PERGAMON

Pattern Recognition 34 (2001) 415–424

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

A genetic approach to the automatic clustering problem☆

Lin Yu Tseng*, Shiueng Bien Yang

Department of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan 402, Republic of China

Received 13 October 1999; received in revised form 7 December 1999; accepted 7 December 1999

Abstract

In solving the clustering problem, traditional methods, for example, the *K*-means algorithm and its variants, usually ask the user to provide the number of clusters. Unfortunately, the number of clusters in general is unknown to the user. Therefore, clustering becomes a tedious trial-and-error work and the clustering result is often not very promising especially when the number of clusters is large and not easy to guess. In this paper, we propose a genetic algorithm for the clustering problem. This algorithm is suitable for clustering the data with compact spherical clusters. It can be used in two ways. One is the user-controlled clustering, where the user may control the result of clustering by varying the values of the parameter, *w*. A small value of *w* results in a larger number of compact clusters, while a large value of *w* results in a smaller number of looser clusters. The other is an automatic clustering, where a heuristic strategy is applied to find a good clustering. Experimental results are given to illustrate the effectiveness of this genetic clustering algorithm. © 2000 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Clustering; Single-linkage algorithm; Genetic clustering algorithm

1. Introduction

The clustering problem is defined as the problem of classifying a collection of objects into a set of natural clusters without any a priori knowledge. For years, many clustering methods were proposed [1–6]. These methods can be basically classified into two categories: hierarchical and non-hierarchical. The hierarchical methods can be further divided into the agglomerative methods and the divisive methods. The agglomerative methods merge together the most similar clusters at each level and the merged clusters will remain in the same cluster at all higher levels. In the divisive methods, initially, the set of all objects is viewed as a cluster and at each level, some clusters are binary divided into smaller clusters. There are also many non-hierarchical methods. Among them, the *K*-means algorithm is an important one. It is an iterative hill-climbing algorithm and the solution obtained depends on the initial clustering. Although the *K*-means algorithm had been applied to many practical clustering problems successfully, it is shown in Ref. [7] that the algorithm may fail to converge to a local minimum under certain conditions. In Ref. [8], a branch-and-bound algorithm was proposed to find the globally optimum clustering. However, it might take much computation time. In Refs. [9,10], simulated annealing algorithms for the clustering problem were proposed. These algorithms may find a globally optimum solution under some conditions. In Ref. [11], the evolution strategies were explored for solving the clustering problem. The clustering was viewed as an optimization problem that tried to optimize the clustering objective function. Most of these clustering algorithms require the user to provide the number of clusters as input. But the user in general has no idea about the number of clusters. Hence, the users are forced to try different numbers of clusters when using these clustering algorithms. This is tedious and the clustering result may be no good especially when the number of clusters is large and not easy to guess. In Ref. [12], the definition of *K*-cluster was proposed. In

☆This research work was partially supported by National Science Council of Republic of China under contract NSC87-2213-E-005-002.

* Corresponding author. Tel.: + 886-4-2874020; fax: + 886-4-2873028.

E-mail address: lytseng@amath.nchu.edu.tw (L.Y. Tseng).

PII: S0031-3203(00)00005-4

particular, the class of 1-clusters is just that obtained by the single-linkage algorithm. Based on a probability model, Ling tried to find a good clustering by using the single-linkage algorithm and two indices for measuring compactness and relative isolation. In Ref. [13], the nearest-neighbor algorithm based on the mean distance from an object to its nearest neighbor was proposed. Just like other neighborhood clustering methods, the threshold of distance for grouping objects together is difficult to decide. Some papers, for example, Refs. [14,15], had been devoted to the problem of determining the threshold.

Since the genetic algorithm is good at searching [16], a genetic algorithm was proposed to search the optimal clusters in Ref. [17]. But this clustering algorithm also requires the user to provide the number of clusters before clustering. In this paper, we propose a genetic clustering algorithm. The clustering algorithm will automatically search for a proper number as the number of clusters and classify the objects into these clusters at the same time. In searching a good clustering, the value of a parameter varies within a given range and several possible clusterings are obtained. Then, a heuristic strategy is applied to choose a good clustering. Before using the genetic clustering algorithm, we also apply the single-linkage algorithm to reduce the size of the data set if the size is large.

The remaining part of the paper is organized as follows. In Section 2, the basic concept of the genetic approach is introduced. In Section 3, the clustering algorithm is described. In Section 4, the heuristic strategy to choose a good clustering is given. Experimental results are described in Section 5 and the paper is concluded in Section 6.

2. The basic concept of the genetic strategy

The genetic strategy consists of an initialization step and the iterative generations. In each generation, there are three phases, namely, the reproduction phase, the crossover phase and the mutation phase.

In the initialization step, a set of strings will be randomly generated. This set of strings is called the population. Each string consists of 0's and 1's. After the initialization step, there is an iteration of generations. The user may specify the number of generations that he or she wants the genetic algorithm to run. The genetic algorithm will run this number of generations and retain the string with the best fitness. This string represents the solution obtained by the genetic algorithm. The three phases in each generation will be introduced in the following.

In the reproduction phase, the fitness of each string is calculated. The calculation of the fitness is the most important part in our algorithm. After the calculation of the fitness for each string in the population, the reproduction operator is implemented by using a roulette wheel with slots sized according to fitness. In the crossover phase, pairs of strings are chosen. For each chosen pair, two random numbers are generated to decide which pieces of the strings are to be interchanged. Suppose the length of the string is n, then each random number is an integer in $[1, n]$. For example, if two random numbers are 2 and 5, position 2 to position 5 of this pair of strings are interchanged. For each chosen pair of strings, the crossover operator is applied by probability p_c. In the mutation phase, bits of the strings in the population will be chosen with probability p_m. Each chosen bit will be changed from 0 to 1 or from 1 to 0.

