K-Means-Based Fuzzy Classifier Design

Ching-Chang Wong, Chia-Chong Chen, and Shih-Liang Yeh

Department of Electrical Engineering, Tamkang University
Tamshui, Taipei Hsien, Taiwan, Republic of China

E-mail: wong@ee.tku.edu.tw

***Abstract*-In this paper, a method based on the K-means algorithm is proposed to efficiently design a fuzzy classifier so that the training patterns can be correctly classified by the proposed approach. In this method, the K-means algorithm is first used to partition the training data for each class into several clusters, and the cluster center and the radius for each cluster are calculated. Then, a fuzzy system design method that uses a fuzzy rule to represent a cluster is proposed such that a fuzzy classifier can be efficiently constructed to correctly classify the training data. The proposed method has the following features: (a) it does not need prior parameter definition; (b) it only needs a short training time; and (c) it is simple. Finally, two examples are used to illustrate and examine the proposed method for the fuzzy classifier design.**

Index Terms- Fuzzy classifier, K-means algorithm.

I. INTRODUCTION

Pattern classification is an important element of many engineering problems such as handwriting recognition, diagnostic etc. Because of the wide range of applicability, pattern classification has been studied in a great deal. On the other hand, fuzzy system was introduced by Zadeh as a means of representing and manipulating data that are not precise, but rather fuzzy. Therefore, fuzzy classifiers provide a measure of the degree to which a pattern fits within a class. Many real-world problems require this property. One example is to develop a machine to recognize the gray levels of images. In traditional approaches, it is difficult to construct a classifier with every degree of gray level, but using a fuzzy classifier, the degree of black or white can be easily determined. In 1992, a fuzzy classifier with hyperbox regions was discussed by Simpson [1]. The classifier has short training time for it only needs to calculate the max-min of training data. But the classification rate of this method will decrease when there are identical data in different classes. Besides, this method has to choose an appropriate hyperbox size. In general, the hyperbox size should be small so that many fuzzy rules will be generated by this method for the fuzzy classifier design. In 1995, a fuzzy classifier based on genetic algorithm is proposed by Ishibuchi et al. [2]. The training time of fuzzy classifier based on genetic algorithm is long and the individual length becomes very long when the training data are high dimensional. In 1997, a fuzzy classifier with ellipsoidal region was proposed by Abe and Thawonmas [3]. The fuzzy classifier with ellipsoidal region has good performance in many classification problems, but it needs vast time to calculate the covariance matrix of the ellipsoidal clusters. In order to avoid the above drawbacks, a simple and systematic fuzzy classifier design method based on K-means algorithm is proposed to efficiently construct a fuzzy system to correctly classify the training data.

The rest of this paper is organized as follows. In Section II, the K-means algorithm is described. In Section III, the training algorithm of the proposed method is described. In Section IV, a two-class example is first used to illustrate the training algorithm, then the iris data is used to examine the performance of the proposed fuzzy classifier. In Section V, a summary of K-means based fuzzy classifier as well as a discussion of future work is described.

II. K-MEANS ALGORITHM

The objective of clustering algorithms is to cluster the given data set into several groups such that data within a group are more similar than those outside the group. Achieving such a partition requires a similarity measure that considers two vectors and returns a value reflecting their similarity. Since most similarity measures are sensitive to the ranges of elements in the input vectors, each of the input variables must be normalized to be within the unit interval. Hence, we assume the given data set has been normalized to be within the unit hypercube. One of the widely used clustering algorithms is the K-means algorithm. The K-means algorithm is described in the following [4].

