ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/3413496

An efficient fuzzy classifier with feature selection based on fuzzy entropy

Article *in* IEEE TRANSACTIONS ON CYBERNETICS · July 2001

DOI: 10.1109/3477.931536 · Source: IEEE Xplore

CITATIONS
182

READS
137

4 authors, including:

Chih-Ming Chen
National Chengchi University
98 PUBLICATIONS **2,296** CITATIONS

SEE PROFILE

Jyh-Ming Chen
11 PUBLICATIONS **289** CITATIONS

SEE PROFILE

All content following this page was uploaded by Jyh-Ming Chen on 11 March 2015.

The user has requested enhancement of the downloaded file. All in-text references underlined in blue are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

An Efficient Fuzzy Classifier with Feature Selection Based on Fuzzy Entropy

Hahn-Ming Lee, Chih-Ming Chen, Jyh-Ming Chen, and Yu-Lu Jou

***Abstract*—This paper presents an efficient fuzzy classifier with the ability of feature selection based on a fuzzy entropy measure. Fuzzy entropy is employed to evaluate the information of pattern distribution in the pattern space. With this information, we can partition the pattern space into nonoverlapping decision regions for pattern classification. Since the decision regions do not overlap, both the complexity and computational load of the classifier are reduced and thus the training time and classification time are extremely short. Although the decision regions are partitioned into nonoverlapping subspaces, we can achieve good classification performance since the decision regions can be correctly determined via our proposed fuzzy entropy measure. In addition, we also investigate the use of fuzzy entropy to select relevant features. The feature selection procedure not only reduces the dimensionality of a problem but also discards noise-corrupted, redundant and unimportant features. Finally, we apply the proposed classifier to the Iris database and Wisconsin breast cancer database to evaluate the classification performance. Both of the results show that the proposed classifier can work well for the pattern classification application.**

***Index Terms*—Feature selection, fuzzy classifier, fuzzy entropy.**

I. Introduction

The objective of this paper is to propose an efficient fuzzy classifier with feature selection capability. A fuzzy entropy measure is employed to partition the input feature space into decision regions and to select relevant features with good separability for the classification task. For a classification problem, the methods of enclosing the decision regions and the dimensionality of the problem have profound effects on classification speed and performance. In this paper, nonoverlapping fuzzy decision regions are generated. Although the decision regions do not overlap, we can still obtain correct boundaries from fuzzy decision regions to achieve good classification performance. Furthermore, since the decision regions do not overlap, both the computational load and complexity of the classifier are reduced and the classification speed would be extremely fast. In addition, the proposed feature selection method based on the fuzzy entropy increases the classification rate by discarding noise-corrupted, redundant, and unimportant features.

Two important issues in developing a classifier are reducing the time consumed and increasing the classification rate [1]. Various methods have been proposed to develop fuzzy classifiers [1]–[3]. Employing neural networks to construct a classifier is an easy method. However, it takes much time to train the neural network [1]. Also, the results are hard to analyze and thus it is difficult to improve the performance of the trained network. To solve these problems, researchers have proposed many methods based on fuzzy region analysis to construct fuzzy classifiers [1], [2], [4]–[8]. Generally, the "shapes" of the fuzzy regions can be classified into the following types:

1) hyperbox regions [2], [4], [6], whose boundaries are parallel to the input axes;
2) ellipsoidal regions [1] and [9];
3) polyhedron regions [5] whose boundaries are expressed by a linear combination of input variables.

A classifier using hyperbox regions is easier and less time consuming than those using ellipsoidal regions or polyhedron regions [1].

Our proposed method is based on hyperbox regions, but the fuzzy regions do not overlap as those in [2]. We use a fuzzy entropy measure instead of fuzzy rules to reflect the actual distribution of classification patterns by computing the fuzzy entropy of each feature dimension. The decision region can be tuned automatically according to the fuzzy entropy measure. Also, due to the ability to detect the actual distribution of patterns, the generated decision regions can be effectively reduced.

Since recent classification systems aim to deal with larger and more complex tasks, the problem of feature selection is becoming increasingly important [10], [11]. Generally, there are two types of feature selection: 1) selecting a relevant feature subset from the original feature set and 2) transforming or combining the original features into new features, which is referred to as *feature extraction* [12], [13].

Feature extraction increases the ability to find new features to describe the pattern space [13]. It is especially useful when the original features do not adequately separate the classes, i.e., we cannot select an optimal feature subset directly from the given feature set. However, the feature extraction method requires extra effort and the transformed features may lose the physical meaning of the original features. In contrast, the feature selection approach keeps the physical meaning of the selected features. For some rule-based systems, retaining the physical meaning of features is important and indispensable [13].

Feature selection has been studied for years and many approaches have been proposed [10], [11], [14], [15]. Many of the proposed methods are based on neural network systems [16]–[20]. However, feature selection based on neural networks is time-consuming. Moreover, the selected features may not be interpretable and may not be suitable for other classification systems. Similarly, the genetic algorithm-based approach to feature selection also has the time-consuming problem [21]–[24]. In order to avoid this problem, we propose an efficient method for selecting relevant features based on fuzzy entropy. Various definitions of fuzzy entropy have been proposed [25], [26]. Basically, a well-defined fuzzy entropy measure must satisfy the four Luca–Termini axioms [25], which will be described in Section II. Our proposed fuzzy entropy is derived from Shannon's entropy [27] and satisfies these four axioms.

The paper is organized as follows. Section II recalls the Shannon's entropy and presents our proposed fuzzy entropy. Section III describes the operation of the fuzzy classifier, including the determination of the number of intervals, the center and width of each interval, and the decision region partition. Section IV presents the feature selection process, and some experimental results are illustrated in Section V. The discussion and the conclusion are given in Section VI.