3. The genetic clustering algorithm

In this section, the clustering algorithm CLUSTERING is described. Let there be n objects, $O_1, O_2, \ldots, O_n$. Suppose each object is characterized by p feature values. Hence, we have $O_i = (o_{i1}, o_{i2}, \ldots, o_{ip}) \in R^p$. The algorithm CLUSTERING consists of two stages. The first stage is the nearest-neighbor algorithm, which consists of the following steps. The distance used in the nearest-neighbor algorithm is based on the average of the nearest-neighbor distances.

Step 1: For each object O_i, find the distance between O_i and its nearest neighbor. That is,

$$d_{\mathrm{NN}}(O_i) = \min_{j \neq i} \|O_j - O_i\|, \tag{1}$$

where $\|O_j - O_i\| = \left(\sum_{q=1}^{p} (O_{jq} - O_{iq})^2\right)^{1/2}$.

Step 2: Compute d_{av}, the average of the nearest-neighbor distances by using Eq. (1) as follows:

$$d_{\mathrm{av}} = \frac{1}{n} \sum_{i=1}^{n} d_{\mathrm{NN}}(O_i). \tag{2}$$

Let $d = u^* d_{\mathrm{av}}$. (d is decided by the parameter u).

Step 3: View the n objects as nodes of a graph. Compute the adjacency matrix $A_{n \times n}$ as follows:

$$A(i,j) = \begin{cases} 1 & \text{if } \|O_i - O_j\| \leqslant d, \\ 0 & \text{otherwise,} \end{cases}$$

where $1 \leqslant j \leqslant i \leqslant n$.

Step 4: Find the connected components of this graph. Let the data sets represented by these connected components be denoted by $B_1, B_2, \ldots, B_m$ and the center of each set be denoted by V_i for $1 \leqslant i \leqslant m$.

Since several objects may be grouped in a set B_i, the number of sets m is less than the number of objects n in the original data set. The objective of using the nearest-neighbor algorithm in the first stage is to reduce the computation time in the second stage. Therefore, the

clustering algorithm can process the large data set efficiently.

In Step 2 of the above algorithm, the value slightly greater than one is chosen for the parameter u in order to make all the objects in the same set close enough to one another. The sets $B_1, B_2, \ldots, B_m$ obtained in the first stage will be taken as the initial clusters in the second stage and each set B_i $(1 \leqslant i \leqslant m)$ is taken as if it is an object and will not be divided in the second stage. Basically, the second stage is a genetic algorithm, which will merge some of these B_i's if they are close enough to one another. The genetic algorithm consists of an initialization step and the iterative generations with three phases in each generation. They are described in the following.

Initialization step: A population of N strings is randomly generated. The length of each string is m, which is the number of the sets obtained in the first stage. N strings are generated in such a way that the number of 1's in the strings uniformly distributes within $[1, m]$. Each string represents a subset of $\{B_1, B_2, \ldots, B_m\}$. If B_i is in this subset, the ith position of the string will be 1; otherwise, it will be 0. Each B_i in the subset is used as a seed to generate a cluster.

Before describing the three phases, let us first describe how to generate a set of clusters from the seeds. Let $T = \{T_1, T_2, \ldots, T_s\}$ be the subset corresponding to a string. The initial clusters C_i's are T_i's and initial centers S_i's of clusters are V_i's for $i = 1, 2, \ldots, s$. The size of cluster C_i is $|C_i| = |T_i|$ for $i = 1, 2, \ldots, s$, where $|T_i|$ denotes the number of objects belonging to T_i.

The generation of the clusters proceeds as follows. The B_i's in $\{B_1, B_2, \ldots, B_m\} - T$ are taken one by one and the distance between the center V_i of the taken B_i and the center S_j of each cluster C_j is calculated. Then we have

$$B_i \subset C_j \quad \text{if } \|V_i - S_j\| \leqslant \|V_i - S_k\| \quad \text{for}$$

$$1 \leqslant k \leqslant s \quad \text{and} \quad k \neq j.$$

If B_i is classified as in the cluster C_j, the center S_j and the size of the cluster C_j will be recomputed by Eqs. (3) and (4) as follows when B_i is included in C_j:

$$S'_j = \frac{S_j * |C_j| + V_i * |B_i|}{|C_j| + |B_i|}, \tag{3}$$

$$|C'_j| = |C_j| + |B_i|. \tag{4}$$

After B_i's in $\{B_1, B_2, \ldots, B_m\} - T$ all have been considered, we obtain the cluster C_j with center S_j generated by the seed T_j for $j = 1, 2, \ldots, s$. We define $\{C_1, C_2, \ldots, C_s\}$ as the set of clusters generated by this string.

Reproduction phase: Let C_i be one of the clusters generated by string R. We define D_{intra} to represent the intra-distance in the cluster C_i and D_{inter} to represent the inter-distance between this cluster C_i and the set of all other clusters in the following equations:

$$D_{\text{intra}}(C_i) = \sum_{B_k \subset C_i} \|V_k - S_i\| * |B_k|, \tag{5}$$

$$D_{\text{inter}}(C_i) = \sum_{B_k \subset C_i} \left(\min_{j \neq i} \|V_k - S_j\| \right) * |B_k|, \tag{6}$$

where the summation in Eq. (6) is over all B_k's that are in the cluster C_i. Then we can define the fitness function of a string R as follows in Eq. (7):

$$\text{Fitness}(R) = \sum D_{\text{inter}}(C_i) * w - D_{\text{intra}}(C_i), \tag{7}$$

where w is a weight. If the value of w is small, we emphasize the importance of $D_{\text{intra}}(C_i)$. This tends to produce more clusters and each cluster tends to be compact. If the value of w is chosen to be large, we emphasize the importance of $D_{\text{inter}}(C_i)$. This tends to produce fewer clusters and each cluster tends to be loose. If R contains only 0's, Fitness (R) is defined to be 0. If R contains only one 1, $D_{\text{inter}}(C_i)$ is defined to be 0. After the calculation of fitness for each string in the population, the reproduction operator is implemented by using a roulette wheel with slots sized according to fitness.