Let $X=\{x_1, x_2, \ldots x_n\}$ be a given set of n vectors in a p-dimensional feature space, where $x_j=(x_j(1), x_j(2), \cdots, x_j(p)) \in R^P$. The K-means algorithm partitions the given data set into c clusters and calculates cluster centers $V=\{m_1, m_2, \cdots, m_c\}$ so that the following objective function can

0-7803-5877-5/00/$10.00 © 2000 IEEE

be minimized.

$$J=\sum_{i=1}^{c}J_i=\sum_{i=1}^{c}\left(\sum_{x_j\in G_i}d_{ij}^2\right)=\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}d_{ij}^2, \quad (1)$$

$J_i=\sum_{x_j\in G_i}d_{ij}^2$ is the cost function within group i, G_i represents the data set belonging to class i, d_{ij} represents the geometric properties (dissimilarity) between i-th cluster center m_i and j-th data point x_j, and u_{ij} is defined as follows:

$$u_{ij}=\begin{cases}1, & if\ \|x_j-m_i\|\le\|x_j-m_k\|,\ for\ k\ne i\\ 0, & otherwise\end{cases} \quad (2)$$

That is, the value of u_{ij} is either 1 or 0. The value $u_{ij}=1$ represents that the datum x_j belongs to group i, and $u_{ij}=0$ otherwise. Consequently, u_{ij} should satisfy the following two conditions:

$$\sum_{i=1}^{c}u_{ij}=1 \quad \forall j=1,\dots,n \quad (3)$$

and

$$\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}=n. \quad (4)$$

When u_{ij}, $i=1,2,..,c$, $j=1,2,..,n$, is determined, the cluster center m_i that minimize (1) is the mean of the data belonging to cluster i and described by

$$m_i=\frac{1}{|G_i|}\sum_{x_j\in G_i}x_j, \quad (5)$$

where $|G_i|$ represents the number of the data belonging to cluster i. To explain more clearly, the procedures of the K-means algorithm are described as follows:

Step 1: Initialize the cluster center $m_i, i=1,\dots c$. This is typically achieved by randomly selecting c points from the data points.

Step 2: Determine u_{ij}, $i=1,2,..,c$, $j=1,2,..,n$, by (2).

Step 3: Compute the cost function according to (1). Stop if either it has converged or the improvement is below a threshold.

Step 4: Update the cluster center m_i using (5), then go to Step 2.

The K-means algorithm is a simple and systematic algorithm for data clustering. For this reason, we choose K-means algorithm to cluster the training data. On the other hand, according to the different shapes of the training data set (e.g. spherical or ellipsoidal) we can choose different distance measure (e.g., Euclidean, Mahalanobis, Hamming, etc.) to partition the training data set. In this paper, we use Euclidean distance as the dissimilarity measure so that the overall cost function can be expressed by

$$J=\sum_{i=1}^{c}\left(\sum_{x_j\in G_i}\|x_j-m_i\|^2\right), \quad (6)$$

where $\|x_j-m_i\|$ represents the Euclidean distances between x_j and m_i. Namely, the data will be partitioned into several hyperspherical regions, and then we map each hyperspherical region to a fuzzy rule. The detailed training algorithm for generating rules based on the information obtained by the K-means algorithm will be discussed in Section III.

III. K-MEANS-BASED FUZZY CLASSIFIER

In this section, the following nomenclatures are adopted. The fuzzy classifier design can be divided into two main steps: 1) cluster generation, and 2) fuzzy rules extraction. These two steps are described in the following two sub-sections.

- U_l: Data set belonging to class l.
- U_l': Updated data set belonging to class l.
- G_{li}: Data set belonging to i-th cluster for l-th class.
- m_{li}: Cluster center of i-th cluster for l-th class.
- r_{li}: Radius of i-th cluster for l-th class.
- d_diff_{li}: Minimum distance between the i-th cluster center of l-th class and the data in different classes.
- $Record_{lk}$: Data set of the successful k-th cluster for l-th class.
- $Record\_Center_{lk}$: Cluster center of the k-th successful cluster for l-th class.
- $Record\_Radius_{lk}$: Radius of the k-th successful cluster for l-th class.

A. Cluster Generation Using K-means Algorithm

In this sub-section, a method based on K-means clustering algorithm is proposed to partition the training data into several clusters. Assume the training data has been normalized in a unit hypercube, and the training data $X = \{x_1, x_2, \ldots x_n\}$, $x_j \in R^p$ contains n vectors in a p-dimensional feature space, where $x_j = (x_j(1), x_j(2), \cdots, x_j(p))$ belongs to one of s classes. The proposed method can be described by the following steps.