II. Entropy Measure

Entropy is a measure of the amount of uncertainty in the outcome of a random experiment, or equivalently, a measure of the information obtained when the outcome is observed. This concept has been defined in various ways [25]–[30] and generalized in different applied fields, such as communication theory, mathematics, statistical thermodynamics, and economics [31]–[33]. Of these various definitions,

Manuscript received January 24, 1999; revised January 9, 2001. This work was supported in part by the National Science Council of Taiwan under Grant NSC-89-2213-E-003-005. This paper was recommended by Associate Editor T. Sudkamp.

H.-M. Lee is with the Department of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C. (e-mail: hmlee@et.ntust.edu.tw).

C.-M. Chen and J.-M. Chen are with the Institute of Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C. (e-mail: ming@neuron.et.ntust.edu.tw; jmchen@neuron.et.ntust.edu.tw;).

Y.-L. Jou is with the INFOLIGHT Technology Corporation, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C. (e-mail:leo@neuron.et.ntust.edu.tw).

Publisher Item Identifier S 1083-4419(01)04860-9.

1083–4419/01$10.00 © 2001 IEEE

Shannon contributed the broadest and the most fundamental definition of the entropy measure in information theory. In what follows, we will first introduce Shannon's entropy and then describe the four Luca–Termini axioms [25] that a well- defined fuzzy entropy measure must satisfy. Finally, we will propose a fuzzy entropy measure which is an extension of Shannon's definition.

A. Shannon's Entropy

Entropy can be considered as a measure of the uncertainty of a random variable X. Let X be a discrete random variable with a finite alphabet set containing N symbols given by $\{x_0, x_1, \ldots, x_{N-1}\}$. If an output x_j occurs with probability $p(x_j)$, then the amount of information associated with the known occurrence of output x_j is defined as

$$I(x_j) = -\log_z\, p(x_j). \tag{1}$$

That is, for a discrete source, the information generated in selecting symbol x_j is $\lceil -\log_2 p(x_j) \rceil$ bits. On average, the symbol x_j will be selected $n \cdot p(x_j)$ times in a total of N selections, so the average amount of information obtained from n source outputs is

$$\begin{aligned} -n \cdot p(x_0) \log_2 p(x_0) - n \cdot p(x_1) \log_2 p(x_1) - \cdots \\ - n \cdot p(x_{N-1}) \log_2 (x_{N-1}). \end{aligned} \tag{2}$$

Dividing (2) by n, we obtain the average amount of information per source output symbol. This is known as the average information, the uncertainty, or the entropy, and is defined as follows.

Definition 1: The entropy $H(X)$ of a discrete random variable X is defined as [27]

$$H(X) = -\sum_{j=0}^{N-1} p(x_j) \log_2 p(x_j) \tag{3}$$

or

$$H(X) = -\sum_{j=0}^{N-1} p_j \log_2 p_j \tag{4}$$

where p_j denotes $p(x_j)$.

Note that entropy is a function of the distribution of X. It does not depend on the actual values taken by the random variable X, but only on the probabilities. Hence, entropy is also written as $H(p)$.

B. Luca–Termini Axioms for Fuzzy Entropy

Kosko [25] proposed that a well-defined fuzzy entropy measure must satisfy the four Luca–Termini axioms. They include the following:

1) $E(A) = 0$ iff $A \in 2^X$, where A is a nonfuzzy set and 2^X indicates the power set of set A.
2) $E(\tilde{A}) = 1$ iff $m_A(x_i) = 0.5$ for all i, where $m_{\tilde{A}}(x_i)$ indicates the membership degree of the element x_i to fuzzy set $\tilde{A}$.
3) $E(\tilde{A}) \le E(\tilde{B})$ if $\tilde{A}$ is less fuzzy than $\tilde{B}$, i.e., if $m_{\tilde{A}}(x) \le m_{\tilde{B}}(x)$ when $m_{\tilde{B}}(x) \le 0.5$ and $m_{\tilde{A}}(x) \ge m_{\tilde{B}}(x)$ when $m_{\tilde{B}}(x) \ge 0.5$, where $\tilde{A}$ and $\tilde{B}$ are fuzzy sets.
4) $E(A) = E(A^C)$.

C. Fuzzy Entropy

Our proposed fuzzy entropy is based on Shannon's entropy and is defined as follows:

Definition 2: Fuzzy Entropy of an Interval for Each Feature Dimension:

1) Let $X = \{r_1, r_2, \ldots, r_n\}$ be a universal set with elements r_i distributed in a pattern space, where $i = 1, 2, \ldots, n$.
2) Let $\tilde{A}$ be a fuzzy set defined on an interval of pattern space which contains k elements ($k < n$). The mapped membership degree of the element r_i with the fuzzy set $\tilde{A}$ is denoted by $\mu_{\tilde{A}}(r_i)$.
3) Let $C_1, C_2, \ldots, C_m$ represent m classes into which the n elements are divided.
4) Let $S_{C_j}(r_n)$ denote a set of elements of class j on the universal set X. It is a subset of the universal set X.
5) The match degree D_j with the fuzzy set $\tilde{A}$ for the elements of class j in an interval, where $j = 1, 2, \ldots, m$, is defined as
$$D_j = \frac{\displaystyle\sum_{r \in S_{C_j}(r_n)} \mu_{\tilde{A}}(r)}{\displaystyle\sum_{r \in X} \mu_{\tilde{A}}(r)}. \tag{5}$$
6) The fuzzy entropy $FE_{C_j}(\tilde{A})$ of the elements of class j in an interval is defined as
$$FE_{C_j}(\tilde{A}) = -D_j \log_2 D_j. \tag{6}$$
7) The fuzzy entropy $FE(\tilde{A})$ on the universal set X for the elements within an interval is defined as
$$FE(\tilde{A}) = \sum_{j=1}^{m} FE_{C_j}(\tilde{A}). \tag{7}$$