An example may be helpful in understanding the relation between the values of w's and the numbers of clusters generated. In Fig. 1, there are four sets B_1, B_2, B_3, B_4 as the input to the second stage. Assume that each B_i contains three objects. In this stage, some of them may be grouped together to form the final clustering. Let string R_1 represent the subset $\{B_1, B_3, B_4\}$ and string R_2 represent the subset $\{B_2, B_4\}$. Each V_i denotes the center of B_i for $1 \leqq i \leqq 4$. Fig. 1(a) shows the three clusters, C_1, C_2 and C_3, generated from bit string R_1 and Fig. 1(b) shows the two clusters, C_1 and C_2, generated from bit string R_2. Each S_i denotes the center of C_i. Note that in Fig. 1(a), $S_2 = V_4$ and $S_3 = V_3$. Also, in Fig. 1(b), $S_2 = V_4$. In Fig. 1(a), let $\|V_1 - S_2\| = 11$, $\|V_2 - S_3\| = 8$, $\|V_3 - S_1\| = 9$, $\|V_4 - S_1\| = 11$, $\|V_1 - S_1\| = 2$, $\|V_2 - S_1\| = 2$. In Fig. 1(b), let $\|V_1 - S_1\| = 7$, $\|V_2 - S_1\| = 5$, $\|V_3 - S_1\| = 6$, $\|V_1 - S_2\| = 11$, $\|V_2 - S_2\| = 12$, $\|V_3 - S_2\| = 12$, $\|V_4 - S_1\| = 10$. When the value of w is 1.5, the fitness of

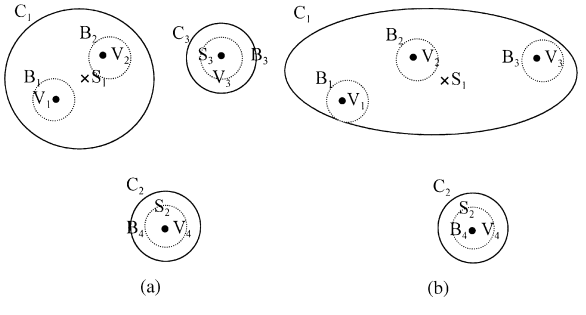

Fig. 1. An example illustrating the relation between the value of w and the number of clusters generated. (a) Three clusters. (b) Two clusters.

two bit strings R_1 and R_2 can be computed by Eqs. (5)–(7) as follows:

$$\text{Fitness}(R_1) = (11*3 + 11*3 + 8*3 + 9*3)*1.5 - (2*3 + 2*3) = 163.5,$$

$$\text{Fitness}(R_2) = (10*3 + 11*3 + 12*3 + 12*3)*1.5 - (7*3 + 5*3 + 6*3) = 148.5.$$

Since Fitness (R_1) is larger than Fitness (R_2), three clusters (Fig. 1(a)) are obtained by the genetic clustering algorithm. However, when the value of w is 2.5, the fitness of two-bit strings R_1 and R_2 can be computed as follows:

$$\text{Fitness}(R_1) = (11*3 + 11*3 + 8*3 + 9*3)*2.5 - (2*3 + 2*3) = 280.5,$$

$$\text{Fitness}(R_2) = (10*3 + 11*3 + 12*3 + 12*3)*2.5 - (7*3 + 5*3 + 6*3) = 283.5.$$

Since Fitness (R_2) is larger than Fitness (R_1), two clusters (Fig. 1(b)) are obtained by the genetic clustering algorithm. As indicated by the above example, if the value of w is large, a small number of loose clusters are produced. If the value of w is small, a large number of compact clusters are produced.

Crossover phase: If a pair of strings R and Q are chosen for applying the crossover operator, two random numbers p and q in $[1, m]$ are generated to decide which pieces of the strings are to be interchanged. Suppose $p < q$, the bits from position p to position q of string R will be interchanged with those bits that are in the same position of string Q. For each chosen pair of strings, the crossover operator is done with probability p_c.

Mutation phase: In the mutation phase, bits of the strings in the population will be chosen with probability p_m. Each chosen bit will be changed from 0 to 1 or from 1 to 0.

The user may specify the number of generations that he or she wants the genetic algorithm to run. The genetic algorithm will run this number of generations and retain the string with the best fitness. The user may specify the value of w used in calculating the fitness function in order to emphasize on either the compactness of clusters or the enlargement of the distances among clusters.

The time complexity of CLUSTERING is analyzed as follows. Let the size of data set be n. In the first stage, Step 1 takes $O(n^2)$ time to calculate the distances between pairs of objects and takes $O(n)$ time to find the minimum. Step 2 takes $O(n)$ time to calculate the average of the nearest-neighbor distances. Step 3 takes $O(n^2)$ time to derive the adjacency matrix and Step 4 also takes $O(n^2)$ time to find the connected components by scanning the adjacency matrix. Therefore, the first stage spends $O(n^2)$ time. In the second stage, let N denote the size of population and m denote the length of the string. It takes $O(m^2)$ time for each component to find the nearest cluster. The time complexity of the second stage is dominated by the calculation of the fitness function. It takes $O(Nm^2)$ time in the worst case. Suppose the genetic algorithm is asked to run G generations, the time complexity will be $O(GNm^2)$. Hence, the time complexity of the whole clustering algorithm is $O(n^2 + GNm^2)$.