Step 1: For class $l\ (l = 1,2,..,s)$, we can get the data set $U_l(l = 1,2,...,s) = \{x_j | x_j \in class\ l\}$ and $U_l^{'}(l = 1,2,...,s) = \{x_j | x_j \in class\ l\}$ ($U_l^{'} = U_l$ in the initial state). Initialize parameters $l = 1, i = 1, k = 1$.

Step 2: For class $l = 1$, set $c = 1$ for K-means algorithm.

Step 3: Partition $U_l^{'}$ by K-means into c clusters and obtain the following information:

1. Data set $G_{li} = \{x_j | x_j \in U_l^{'}, u_{ij} = 1\}$.
2. Cluster center $m_{li} = \frac{1}{|G_{li}|} \sum_{x_j \in G_{li}} x_j$.
3. Cluster radius $r_{li} = \max_{x_j \in G_{li}} \|x_j - m_{li}\|$.
4. Distance from i-th cluster center of l-th class to the nearest different class datum $d\_diff_{li} = \min_{x_j \notin U_l} \|x_j - m_{li}\|$.

   In a special case, when only one datum belongs to cluster G_{li}, we define the radius r_{li} of the cluster $= 0.1 * d\_diff_{li}$.

Step 4: Check every data set belonging to class l ($G_{li}\ (i = 1,2,..,c)$):
If $r_{li} < d\_diff_{li}$, then cluster G_{li} has been successfully partitioned, record G_{li} by assigning $Record_{lk} = G_{li}$, $Record\_Center_{lk} = m_{li}$, $Record\_Radius_{lk} = r_{li}$, $v_{lk} = |G_{li}|$, $k = k + 1$, and remove G_{li} from $U_l{'}$.
Else $r_{li} \geq d\_diff_{li}$, then fail partition; do not record this cluster.

Step 5: If $U_l^{'}$ is empty, go to Step 6.
Else, check the conditions described as below:
If these are any successful partition in Step 4, we assume $c = 1$ then go to Step 3.
Else $c = c + 1$, then go to Step 3.

Step 6: When $U_l^{'}$ is empty, the data belonging to class l has been partitioned successfully; then if $l < s$, we change class $l = l + 1$ and initialize parameters $i = 1, k = 1, c = 1$, and then go to Step 3. Else means all the classes have been considered ($l = s$), the cluster algorithm stops.

After the partition process, the training data in each class could be partitioned into several circular regions given by $Record_{lk}$. In addition, there are some advantages of the clustering algorithm:

1) Owing to the check of r_{li} and d_diff_{li} in Step 5, the generated spherical regions will not contain the training data of the different class. It guarantees the training data can be 100% classified into the correct class, provided there are no identical data in different classes.

2) Each circular partition $Record_{lk}$ is associated with a value $v_{lk} = |Record_{lk}|$, which denotes the "total" number of data belonging to the partition. We can judge the importance of the partition by ordering v_{lk}, where the larger v_{lk} means the more important partition.

B. *Fuzzy Rules Extraction*

After the cluster algorithm, the cluster centers and radius of the clusters $Record_{lk}$ are recorded as $Record\_Center_{lk}$ and $Record\_Radius_{lk}$. For each cluster, a fuzzy IF-THEN rule is used to represent it and is described by

$$R_{lk}:\ IF\ \ x\ is\ A_{lk}\ \ THEN\ \ x\ belongs\ to\ class\ l \tag{7}$$

where A_{lk} is a fuzzy set described by the following membership function:

$$A_{lk}(x) = exp\left(\frac{-\|x - Record\_Center_{lk}\|^2}{2Record\_Radius_{lk}^{\ 2}}\right), \tag{8}$$

where x is a p-dimensional data. The membership value of input x for A_{lk} is denoted by $g_{lk} = A_{lk}(x)$, when x is the same as $Record\_Center_{lk}$, $g_{lk} = 1$ and while x moves away from $Record\_Center_{lk}$, g_{lk} will decrease to approximate 0.