In (6), the fuzzy entropy $FE_{C_j}(\tilde{A})$ is a nonprobabilistic entropy. Therefore, we coin the new term "match degree" for D_j. The basic property of proposed fuzzy entropy is similar to that of Shannon's entropy and it satisfies these four Luca–Termini axioms; however, their ways of measuring information are not the same. The probability p_j of Shannon's entropy is measured via the number of occurring elements. In contrast, the match degree D_j in fuzzy entropy is measured via the membership values of occurring elements. Furthermore, the fuzzy entropy of decision regions can be obtained via summation of the fuzzy entropy of individual intervals in each feature dimension. Finally, we illustrate the difference between Shannon's entropy and proposed fuzzy entropy by measuring the information of two distributions as shown in Fig. 1.

Since the probabilities of these two distributions within any interval of the pattern space [0,1] in Fig. 1(a) and (b) are the same, both of these two distributions have the same Shannon's entropy. For instance, the computation of Shannon's entropy in the interval $[i_1, i_2]$ is demonstrated as follows.

The probability of "$\star$" is: $p_{star} = 5/6$, the probability of "◯" is $1 - p_{star} = 1/6$.

Shannon's entropy is

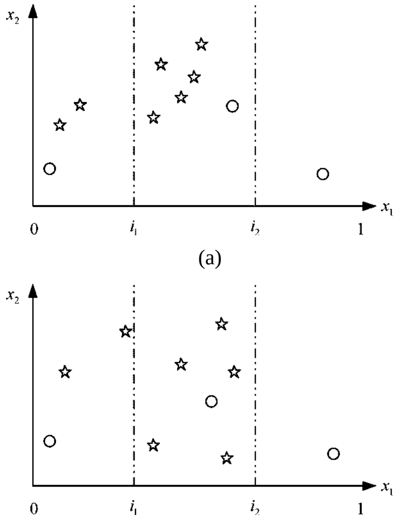
$$\begin{aligned} H(p) &= -p_{star} \log_2 p_{star} - (1 - p_{star}) \log_2 (1 - p_{star}) \\ &= -\tfrac{5}{6} \log_2 \tfrac{5}{6} - \tfrac{1}{6} \log_2 \tfrac{1}{6} \\ &= 0.650\,024. \end{aligned}$$

As mentioned above, the match degree D_j in fuzzy entropy is based on mapped membership values of elements. Assume we begin by assigning three triangular membership functions with overlapped regions in the pattern space of [0,1], as shown in Fig. 2, for the pattern space shown in Fig. 1. The value of a membership function can be viewed as the degree to which a pattern belongs to a specified pattern space.

The fuzzy entropy of the interval $[i_1, i_2]$ for the two distributions shown in Fig. 2 are as follows:

Distribution (a):

1) From the corresponding membership function $\tilde{A}$, the total membership degree of "$\star$" is $0.35 + 0.3 + 0.1 + 0.0 + 0.0 = 0.75$.
   Total membership degree of "◯" is 0.0.



(a)

(b)

Fig. 1. Two examples of pattern distribution with two features and classes. (⋆ and ◯ denote the patterns of class 1 and class 2, respectively).

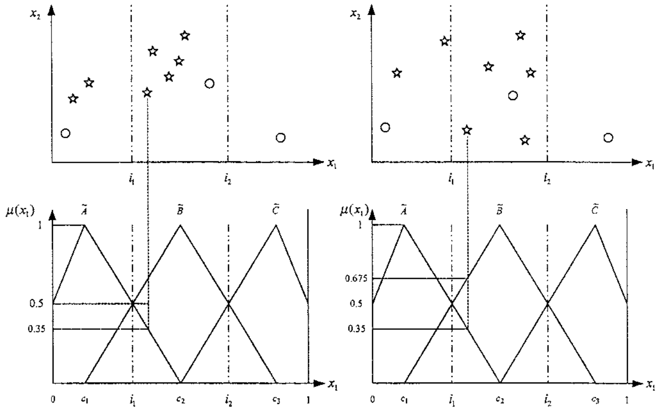


Fig. 2. Two examples of pattern distribution with corresponding membership functions. (c_1–c_3 denote the centers of three triangular fuzzy sets, respectively).

The match degree of "⋆" is $D_1 = 0.75/(0.75 + 0.0) = 1.0$.
The match degree of "◯" is $D_2 = 0.0/(0.75 + 0.0) = 0.0$.
The fuzzy entropy of $FE_{C_j}(\tilde{A})$ is

$$FE_{C_1}(\tilde{A}) = -1.0 \times \log_2(1.0) = 0.0$$
$$FE_{C_2}(\tilde{A}) = -0.0 \times \log_2(0.0) = 0.0.$$

Hence, the fuzzy entropy of $FE(\tilde{A})$ for the patterns of the interval $[i_1, i_2]$ in the feature dimension x_1 is

$$FE(\tilde{A}) = FE_{C_1}(\tilde{A}) + FE_{C_2}(\tilde{A}) = 0.0 + 0.0 = 0.0.$$

2) From the corresponding membership function $\tilde{B}$, the total membership degree of "⋆" is $0.675+0.75+0.9+1.0+0.95 = 4.275$.
Total membership degree of "◯" is 0.7.
The match degree of "⋆" is $D_1 = 4.275/(4.275 + 0.7) = 0.86$.
The match degree of "◯" is $D_2 = 0.7/(4.275+0.7) = 0.14$.
The fuzzy entropy of $FE_{C_j}(\tilde{B})$ is

$$FE_{C_1}(\tilde{B}) = -0.86 \times \log_2(0.86) = 0.187\,13$$
$$FE_{C_2}(\tilde{B}) = -0.14 \times \log_2(0.14) = 0.397\,11.$$