4. The heuristic strategy to find a good clustering

In this section, the heuristic strategy to find a good clustering is described. Assume q clusters $\{C_1, C_2, \ldots, C_q\}$ are derived by using the algorithm CLUSTERING. We define D_1 and D_2 in Eqs. (8) and (9).

$$D_1(w) = \min_{1 \leqslant i < j \leqslant q} \|S_i - S_j\|, \tag{8}$$

$$D_2(w) = \max_{1 \leqslant i \leqslant q} \max_{B_k \subset C_1} \frac{\text{The mean radius of } C_i}{\text{The mean radius of } B_k} = \max_{1 \leqslant i \leqslant q} \max_{B_k \subset C_1} \frac{\sum_{O_j \in C_i} \|O_j - S_i\|/|C_i|}{\sum_{O_j \in B_k} \|O_j - V_k\|/|B_k|}, \tag{9}$$

where w is the value of the parameter w used in CLUSTERING. $D_1(w)$ represents the shortest distance among the centers of the clusters. Each cluster C_i may contain several B_k's. Let r_i represent the ratio of the mean radius of C_i to the smallest mean radius of these B_k's. Then $D_2(w)$ denotes the maximum of r_i's. $D_1(w)$ estimates the closeness of the clusters in the clustering. $D_2(w)$ estimates the compactness of the clusters in the clustering. In the heuristic strategy, a good clustering is decided by using CLUSTERING with the value of the parameter w varying within a range $[w_1, w_2]$. The values of w's are chosen from $[w_1, w_2]$ by some kind of binary search. The binary search continues until the distance between consecutive w's is less than a small threshold λ. The strategy is described in the following.

Step 1: Initially, let variables w_S and w_L indicate, respectively, the smallest value and the largest value within the given range, that is, $w_S = w_1$ and $w_L = w_2$. Use CLUSTERING with the parameter w_S to cluster the data set. Use CLUSTERING with the parameter w_L to cluster the data set.

Step 2: **Do while** $w_L - w_S > \lambda$ (λ is 0.125 in our experiments)

Begin

Let $w_m = (w_S + w_L)/2$. Use CLUSTERING with the parameter w_m to cluster the data set. Calculate the ratios $D_1(w_m)/D_1(w_S)$, $D_1(w_L)/D_1(w_m)$, $D_2(w_m)/D_2(w_S)$ and $D_2(w_L)/D_2(w_m)$. Among all subranges within the whole range $[w_1, w_2]$, find the subrange

$[w_a, w_b]$ that has the largest ratio of $D_1(w_b)/D_1(w_a)$. Let $w_S = w_a$ and $w_L = w_b$.

End

$w' = w_L$.

Step 3: Among all subranges within the whole range $[w_1, w_2]$, find the subrange $[w_a, w_b]$ that has the largest ratio of $D_2(w_b)/D_2(w_a)$. Let $w_S = w_a$ and $w_L = w_b$.

Step 4: **Do while** $w_L - w_S > \lambda$

Begin

Let $w_m = (w_S + w_L)/2$. Use CLUSTERING with the parameter w_m to cluster the data set. Calculate the ratios $D_2(w_m)/D_2(w_S)$ and $D_2(w_L)/D_2(w_m)$. Among all subranges within the whole range $[w_1, w_2]$, find the subrange $[w_a, w_b]$ that has the largest ratio of $D_2(w_b)/D_2(w_a)$. Let $w_S = w_a$ and $w_L = w_b$.

End

$w'' = w_L$.

$w''' = w_S$.

