of SC follows such criteria. For each data $x_i \in X$, we label it with the class of reference point $r_j$, which has the largest relationship with $x_i$

$$P(x_i, r_j) = \max(P(x_i, r_g)), \quad g = 1, \ldots, k.$$  

For example, if $r_j$ belongs to cluster $h(1 \leq h \leq c)$, we label data $x_i$ with cluster $h$.

Pseudocode: The SC procedure can be summarized by the following pseudocode.

1) Given an unlabeled data set $X = \{x_1, \ldots, x_N\}$, fix: $\|x\|_{\alpha}$; $m = 2$; and $\varepsilon > 0$, a small positive constant.

2) Apply fuzzy c-means learning to generate $k$ reference points.

3) Calculate permission threshold $\xi$ and Mean-Square-Distance $\alpha$ as described in (10), (11). For each data $x_i \in X$, calculate its relationships with $k$ reference points. If there are more than one relationships that are bigger than $\xi, x_i$ votes the similarity between any two of those reference points.

4) Build the CCF of reference points, determine the number of clusters and generate the cluster skeletons.

5) Label the data with the cluster skeletons.

III. EXPERIMENTAL RESULTS

We have studied the clustering performance of SC on three artificial sets. Two sets are two-dimensional (2-D) and one is in three-dimensional (3-D) feature space. All three data sets represent different cluster structures, which are used to demonstrate that SC can cluster data of any dimensions and with any arbitrary inherent distribution shapes by using CCF in an unsupervised manner. This study also demonstrates that even in the presence of some considerable noise, SC is robust and able to correctly determine the number of clusters and then form the skeleton for each cluster.

A. Two Semi-Circle Shape Clusters

Set 1 consists of 100 data points uniformly distributed along the two semi-circles, and each class has 50 samples, as depicted in Fig. 4(a). In order to study the clustering performance of SC by using CCF, we fix the values of parameters as discussed in Section II. That is $\|x\|_{\alpha}$ is
the Euclidean distance, and $m = 2$. The ten reference points generated through fuzzy c-means learning are also shown in Fig. 4(a) with "star *." After one cycle of set $X$ voting, the similarities between each two of the ten reference points are calculated and shown in Table I, and the positions of the ten reference points are shown in Table I as well. Corresponding to the similarities of those ten reference points of Set 1, we build its CCF as shown in Fig. 4(c). The CCF indicates that there exists the longest stay of the similarity threshold with the number of clusters remaining 2. Hence, a similarity-threshold can be selected within the longest stay. When $\delta = 6$ is chosen, we can determine that reference points 1, 3, 4, 6, and 7 form one cluster skeleton and reference points 2, 5, 8, 9, and 10 form the other. All of them are displayed in Fig. 4(b). Fig. 4(d) and (f) show the clustering performance of SC applied to Set 1 added with 40 randomly generated noise points. The above experimental results demonstrate that SC can still determine the number of clusters, form the cluster skeletons convincingly and achieve robust clustering property even when the samples are contaminated with over 28% of noise. We have also compared the performance of SC with
two graph clustering methods, Threshold Graph Clustering [20] and Minimum Cuts Graph Clustering [24]. The comparison results are performed on a PC Pentium-300 with Matlab 5.0, which are illustrated in Table II and Fig. 4(e). They show that Threshold Graph Clustering can achieve a higher speed in performance but it has generated improper results in the presence of noise. The results also show that Minimum Cuts Graph Clustering is able to correctly cluster the data sets but it is computationally expensive.

B. Two 3-D Clusters

Set 2 is used to further confirm that SC can be used to cluster data of any dimensions. There are two classes in Set 2. One class has 50 samples distributed along a 3-D line, and the other has 150 samples distributed along a spiral curve of 3-D, as shown in Fig. 5(a). The 14 reference points generated by fuzzy c-means learning are also shown in Fig. 5(a) with “star *.” By using the similarities of those 14 refer-

### TABLE III

<table>
<thead>
<tr>
<th></th>
<th>Set 1</th>
<th>Set 2</th>
<th>Set 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class 1</td>
<td>Class 2</td>
<td>Class 1</td>
</tr>
<tr>
<td>SC</td>
<td>Correct percent</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Hard c-means</td>
<td>Correct percent</td>
<td>74%</td>
<td>74%</td>
</tr>
</tbody>
</table>

Fig. 5. Clustering results of the Set 2 data using SC. (a) Set 2 data and its 14 reference points. (b). CCF of Set 2. (c). Cluster skeletons of Set 2 with similarity threshold $\delta = 2$. 

**TABLE III**

**THE PERFORMANCE COMPARISON OF SC WITH HARD c-MEANS IN CLUSTERING SET 1, 2 AND 3**
ence points we build the CCF of them as shown in Fig. 5(b). It has the longest stay at cluster number being 2. Therefore, according to the criteria in Section II, we determine that the number of clusters of Set 2 is 2. In addition, we select similarity-threshold $\delta = 2$ to do the combination of reference points. The cluster skeletons formed with the above similarity-threshold are indicated in Fig. 5(c). The experiment results have demonstrated, with the guide of the CCF, SC can be applied for cluster data in high dimensions.