Hence, the fuzzy entropy of $FE(\tilde{B})$ for the patterns of the interval $[i_1, i_2]$ in the feature dimension x_1 is

$$\begin{aligned} FE(\tilde{B}) &= FE_{C_1}(\tilde{B}) + FE_{C_2}(\tilde{B}) \\ &= 0.187\,13 + 0.397\,11 \\ &= 0.584\,24. \end{aligned}$$

3) From the corresponding membership function $\tilde{C}$, the total membership degree of "⋆" is $0.0 + 0.0 + 0.0 + 0.0 + 0.05 = 0.05$.
Total membership degree of "◯" is 0.3.
The match degree of "⋆" is $D_1 = 0.05/(0.05 + 0.3) = 0.14$.
The match degree of "◯" is $D_2 = 0.3/(0.05 + 0.3) = 0.86$.
The fuzzy entropy of $FE_{C_j}(\tilde{C})$ is

$$FE_{C_1}(\tilde{C}) = -0.14 \times \log_2(0.14) = 0.397\,11$$
$$FE_{C_2}(\tilde{C}) = -0.86 \times \log_2(0.86) = 0.187\,13.$$

Hence, the fuzzy entropy of $FE(\tilde{C})$ for patterns of the interval $[i_1, i_2]$ in the feature dimension x_1 is

$$\begin{aligned} FE(\tilde{C}) &= FE_{C_1}(\tilde{C}) + FE_{C_2}(\tilde{C}) \\ &= 0.397\,11 + 0.187\,13 \\ &= 0.584\,24. \end{aligned}$$

Finally, we can obtain the whole fuzzy entropy via summation of all corresponding fuzzy entropies as follows:

$$\begin{aligned} FE &= FE(\tilde{A}) + FE(\tilde{B}) + FE(\tilde{C}) \\ &= 0.0 + 0.584\,24 + 0.584\,24 \\ &= 1.168\,48. \end{aligned}$$

Distribution (b): Using the same computational method as distribution (a), we can get the whole fuzzy entropy of distribution (b) in the interval $[i_1, i_2]$ as follows:

$$\begin{aligned} FE &= FE(\tilde{A}) + FE(\tilde{B}) + FE(\tilde{C}) \\ &= 0.0 + 0.701\,48 + 0.634\,31 \\ &= 1.335\,79. \end{aligned}$$

As shown above, the fuzzy entropy of distribution (a) is lower than that of distribution (b). This result means that the "⋆" distribution in (a) has more order than that in (b). Looking at the actual distributions, this result makes sense. In contrast, using Shannon's entropy, the values for these two distributions are identical in the interval $[i_1, i_2]$.

From the above illustration, we can see that proposed the fuzzy entropy is able to discriminate the actual distribution of patterns better. By employing membership functions for measuring match degrees, the value of entropy not only considers the number of patterns but also takes the actual distribution of patterns into account.

III. Operations of the Fuzzy Classifier FEBFC

For a classification system, the most important procedure is partitioning the pattern space into decision regions. Once the decision regions are decided, we can apply these partitioned decision regions to classify the unknown patterns. The partition of decision regions is part of the learning procedure or training procedure since the decision regions are decided by the training patterns.

In the proposed fuzzy entropy-based fuzzy classifier (FEBFC), decision regions are enclosed by the surfaces produced from each dimension. The surfaces are determined by the distribution of input patterns. For a two-class, two-dimension (2-D) pattern recognition example, the decision regions are the partitioned fuzzy subspaces as shown in Fig. 3.

These subspaces divide the pattern space into a light gray area for “○” class and a dark gray area for “$\star$” class, respectively. The surfaces of each subspace are parallel to each dimension, i.e., the surfaces are extended from the boundaries of membership functions (named intervals) on each dimension.

To produce the intervals on each dimension, or equivalently, to generate several triangular membership functions for each real-value attribute (a process also referred as the discretization of attributes [33]–[35]), some points need to be considered. First, the number of intervals on each dimension needs to be determined. Second, the center and the width of each interval have to be computed. For the first consideration, we employ the fuzzy entropy to determine the appropriate number of intervals. For the second consideration, we employ the K-means clustering algorithm [36], [37] to find the interval centers. After the interval centers are determined, it is easy to decide on the width of each interval.

From the above description, we can summarize the proposed FEBFC with the following four steps:

1) Determination of the number of intervals on each dimension.
2) Determination of the interval locations, i.e., determining the center and width of each interval.
3) Membership function assignment for each interval.
4) Class label assignment for each decision region.

A. Determination of the Number of Intervals

As pointed out in [33] and [38], the number of intervals on each dimension has a profound effect on learning efficiency and classification accuracy. If the number of intervals is too large, i.e., the partition is too fine, it will take too long to finish the training and classification process, and overfitting may result. On the other hand, if the number of intervals is too small, the size of each decision region may be too big to fit the distribution of input patterns, and classification performance may suffer.

However, the selection of the optimal number of intervals is seldom addressed in the existing literature. In most cases, it is determined arbitrarily or heuristically [33], [36], [38]. In this subsection, we will investigate a systematic method to select the appropriate number of intervals. The proposed criterion is based on fuzzy entropy measure as described in Section II-C, since it has the ability to reflect the actual distribution of pattern space. The steps involved in selecting the interval number for each dimension is described as follows:

Step 1) Set the initial number of intervals $I = 2$.

Step 2) Locate the centers of intervals.

A clustering algorithm will be used to locate the center of each interval for determining the interval boundaries and the width of intervals. The details of this step will be discussed in Section III-B.