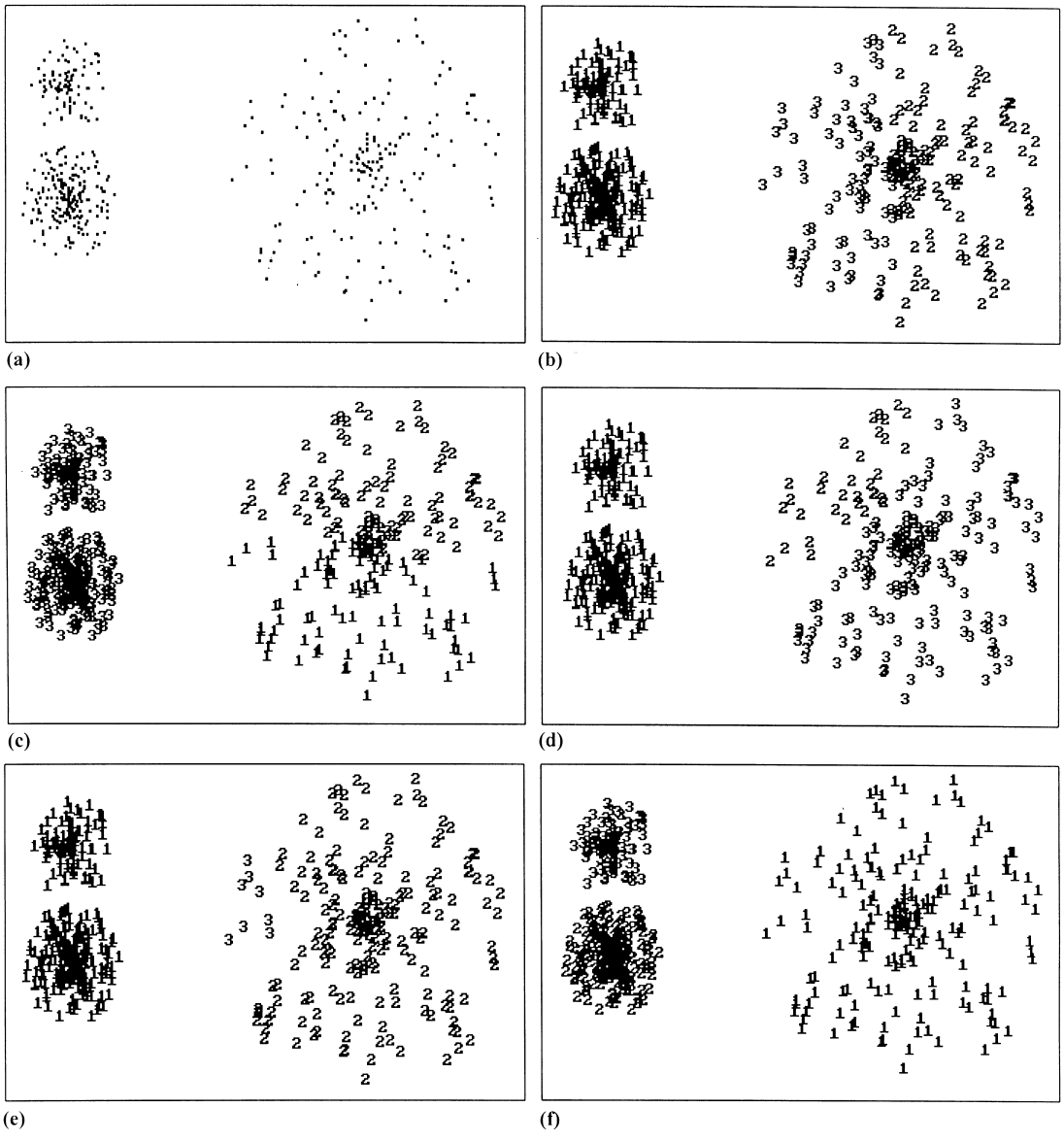

Fig. 2. An example of three clusters. (a) The original data set with three groups of points. (b) The clustering by K-means algorithm (4/10). (c) The clustering by K-means algorithm (3/10). (d) The clustering by complete-link method. (e) The clustering by single-link method. (f) The clustering by the algorithm CLUSTERING and K-means algorithm (3/10).

Step 5: **If** $w'' \geqslant w'$ **Then** Output the clustering obtained with the parameter w'.
If $w'' < w'$ **Then** Output the clustering obtained with the parameter w'''.

The heuristic strategy finds the greatest jump on the values of $D_1(w)$'s and the greatest jump on the values of $D_2(w)$'s. Based on these jumps, it then decides which a good clustering is. Three experiments in Section 5 illustrate the determination of good clusterings.

5. Experiments

In the experiments, the parameters used in the genetic algorithm are described in the following. The population

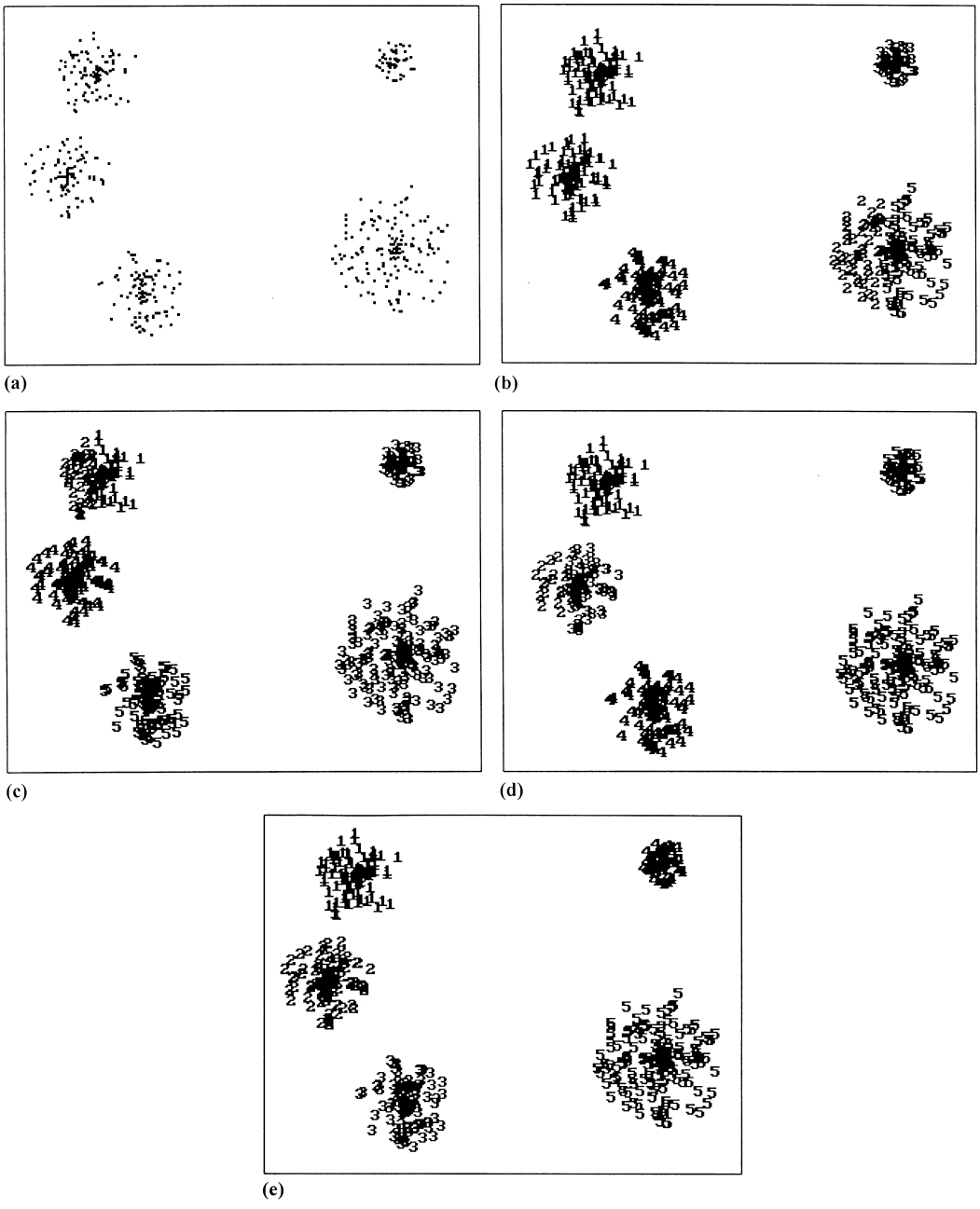

Fig. 3. An example of five clusters. (a) The original data set with five groups of points. (b) The clustering by K-means algorithm (3/10). (c) The clustering by K-means algorithm (3/10). (d) The clustering by K-means algorithm (3/10). (e) The clustering by CLUSTERING, complete-link, single-link and K-means (1/10).