After the rule extraction, we combine these rules to construct a fuzzy classifier. As an example, when the membership value to each rule is calculated corresponding to the input x, and the maximum value $g_{l^*k^*}$ is determined, then x is classified into the class l^*.

IV. Simulation Results

In this section, we first give an example to illustrate the proposed method and then use the iris data to examine the performance of the proposed method.

Consider a two-dimensional, two-class classification problem shown in Fig. 1, where the symbols 'x' and 'o' denote the data belonging to class 1 and class 2, respectively. We want to construct a fuzzy classifier to correctly classify these two classes. First, we use the clustering algorithm proposed in Section III to partition the training data. The results are shown in Fig. 2(a)-(e), where the training data were partitioned into eight clusters. We see that the different class training data do not exist in the same spherical region. A fuzzy rule is used to represent each cluster, and then eight fuzzy rules are described to construct the fuzzy classifier. According to the classifier, a boundary between the two classes can be sketched (see Fig. 2(f)), where the pattern lies in the left side of the boundary are classified into class 1 and else class 2.

Fisher's Iris data [5] is a well-known test pattern for classification; the Iris data consists of 150 four-dimensional vectors belonging to three classes. To examine the performance of proposed method, the training data are composed of the first 25 data of each class, and the test data are composed of the remaining 25 data of each class. The simulation result is listed in Table I. It illustrates that the training data can be guaranteed 100% classified by the design fuzzy classifier. In addition, the test data can achieve 96% classified rate by the generated fuzzy classifier with eight fuzzy rules so that the proposed design method has good generalization ability in the classification problem of iris data.

V. Conclusions

In this paper, we propose a K-means based fuzzy classifier, which has several advantages such as simple and systematic structure. In the cluster generation process, a clustering algorithm based on K-means is proposed to cluster the data such that the training data can be guaranteed 100% classified by the fuzzy classifier. In addition, it has good generalization ability and can achieve 96% classified rate in the classification problem of iris data. Our future work consists of the following two parts: Part 1: Fuzzy rule elimination. For example in Section III, we have described a parameter v_{ik}, which represents the cluster size. In general, the cluster that contains more data is more important. Thus, we can simplify the classifier architecture by removes the clusters that have smaller v_{ik}. Part 2: Classification improvement. In this paper, we do not mention fuzzy rule tuning, because the objective of this paper is to propose a simple and systematic approach for constructing a construct fuzzy classifier. In the future, we will aim at rule adjustment to improve the classification rate.

Acknowledgment

This research was supported in part by the National Science Council of the Republic of China under contract NSC-88-2213-E-032-024.

References

[1] P. K. Simpson, "Fuzzy Min-Max Neural Networks-Part 1: Classification," *IEEE Trans. Neural Networks*, vol.3, no. 5, pp.776-786, 1992.

[2] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms," *IEEE Trans. Fuzzy Systems*, vol.3, no. 3, pp.260-270, 1995.

[3] S. Abe, and R. Thawonmas, "A Fuzzy Classifier with Ellipsoidal Regions," *IEEE Trans. Fuzzy Systems*, vol.5, no. 3, pp.358-368, 1997.

[4] J. S. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Prentice Hall, New Jersey, 1997.

[5] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annal Eugenics*, vol. 7, pp. 179-188, 1936.

TABLE I.
Classification Rate of The Proposed Classifier with 8 Fuzzy Rules

Data set	Number of misclassify patterns	Classification rate
Training data	0	100%
Test data	3	96%