Step 3) Assign membership function for each interval.

In order to apply the fuzzy entropy to measure the distribution information of patterns in an interval, we have to assign a membership function to each interval. The details will be described in Section III-C.

Step 4) Compute the total fuzzy entropy of all intervals for **I** and $\mathbf{I-1}$ intervals.

We compute the fuzzy entropy of all intervals on each dimension to obtain the distribution information of patterns projecting on this dimension. The fuzzy entropy function used is that described in Section II-C.

Step 5) Does the total fuzzy entropy decrease?

If the total fuzzy entropy of **I** intervals is less than that of $\mathbf{I-1}$ intervals, (i.e., partitioning this dimension into **I** intervals will be more “order” than into $\mathbf{I-1}$ intervals), then partition again ($\mathbf{I := I+1}$) and go to **Step 2**; else go to **Step 6.**

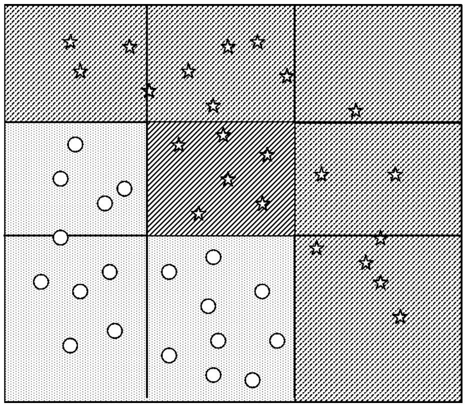


Fig. 3. Example with 2-D decision regions with two-classes. The light gray area represents “○” class and the dark gray area indicates “$star$” class, respectively.

Step 6) $\mathbf{I-1}$ is the number of intervals on specified dimension.

Since the fuzzy entropy does not decrease, we stop further partitioning on this dimension and $I-1$ is the number of intervals on this dimension.

B. Determination of the Interval Locations

The process of determining the interval locations begins with finding the center points of intervals. Once the center of an interval is determined, the width and boundaries of an interval can be easily decided. The method of determining the width and boundaries of an interval is described in Section III-C. To find the centers, the *K*-means clustering algorithm [36], [37] is applied. This is a useful, unsupervised learning method and it is based on the Euclidean distance measure.

Suppose there are N M-dimensional vectors $V_i = (v_{i1}, v_{i2}, \ldots, v_{iM})^T$, $i = 1, 2, \ldots, N$, corresponding to N elements. For partitioning the elements into several intervals on dimension j, we first extract N values from elements projected on this dimension $x_i^{(j)} = v_{ij}$, $i = 1, 2, \ldots, N$. The *K*-means clustering algorithm is then used to perform clustering on $x_i^{(j)}$, $i = 1, 2, \ldots, N$. The algorithm consists of the following steps.

Step 1) Set the initial number of clusters, **I**.

This is the procedure for determining the number I of intervals described in Section III-A.

Step 2) Set initial centers of clusters.

The initial cluster centers $c_1, c_2, \ldots, c_I$ can be randomly selected from $x_i^{(j)}$, $i = 1, 2, \ldots, N$. In the proposed system, the cluster center c_q of an arbitrary cluster q is assigned as follows:

$$c_q = \frac{q-1}{I-1}, \qquad q = 1, 2, \ldots, I. \tag{8}$$

Step 3) Assign cluster label to each element.

After determining the cluster centers, we assign each element a cluster label according to which cluster center is the “closest.” That is the center with the smallest Euclidean distance from the element. Thus, the closest center satisfies the following distance measure:

$$\left| x_i^{(j)} - c_q^* \right| = \min_{1 \le q \le I} \left| x_i^{(j)} - c_q \right| \tag{9}$$

where c_q^* is the closest center to the element $x_i^{(j)}$, i.e., among $c_1, c_2, \ldots, c_I$, c_q^* has the minimum Euclidean distance to $x_i^{(j)}$.

Step 4) Recompute the cluster centers.

Since the initial centers are selected randomly, we have to re-evaluate each center by computing the following estimate:

$$c_q = \frac{\sum_{q=1}^{N_q} x_i^{(j)}}{N_q} \tag{10}$$

where N_q is the total number of patterns within the same cluster q.

Step 5) Does any center change?

If each cluster center is determined appropriately, the recomputed center in Step 4 would not change. If so, stop the determination of interval centers, otherwise go to Step 3.

C. Membership Function Assignment

Membership function assignment is a procedure for assigning a membership function to each interval. In order to apply the fuzzy entropy to evaluate the information of pattern distribution in an interval, we have to assign a corresponding membership function to each interval to indicate the membership degrees of the elements. The membership value of an element within an interval can be viewed as the degree of this element belonging to this interval. Intuitively, the center of an interval has the highest membership value, and the membership value of an element decreases as the distance between this element and the corresponding interval center increases. Hence, we assign the highest membership value "1.0" to the center of an interval, and the lowest value "0.0" to the centers of this interval's neighbors. In this paper, we use triangular fuzzy sets to implement this system. Fig. 4 shows an example of the intervals with corresponding membership functions on one dimension. In this figure, assume c_1, c_2, c_3, and c_4 are the interval centers. Note that the values of all the elements have been normalized to fit the interval [0,1] for simplicity.

When assigning a membership function to an interval, we have to consider the following three cases:

Case I) The left-most interval.