Table 1
The clustering results of the first data set

		$(N(w), D_1(w), D_2(w), D_1(w_b)/D_1(w_a), D_2(w_b)/D_2(w_a))$					
u	# components	$w = 1$	$w = 1.125$	$w = 1.25$	$w = 1.5$	$w = 2$	$w = 3$
1.4	110	①(12, 8.72, 5.24, 1, 1)	⑥(12, 8.72, 5.24, 1, 1)	⑤(3, 21.28, 12.33, 2.44, 2.35)	④(3, 21.28, 12.33, 1, 1)	③(3, 21.28, 12.33, 1, 1)	②(2, 38.48, 12.33, 1.81, 1)
1.5	104	①(12, 8.72, 5.11, 1, 1)	⑥(12, 8.72, 5.11, 1, 1)	⑤(3, 21.28, 12.01, 2.44, 2.35)	④(3, 21.28, 12.01, 1, 1)	③(3, 21.28, 12.01, 1, 1)	②(2, 38.48, 12.01, 1.81, 1)
1.6	92	①(12, 8.72, 4.92, 1, 1)	⑥(12, 8.72, 4.92, 1, 1)	⑤(3, 21.28, 11.92, 2.44, 2.42)	④(3, 21.28, 11.92, 1, 1)	③(3, 21.28, 11.92, 1, 1)	②(2, 38.48, 11.92, 1.81, 1)
1.7	83	①(12, 8.72, 4.83, 1, 1)	⑥(3, 21.28, 11.85, 2.44, 2.45)	⑤(3, 21.28, 11.85, 1, 1)	④(3, 21.28, 11.85, 1, 1)	③(2, 38.48, 11.85, 1.81, 1)	②(2, 38.48, 11.85, 1, 1)
1.8	72	①(12, 8.72, 4.55, 1, 1)	⑥(3, 21.28, 11.63, 2.44, 2.56)	⑤(3, 21.28, 11.63, 1, 1)	④(3, 21.28, 11.63, 1, 1)	③(2, 38.48, 11.63, 1.81, 1)	②(2, 38.48, 11.63, 1, 1)

Table 2
The clustering results of the second data set

		$(N(w), D_1(w), D_2(w), D_1(w_b)/D_1(w_a), D_2(w_b)/D_2(w_a))$						
u	# components	$w = 1$	$w = 1.125$	$w = 1.25$	$w = 1.5$	$w = 2$	$w = 2.5$	$w = 3$
1.4	163	①(18, 18.32, 3.81, 1, 1)	⑥(12, 22.27, 4.02, 1.22, 1.06)	⑤(5, 52.66, 5.32, 2.36, 1.32)	④(5, 52.66, 5.32, 1, 1)	③(5, 52.66, 5.32, 1, 1)	⑦(4, 64.48, 8.34, 1.22, 1.57)	②(2, 78.52, 12.62, 1.22, 1.51)
1.5	163	①(18, 18.32, 3.81, 1, 1)	⑥(12, 22.27, 4.02, 1.22, 1.06)	⑤(5, 52.66, 5.32, 2.36, 1.32)	④(5, 52.66, 5.32, 1, 1)	③(5, 52.66, 5.32, 1, 1)	⑦(4, 64.48, 8.34, 1.22, 1.57)	②(2, 78.52, 12.62, 1.22, 1.51)
1.6	141	①(18, 18.32, 3.62, 1, 1)	⑥(12, 22.27, 3.75, 1.22, 1.04)	⑤(5, 52.66, 5.01, 2.36, 1.34)	④(5, 52.66, 5.01, 1, 1)	③(5, 52.66, 5.01, 1, 1)	⑦(4, 64.48, 7.99, 1.22, 1.60)	②(2, 78.52, 12.35, 1.22, 1.55)
1.7	141	①(18, 18.32, 3.62, 1, 1)	⑥(12, 22.27, 3.75, 1.22, 1.04)	⑤(5, 52.66, 5.01, 2.36, 1.34)	④(5, 52.66, 5.01, 1, 1)	③(5, 52.66, 5.01, 1, 1)	⑦(4, 64.48, 7.99, 1.22, 1.60)	②(2, 78.52, 12.35, 1.22, 1.55)
1.8	118	①(12, 22.27, 3.53, 1, 1)	⑥(12, 22.27, 3.53, 1, 1)	⑤(5, 52.66, 4.83, 2.36, 1.37)	④(5, 52.66, 4.83, 1, 1)	③(4, 64.48, 7.86, 1.22, 1.63)		②(2, 78.52, 12.02, 1.22, 1.53)