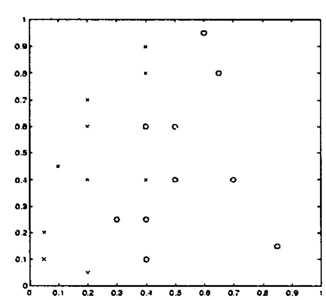

Fig. 1. Two-class classification problem

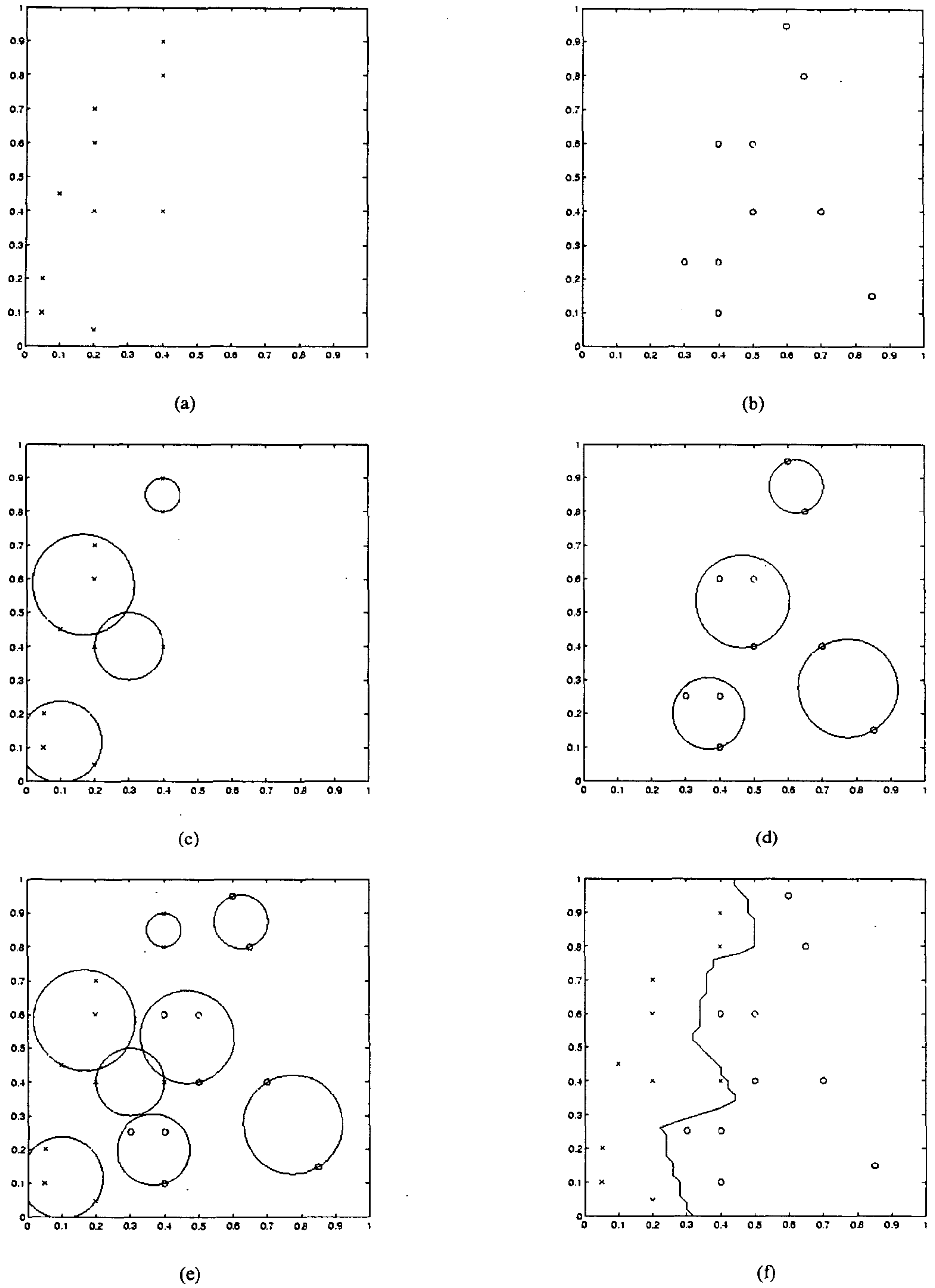

Fig. 2. The clustering and classification results determined by the proposed method. (a) The training data belonging to class 1. (b) The training data belonging to class 2. (c) The clustering results for class 1. (d) The clustering results for class 1. (e) All clusters for the two class (Each circle represents a fuzzy rule). (f) The boundary determined by the designed fuzzy classifier.