In this case, as shown in Fig. 4, the first interval center c_1 on this dimension is bounded by only one interval center c_2. The highest membership value "1.0" of this interval is located at c_1, and the lowest membership value "0.0" is located at c_2. When $x = 0$, the membership value is set to be 0.5, as shown in Fig. 4. That is, the membership function μ_{i1} of the left-most interval on dimension i is defined as follows:

$$\mu_{i1}(x) = \begin{cases} \dfrac{c_1 + x}{2c_1}, & \text{for } x \le c_1 \\ \max\left\{1 - \dfrac{|x - c_1|}{|c_2 - c_1|}, 0\right\}, & \text{for } x > c_1 \end{cases} \tag{9}$$

where c_1 is the center of the left-most interval, and c_2 is the center of the first interval center to the right of c_1.

Case II) The right-most interval.

In this case, as shown in Fig. 4, the membership function μ_{i4} of the right-most interval on dimension i is defined by

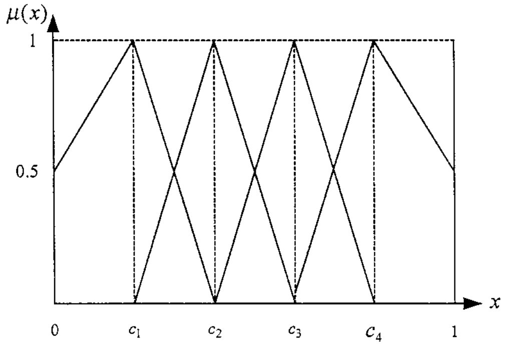


Fig. 4. Example of membership function assignment. c_1, c_2, c_3, c_4 are the interval centers and the triangles are the corresponding membership functions.

$$\mu_{i4}(x) = \begin{cases} \max\left\{1 - \dfrac{|c_4 - x|}{|c_4 - c_3|}, 0\right\}, & \text{for } x \le c_4 \\ \dfrac{2 - x - c_4}{2(1 - c_4)}, & \text{for } x > c_4 \end{cases} \tag{10}$$

where c_4 is the center of the right-most interval, and c_3 is the center of the first interval to the left of c_4.

Case III) The internal intervals.

In this case, as shown in Fig. 4, an internal interval center c_3 is bounded by its left interval center c_2 and right interval center c_4. The highest membership value is located at c_3, and the lowest values is at c_2 and c_4. The membership function μ_{i3} of the third interval on dimension i in this case is defined as

$$\mu_{i3}(x) = \begin{cases} \max\left\{1 - \dfrac{|c_3 - x|}{|c_3 - c_2|}, 0\right\}, & \text{for } x \le c_3 \\ \max\left\{1 - \dfrac{|c_3 - x|}{|c_4 - c_3|}, 0\right\}, & \text{for } x > c_3. \end{cases} \tag{11}$$

D. Class Label Assignment

In order to assign a class label to each decision region, we must use the similar evaluation method of fuzzy entropy described in Section II-C to set the class of each decision region. Differing from the previous example of measuring fuzzy entropy of an interval, the fuzzy entropy of decision regions for the patterns of each class is computed to determine the class of each decision region. In fact, the fuzzy entropy of decision regions can be obtained via summation of the fuzzy entropy of individual intervals in each feature dimension. We assign the decision region to the class with the lowest fuzzy entropy in this region. Once each decision region is given a class label, the training process is completed.

IV. Feature Selection

In this section, we give a new feature selection approach based on our proposed fuzzy entropy measure. As illustrated in Section II, the fuzzy entropy reflects more information in the actual distribution of patterns in the pattern space. Since the fuzzy entropy is able to discriminate pattern distribution better, we employ it to evaluate the separability of each feature. Intuitively, the lower the fuzzy entropy of a feature is, the higher the feature's discriminating ability is. The procedure for computing the fuzzy entropy of each feature is described as follows:

Step 1) Determine the number of intervals.

See Section III-A.

Step 2) Determine the locations of intervals.

See Section III-B.

Step 3) Assign a membership function for each interval.

See Section III-C.

Step 4) Compute the fuzzy entropy of each feature via summation of the fuzzy entropy of all intervals in this feature dimension.

See the example of Section II-C.

After the fuzzy entropy of each feature has been determined, we can select the features by *forward selection* or *backward elimination*. The forward selection method is to select the relevant features beginning with an empty set and iteratively add features until the termination criterion is met. In contrast, the backward elimination method starts with the full feature set and removes features until the termination criterion is met. In our feature selection process, we use the backward elimination to select the relevant features and the termination criterion in our method is based on the classification rate of the classifier. Since features with higher fuzzy entropy are less relevant to our classification goal, we remove the feature which has the highest fuzzy entropy if doing so does not decrease the classification rate. Then we repeat the above step until all "irrelevant" features are removed. Finally, the left features are served as features for classification.

With this feature selection procedure, we can reduce the dimensionality of the problem to speed up the classification process. In some cases, we can also achieve better classification results by discarding redundant, noise-corrupted, or unimportant features.

V. Experiments

In this section, we evaluate the performance of the proposed FEBFC by applying it to the Iris data set [39] and Wisconsin Breast Cancer data set [40]. These two data sets are well-known benchmark data for evaluating the performance of classifiers. We also compare the performance of the FEBFC using these databases with the performance of other classifiers.

A. Iris Database

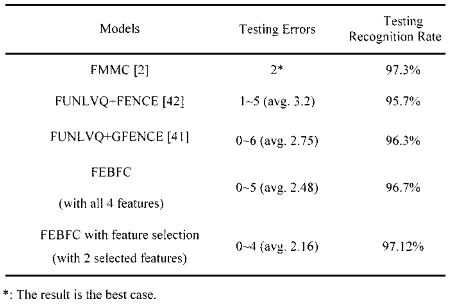
The Iris database [39] created by Fisher includes three classes: Virginica, Setosa, and Versicolor. This database has four continuous features (sepal length, sepal width, petal length, and petal width) and consists of 150 instances: 50 for each class. The Setosa class is linearly separable from the other two, but the other two overlap each other. One half of the 150 instances are randomly selected as the training set, and the remaining patterns are used as the testing set.