Table 3
The clustering results of the set of 20 000 spectral feature vectors

		$(N(w), D_1(w), D_2(w), D_1(w_b)/D_1(w_a), D_2(w_b)/D_2(w_a))$					
u	# components	$w = 1$	$w = 1.25$	$w = 1.5$	$w = 1.625$	$w = 1.75$	$w = 1.875$
1.4	3263	①(2212, 2112, 1.54, 1, 1)		④(1362, 2749, 1.72, 1.30, 1.12)		⑦(1123, 3410, 1.81, 1.24, 1.05)	
1.5	2672	①(1838, 2358, 1.42, 1, 1)		④(1295, 3012, 1.69, 1.23, 1.19)		⑤(882, 3873, 1.91, 1.29, 1.13)	⑦(882, 3873, 1.91, 1, 1)
1.6	2111	①(1546, 2549, 1.52, 1, 1)	⑦(1285, 3026, 1.69, 1.19, 1.11)	④(1084, 3519, 1.78, 1.16, 1.05)		⑤(892, 3873, 2.03, 1.10, 1.14)	⑧(892, 3873, 2.03, 1, 1)
1.7	1628	①(1130, 3390, 1.49, 1, 1)		⑤(948, 3650, 1.64, 1.08, 1.10)		⑦(882, 3873, 1.82, 1.06, 1.11)	⑧(627, 5183, 2.18, 1.34, 1.20)
1.8	1282	①(922, 3748, 1.47, 1, 1)		④(892, 3873, 1.82, 1.03, 1.24)	⑦(892, 3873, 1.82, 1, 1)	⑥(633, 5174, 2.05, 1.34, 1.13)	⑧(633, 5174, 2.05, 1, 1)

size is 50, the crossover rate is 80% and the mutation rate is 5%. 100 generations were run and the best solution was retained. The smallest value w_1 and the largest value w_2 of the parameter w were set to 1 and 3, respectively. Three sets of data were used in our experiments. The first set of data consists of three groups of points on the plane as shown in Fig. 2(a). The sizes of three groups are 100, 200 and 200. The densities of three groups are not the same. The clustering results of the algorithm CLUSTERING are shown in Table 1.

Several different values of u were applied to illustrate that a good clustering can be found no matter what the exact value of u is. That is, as long as u takes its value within a proper range, the exact value of u is not important. In this paper, five values of u, namely 1.4, 1.5, 1.6, 1.7 and 1.8, are used to illustrate that with a suitable choice of the value of w, a good clustering can be found with all five values of u. This means that u may be chosen from the interval [1.4, 1.8], and the exact value of u is not crucial to the clustering result.

In Table 1, the column "# of components" denotes the number of connected components obtained in the first stage of the algorithm. The numeral in a circle before the parenthesis denotes the sequence when conducting the binary search on the values of parameter w. The first value in the parenthesis, $N(w)$, represents the number of clusters obtained in the second stage of the algorithm. Since w'' equals w' in all the cases shown in Table 1 (that is, $w'' = w' = 1.25$ for $u = 1.4$, 1.5, 1.6 and $w'' = w' = 1.125$ for $u = 1.7$, 1.8), according to the heuristic strategy, the best clustering found is the clustering with three clusters. Fig. 2(f) depicts this clustering.

For comparison, three clustering methods, the K-means algorithm, the complete-link method [18] and the single-link method [18], were also applied to this data set. Fig. 2(d) depicts the clustering result obtained by the complete-link method. Fig. 2(e) depicts the clustering result obtained by the single-link method. Both clustering results are not good. These two methods are not suitable in clustering data with different densities. The K-means algorithm was applied ten times with the number of clusters being set to three. Figs. 2(b), (c) and (f) show, respectively, the clustering results of the four, three and the other three times of application of the K-means algorithm. For K-means algorithm, only 30% of clustering results were good even when the number of clusters was known in advance.

The second set of data as shown in Fig. 3(a) consists of five groups of points on the plane. The sizes of five groups are 100, 100, 100, 40 and 160, respectively. The densities of five groups are similar. Table 2 expresses the clustering results of the algorithm CLUSTERING. In Table 2, w' is smaller than w'' for all cases. That is, $w' = 1.25$ for all u, $w'' = 2$ for $u = 1.8$ and $w'' = 2.5$ for all other u. Therefore, the best clustering found is that with five clusters. This clustering result is depicted in Fig. 3(e). Both the complete-link method and the single-link method worked well this time. Their clustering results are also shown in Fig. 3(e). The K-means algorithm was applied 10 times on this data set. But the result was not good. Only 10% of the clustering results were good even when the number of clusters was known in advance, as expressed in Figs. 3(b)–(e). The choice of initial centers in the K-means algorithm affects the clustering result very much.

The last data set is a large one. It contains 20 000 spectral feature vectors derived from 40 speeches in the TIMIT database [19]. Each vector contains 64

Table 3 (*Continued*)

$w = 2$	$w = 2.125$	$w = 2.25$	$w = 2.375$	$w = 2.5$	$w = 2.75$	$w = 3$
③(892, 3873, 2.03, 1.14, 1.12)	⑧(627, 5183, 2.31, 1.34, 1.14)	⑥(627, 5183, 2.31, 1, 1)	⑨(553, 5527, 2.75, 1.07, 1.19)	⑤(553, 5527, 2.75, 1.07, 1)		②(505, 6843, 3.18, 1.24, 1.16)
③(633, 5174, 2.21, 1.34, 1.16)		⑧(633, 5174, 2.21, 1, 1)	⑨(548, 5592, 2.64, 1.08, 1.19)	⑥(538, 5633, 2.85, 1.01, 1.08)		②(486, 7022, 3.12, 1.25, 1.09)
③(633, 5174, 2.21, 1.34, 1.09)	⑩(633, 5174, 2.21, 1, 1)	⑨(548, 5592, 2.64, 1.08, 1.19)		⑥(538, 5633, 2.85, 1.09, 1.08)		②(412, 7370, 2.98, 1.31, 1.05)
③(627, 5183, 2.18, 1, 1)	⑩(550, 5633, 2.64, 1.09, 1.21)	⑨(550, 5633, 2.64, 1, 1)		④(532, 5633, 2.74, 1, 1.04)	⑥(412, 7370, 2.79, 1.31, 1.02)	②(377, 8102, 2.96, 1.10, 1.06)
③(548, 5592, 2.58, 1.08, 1.26)				⑤(456, 7129, 2.63, 1.27, 1.02)		②(350, 8232, 2.95, 1.15, 1.12)