C. Three Clusters with Arbitrary Shapes

Set 3 contains three clusters in 2-D feature space. Two clusters take sinusoidal shapes, and the third is of irregular shape. Each of the three clusters has 50 samples. Firstly through the learning of fuzzy c-means, we obtain 12 reference points shown in Fig. 6(a) accompanied with the data of Set 3. One cycle of the data voting acquires the similarities of each two of those 12 reference points, and hence builds its CCF in Fig. 6(b). With the "help" of the longest stay in the CCF, the number of clusters is concluded to be 3. Fig. 6(c) has illustrated the three cluster skeletons of Set 3 formed with the similarity-threshold $\delta = 3$.

The purpose of the experiments discussed above is to demonstrate the feasibility of SC for clustering data with arbitrary shapes. Due to the scope limitation of study, no strict comparisons were made to compare the performance of SC with the existing techniques when the underlying distributions of data are known beforehand. However a typical point-prototype-based clustering method hard c-means has been applied to highlight the limitations of the existing approaches for data clustering with unknown distributions. These can been seen in Figs. 7(a)–(c) and Table III. It is noted from them that using hard c-means cannot do proper clustering for all the three sets.

IV. CONCLUSIONS

We have presented and discussed a new SC method that is capable of detecting hidden structures in data of arbitrary distribution shapes. SC applies cluster skeletons as the prototype of clusters to overcome the inadequacy of using a point to represent a cluster with irregular shape. The whole learning process is performed without knowing in advance the number of clusters and without making any assumptions on the data.

The SC technique consists of two phases. As for reference-point generation, it applies fuzzy c-means to obtain the central points of local distributions of data. The permission threshold is therefore found as its output. The similarity of two reference points is measured by only one cycle of sample voting. Corresponding to the entire similarities of
the reference points we have devised a novel function CCF to describe the relation between the number of clusters and the selection of similarity-threshold. We have also proved that CCF can be used to detect the inherent information about the number of clusters under certain conditions. After appropriately choosing the similarity-threshold within the longest stay in the CCF, we can form the cluster skeletons and label the data accordingly.

The performance of SC has been studied with three simulations. All experiment results have shown that SC can robustly and quickly detect the hidden structures of the data set such as the number of clusters, the underlying shapes of clusters and the labels of data. This new approach can be feasibly used for data clustering when the underlying distributions of data are unknown. However, based on the comparison results with other clustering methods, there seems to be room for further research to speed up the SC algorithm and make SC more applicable to larger database clustering. The CCF in this case is a statistical way to generate skeletons. In future work, the present approach could be enhanced by using a more sophisticated technique to analyze the relationships among the reference points.

REFERENCES

A GA-Based Method for Constructing Fuzzy Systems Directly from Numerical Data

Ching-Chang Wong and Chia-Chong Chen

Abstract—A method based on the concepts of genetic algorithm (GA) and recursive least-squares method is proposed to construct a fuzzy system directly from some gathered input-output data of the discussed problem. The proposed method can find an appropriate fuzzy system with a low number of rules to approach an identified system under the condition that the constructed fuzzy system must satisfy a predetermined acceptable performance. In this method, each individual in the population is constructed to determine the number of fuzzy rules and the premise part of the fuzzy system, and the recursive least-squares method is used to determine the consequent part of the constructed fuzzy system described by this individual. Finally, three identification problems of nonlinear systems are utilized to illustrate the effectiveness of the proposed method.

Index Terms—Fuzzy system, genetic algorithm, recursive least-squares method.

I. INTRODUCTION

Fuzzy modeling is a branch of system identification. It concerns the construction of the fuzzy inference system that can explain the behavior of an unknown system described by a set of sample data [1], [3], [5]–[18]. In general, the premise part of a fuzzy rule defines a local fuzzy region, while the consequent part describes the behavior within the region via various constituents. The consequent constituent can be a fuzzy set, a constant or a linear equation. That is, different consequent constituents result in different fuzzy inference systems, but their premise parts are always the same. In this paper, the Takagi-Sugeno fuzzy model [12] is considered. In general, the fuzzy system design is divided into two steps: the structure identification and the parameter identification. In the structure identification step, the input space can be partitioned to describe the inherent structure of an identified system so that the number of fuzzy rules and the shapes of the fuzzy sets in the premise part are determined. Subsequently, in the parameter identification step, a parameter estimation method can be applied to fine-tune the parameters of the obtained fuzzy system in the structure identification step. In this paper, each individual in the population is considered to partition the input space to determine a rough fuzzy system structure and its premise part, then the recursive least-squares method is applied to determine the consequent part of the constructed fuzzy system. Therefore, a fuzzy system is constructed by the hybrid GA and recursive least-squares method.

The disadvantage of conventional approaches for the fuzzy system design is that the number of fuzzy sets of each input variable must be defined in advance. Therefore, they may have redundant fuzzy sets such that a large number rules are generated for the conventional approaches. Furthermore, when we only have the input-output data of the considered system, it is difficult to extract appropriate fuzzy rules for a predefined system performance directly from numerical data. Consequently, in order to avoid these drawbacks, we propose a method based on the concepts of GA [2], [4], [8], [15]–[17] and recursive least-squares method.

Manuscript received February 17, 1999; revised September 6, 1999 and July 11, 2000. This work was supported in part by the National Science Council of the Republic of China under Contract NSC-89-2213-E-032-012. This paper was recommended by Associate Editor H. Teodorescu.

C. C. Wong is with the Department of Electrical Engineering, Tamkang University, Taiwan, R.O.C. (e-mail: wong@ee.tku.edu.tw).

C. C. Chen is with the Department of Electronic Engineering, Wufeng Institute of Technology, Taiwan, R.O.C.

Publisher Item Identifier S 1083-4419(00)08806-3.