Table I shows results using the FMMC [2] classifier, our previous classifier [41], [42], and the FEBFC. The experimental results of the proposed FEBFC are evaluated under two different circumstances: without feature selection (using the four original features) and with feature selection (using two selected features). The testing recognition rate of our proposed FEBFC without feature selection is 96.7%. After the proposed feature selection approach is included, two redundant features are removed from four candidate features. The experimental results show that the testing recognition rate is effectively accelerated from 96.7% to 97.12%.

B. Wisconsin Breast Cancer Diagnostic Database

The Wisconsin Breast Cancer Diagnostic Database (WBCDD) [40] was obtained form the University of Wisconsin Hospitals, Madison from Dr. W. H. Wolberg. This data set contains 699 patterns in two classes; 458 patterns belong to the "benign" class, the other 241 patterns are the "malignant" class. Each pattern is described by nine features. Since 16 of the data set have missing values, we use 683 patterns to evaluate the proposed classifier. Half of the 683 patterns are used as the training set, and the remaining patterns are used as the testing set.

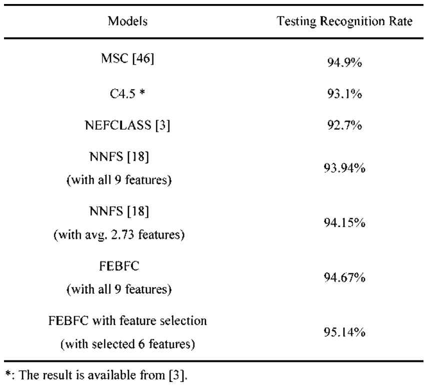
TABLE I
CLASSIFICATION RESULTS ON IRIS



Models	Testing Errors	Testing Recognition Rate
FMMC [2]	2*	97.3%
FUNLVQ+FENCE [42]	1~5 (avg. 3.2)	95.7%
FUNLVQ+GFENCE [41]	0~6 (avg. 2.75)	96.3%
FEBFC (with all 4 features)	0~5 (avg. 2.48)	96.7%
FEBFC with feature selection (with 2 selected features)	0~4 (avg. 2.16)	97.12%

*: The result is the best case.

TABLE II
CLASSIFICATION RESULTS ON WISCONSIN BREAST CANCER DATABASE



Models	Testing Recognition Rate
MSC [46]	94.9%
C4.5 *	93.1%
NEFCLASS [3]	92.7%
NNFS [18] (with all 9 features)	93.94%
NNFS [18] (with avg. 2.73 features)	94.15%
FEBFC (with all 9 features)	94.67%
FEBFC with feature selection (with selected 6 features)	95.14%

*: The result is available from [3].

Table II shows results from the various traditional, fuzzy, and neural network classifiers. Besides the FEBFC, only the NNFS has the ability to perform the feature selection. The experimental results of the NNFS and the proposed classifier FEBFC are evaluated under two different circumstances: without feature selection and with feature selection. The proposed feature selection approach selects six features whereas the NNFS selects 2.73 features on average. Although the number of features selected by the NNFS is less, the classification result shows that the proposed classifier FEBFC performs better.

As the results show, feature selection in the Iris and Wisconsin Breast Databases not only reduces the dimensions of problem, but also improves classification performance by discarding redundant, noise-corrupted, or unimportant features.

VI. Discussion and Conclusion

The goal of traditional pattern classification is to partition the pattern space into decision regions, one region for each class [37]. In many classification systems, the decision regions are partitioned into overlapping areas. Although a classifier with overlapping decision regions may achieve better classification performance, it takes much time to enclose the decision regions. Many approaches [1]–[3], [41]–[45] have been proposed to solve this problem.

In this paper, we presented an efficient classifier with feature selection based on fuzzy entropy for pattern classification. The pattern space is partitioned into nonoverlapping fuzzy decision regions. Since the decision regions are fuzzy subspaces, we can obtain smooth boundaries to achieve better classification performance even though the decision regions are nonoverlapping. Furthermore, since the decision regions do not overlap, the computational load and complexity of the classifier are both reduced.

Also we use the fuzzy entropy to select the relevant features. Applying feature selection not only reduces the dimensions of the problem, but also improves classification performance by discarding redundant, noise- corrupted, or unimportant features. Also, the K-means clustering algorithm was used to determine the membership functions of each feature. The combination of the approaches mentioned above yield an efficient fuzzy classifier.

References

[1] S. Abe and R. Thawonmas, "A fuzzy classifier with ellipsoidal regions," *IEEE Trans. Fuzzy Syst.*, vol. 5, pp. 358–367, June 1997.

[2] P. K. Simpson, "Fuzzy Min–Max neural networks—Part 1: Classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 776–786, Oct. 1992.

[3] D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data," *Fuzzy Sets Syst.*, vol. 89, no. 3, pp. 277–288, Aug. 1997.

[4] H. Ishibuchi *et al.*, "Construction of fuzzy classification systems with rectangular fuzzy rules using genetic algorithms," *Fuzzy Sets Syst.*, vol. 65, no. 2, pp. 237–253, Aug. 1994.

[5] F. Uebele *et al.*, "A neural network-based fuzzy classifier," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, pp. 353–361, Feb. 1995.

[6] K. Nozaki *et al.*, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," *Fuzzy Sets Syst.*, vol. 86, no. 3, pp. 251–270, Mar. 1997.

[7] D. P. Mandal, "Partitioning of feature space for pattern classification," *Pattern Recognit.*, vol. 30, no. 12, pp. 1971–1990, Dec. 1997.

[8] H. Genther and M. Glesner, "Advanced data preprocessing using fuzzy clustering techniques," *Fuzzy Sets Syst.*, vol. 85, no. 2, pp. 155–164, Jan. 1997.