Table 4
The comparison of four methods

Methods	(# of clusters, average distance from center)	
K-means	(627, 5239)	(633, 5208)
Single link	(627, 5673)	(633, 5512)
Complete link	(627, 5425)	(633, 5317)
CLUSTERING	(627, 4328)	(633, 4282)

components. Table 3 shows the clustering results of this set of 20 000 spectral feature vectors. In this experiment, the good clusterings obtained by the algorithm CLUSTERING and the heuristic strategy are that with 627 clusters (for $u = 1.4$ and 1.7) and that with 633 clusters (for $u = 1.5$, 1.6 and 1.8). For comparison, the *K*-means algorithm, the single-link method and the complete-link method were also applied to this data set to produce both 627 and 633 clusters. The average distance from center was used as an indication for comparison. The average distance from center means the average of the distances of all data points to their cluster centers. From Table 4, the average distance from center is much smaller for the clustering results of the algorithm CLUSTERING.

6. Conclusions

In this paper, we propose a genetic algorithm for the clustering problem. The proposed algorithm CLUSTERING with the heuristic strategy is an automatic clustering algorithm. The first stage of CLUSTERING uses the nearest-neighbor clustering method to group those data that are close to one another. At the end of the first stage, a set of small clusters is obtained. The second stage is a genetic clustering algorithm. This algorithm will group the small clusters into larger clusters. A heuristic strategy is then used to find a good clustering. The main purpose of the first stage is to reduce the size of the data to a moderate one that is suitable for the genetic clustering algorithm in the second stage. If the initial data set is not large, the first stage can be omitted. The experimental results show that CLUSTERING is effective.

Unlike other clustering algorithms, the proposed algorithm CLUSTERING with the heuristic strategy can automatically search for a proper number as the number of clusters. From our experience, for almost all kinds of data, a good clustering can be found by setting u to 1.4, 1.5, 1.6, 1.7 or 1.8 in the first stage and binary searching the value of the parameters w within the interval [1,3] in the second stage. In fact, Tables 1–3 can give us much information about what a good clustering is. Although one may apply the *K*-means algorithm many times in order to find a good clustering, for a clustering problem with a large data set (for example, in the third experiment which has 20 000 feature vectors), it is not easy for the user to guess how many clusters should be there. The *K*-means algorithm will not produce a good clustering if the number of clusters given by the user is not proper.

References

[1] M.R. Anderberg, Cluster Analysis for Applications, Academic Press, New York, 1973.
[2] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addision-Wesley, Reading, MA, 1974.

[3] J.A. Hartigan, Clustering Algorithms, Wiley, New York, 1975.
[4] K.S. Fu, Communication and Cybernetics: Digital Pattern Recognition, Springer, Berlin, 1980.
[5] R. Dubes, A.K. Jain, Clustering Methodology in Exploratory Data Analysis, Academic Press, New York, 1980.
[6] P.A. Devijver, J. Kittler, Pattern Recognition – A Statistical Approach, Prentice-Hall, London, 1982.
[7] S.Z. Selim, M.A. Ismail, *K*-means-type algorithm: generalized convergence theorem and characterization of local optimality, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1984) 81–87.
[8] W.L. Koontz, P.M. Narendra, K. Fukunaga, A branch and bound clustering algorithm, IEEE Trans. Comput. c-24 (1975) 908–915.
[9] S.Z. Selim, K.S. Al-Sultan, A simulated annealing algorithm for the clustering problem, Pattern Recognition 24 (1991) 1003–1008.
[10] R.W. Klein, R.C. Dubes, Experiments in projection and clustering by simulated annealing, Pattern Recognition 22 (1989) 213–220.
[11] G.P. Babu, M.N. Murty, Clustering with evolution strategies, Pattern Recognition 27 (1994) 321–329.
[12] R.F. Ling, A probability theory of cluster analysis, J. Amer. Statist. Assoc. 68 (1973) 159–164.
[13] P.-Y. Yin, L.-H. Chen, A new non-iterative approach for clustering, Pattern Recognition Lett. 15 (1994) 125–133.
[14] G.C. Osbourn, R.F. Martinez, Empirically defined regions of influence for clustering analyses, Pattern Recognition 28 (1995) 1793–1806.
[15] P.S. Stephen, Threshold validity for mutual neighborhood clustering, IEEE Trans. Pattern Anal. Mach. Intell. 15 (1993) 89–92.
[16] M. Srinivas, M. Patnaik, Genetic algorithm – A survey, IEEE Computer 27 (1994) 17–26.
[17] C.A. Murthy, N. Chowdhury, In search of optimal clusters using genetic algorithms, Pattern Recognition Lett. 17 (1996) 825–832.
[18] R. Dubes, A.K. Jain, Clustering techniques: the user's dilemma, Pattern Recognition 8 (1976) 247–260.
[19] DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus, National Institute of Standards and Technology Speech Disc 1-1.1, 1990.

About the Author—LIN YU TSENG received the B.S. degree in mathematics from the National Taiwan University, Taiwan, in 1975, and the M.S. degree in Computer Science from the National Chiao Tung University, Taiwan, 1978. After receiving the M.S. degree, he worked in industry and taught at the university for several years. He received the Ph.D degree in Computer Science from National Tsing Hua University, Taiwan, in 1988. He is presently a Professor of the Department of Applied Mathematics, National Chung Hsing University. His research interests include pattern recognition, document processing, speech coding and recognition, neutral networks and algorithm design.

About the Author—SHIUENG BIEN YANG received the B.S. degree in 1993 from the Department of Applied Mathematics, National Chung Hsing University. He is now working towards the Ph.D. degree in the same department. His research interests include pattern recognition, speech coding and recognition, image coding and neural networks.