[9] S. Abe, "Feature selection by analyzing class regions approximated by ellipsoids," *IEEE Trans. Syst., Man, Cybern. C*, vol. 28, pp. 282–287, May 1998.

[10] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artif. Intell.*, vol. 97, no. 2, pp. 245–271, Dec. 1997.

[11] R. Greiner *et al.*, "Knowing what doesn't matter: Exploiting the omission of irrelevant data," *Artif. Intell.*, vol. 97, no. 2, pp. 345–380, Dec. 1997.

[12] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997.

[13] R. Thawonmas and S. Abe, "A novel approach to feature selection based on analysis of class regions," *IEEE Trans. Syst., Man, Cybern. B.*, vol. 29, pp. 196–207, Apr. 1997.

[14] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 2, pp. 273–324, Dec. 1997.

[15] M. Dash *et al.*, "Dimensionality reduction of unsupervised data," in *Proc. IEEE Int. Conf. Tools Artificial Intell.*, Los Alamitos, CA, 1997, pp. 532–539.

[16] L. M. Belue and K. W. Bauer, "Determining input features for multilayer perceptrons," *Neurocomputing*, vol. 7, no. 2, pp. 111–121, Mar. 1995.

[17] J. M. Steppe, Jr. *et al.*, "Integrated feature and architecture selection," *IEEE Trans. Neural Networks*, vol. 7, pp. 1007–1014, Aug. 1996.

[18] R. Setiono *et al.*, "Neural-network feature selector," *IEEE Trans. Neural Networks*, vol. 8, pp. 654–662, June 1997.

[19] R. K. De *et al.*, "Feature analysis: Neural network and fuzzy set theoretic approaches," *Pattern Recognit.*, vol. 30, no. 10, pp. 1579–1590, Oct. 1997.

[20] R. Setiono, "A penalty-function approach for pruning feedforward neural networks," *Neural Computat.*, vol. 9, no. 1, pp. 185–204, Jan. 1997.

[21] G. S.-K. Fung *et al.*, "Fuzzy genetic algorithm approach to feature selection problem," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, vol. 1, 1997, pp. 441–445.

[22] L. Y. Tseng and S. B. Yang, "Genetic algorithms for clustering, feature selection and classification," in *Proc. IEEE Int. Conf. Neural Networks*, vol. 3, New York, 1997, pp. 1612–1615.

[23] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Intell. Syst.*, vol. 13, pp. 44–49, Mar./Apr. 1998.

[24] H. Vafaie and K. De Jong, "Feature space transformation using genetic algorithm," *IEEE Intell. Syst.*, vol. 13, pp. 57–65, Mar./Apr. 1998.

[25] B. Kosko, "Fuzzy entropy and conditioning," *Inform. Sci.*, vol. 40, pp. 165–174, Dec. 1986.

[26] S. K. Pal and B. Chakraborty, "Fuzzy set theoretic measure for automatic feature evaluation," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, pp. 754–760, May 1986.

[27] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.

[28] R. V. Hartley, "Transmission of information," *Bell Syst. Tech. J.*, vol. 7, pp. 535–563, 1928.

[29] N. Wiener, *Cybernetics*. New York: Wiley, 1961.

[30] A. Renyi, "On the measure of entropy and information," in *Proc. Fourth Berkeley Symp. Math. Statistics Probability*, vol. 1, Berkeley, CA, 1961, pp. 541–561.

[31] R. E. Belahut, *Principles and Practice of Information Theory*. Reading, MA: Addison-Wesley, 1987.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1992.

[33] J. Y. Ching *et al.*, "Class-dependent discretization for inductive learning form continuous and mixed-mode data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 641–651, July 1995.

[34] H. Liu and R. Setiono, "Feature selection via discretization," *IEEE Trans. Knowl. Data Eng.*, vol. 9, pp. 642–645, July/Aug., 1997.

[35] B. H. Jun *et al.*, "A new criterion in selection and discretization of attributes for the generation of decision trees," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 1371–1375, Dec. 1997.

[36] Z. Chi and H. Yan, "Feature evaluation and selection based on an entropy measure with data clustering," *Opt. Eng.*, vol. 34, pp. 3514–3519, Dec. 1995.

[37] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[38] K. Nozaki *et al.*, "Adaptive fuzzy rule-based classification systems," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 238–250, June 1996.

[39] R. A. Fisher, "The use of multiple measurements in taxonomic problem," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[40] O. L. Mangasarian and W. H. Wolberg, "Cancer diagnosis via linear programming," *SIAM News*, vol. 23, no. 5, pp. 1–18, 1990.

[41] H. M. Lee, "A neural network classifier with disjunctive fuzzy information," *Neural Networks*, vol. 11, no. 6, pp. 1113–1125, Aug. 1998.

[42] K. H. Chen *et al.*, "A multiclass neural network classifier with fuzzy teaching inputs," *Fuzzy Sets Syst.*, vol. 91, no. 1, pp. 15–35, Oct. 1997.

[43] D. Wettschereck and T. G. Dietterich, "An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms," *Mach. Learn.*, vol. 19, no. 1, pp. 5–27, Apr. 1995.

[44] S. Salzberg, "A nearest hyperrectangle learning method," *Mach. Learn.*, vol. 6, no. 3, pp. 251–276, May 1991.

[45] M. A. Abounasr and M. A. Sidahmed, "Fast learning and efficient memory utilization with a prototype based neural classifier," *Pattern Recognit.*, vol. 28, no. 4, pp. 581–593, Apr. 1995.

[46] B. C. Lovel and A. P. Bradley, "The multiscale classifier," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 124–137, Feb. 1996.

View publication stats