Identification of a Gaussian Fuzzy Classifier

Heesoo Hwang

Abstract: This paper proposes an approach to deriving a fuzzy classifier based on evolutionary supervised clustering, which identifies the optimal clusters necessary to classify classes. The clusters are formed by multi-dimensional weighted Euclidean distance, which allows clusters of varying shapes and sizes. A cluster induces a Gaussian fuzzy antecedent set with unique variance in each dimension, which reflects the tightness of the cluster. The fuzzy classifier is composed of as many classification rules as classes. The clusters identified for each class constitute fuzzy sets, which are joined by an "and" connective in the antecedent part of the corresponding rule. The approach is evaluated using six data sets. The comparative results with different classifiers are given.

Keywords: Fuzzy classifier, pattern classification, clustering, differential evolution.

1. INTRODUCTION

Fuzzy logic improves classification and decision support systems by allowing the use of overlapping class definitions and improves the interpretability of the results by providing more insight into the classifier structure and decision process [9]. Several different techniques, such as neuro-fuzzy methods [5], genetic algorithm based rule selection [6], and clustering with evolutionary optimization [6, 7, 13] have been produced for the automatic determination of fuzzy classification rules from data. Most of the last approaches are devoted to improving the results of the fuzzy c-means (FCM) algorithm by using genetic algorithms (GAs) to optimize some parameters of the algorithm. The use of GAs generates three different groups of fuzzy clustering algorithms. Prototype-based algorithms encode the fuzzy cluster prototypes and evolve them by means of GAs guided by any centroid objective function [12, 16]. Fuzzy partition-based algorithms encode and evolve the fuzzy membership matrix [18]. The second group uses GAs to define the distance norm of FCM algorithms. The system considers the adaptive distance function and employs GAs to learn its parameters and obtain an optimal behaviour of the FCM algorithm [4]. Finally, the third group is based on directly solving the clustering problem without interaction with any FCM algorithm [3]. This group is of our concern. In any group, the quality of data clustering and the accuracy of classification are influenced by three parameters: the number of clusters for each class, their positions, and their shapes. Most clustering algorithms require a priori knowledge of the problem to fix the number and starting positions of the clusters. Although such knowledge may be assumed for domains whose dimensionality is fairly small or whose underlying structure is intuitive, it is clearly much less accessible in a hyper-dimensional system.

To tackle this problem, evolutionary supervised clustering (ESC) is proposed. ESC produces multi-dimensional weighted Euclidean distance-based clusters, which can easily be converted to Gaussian fuzzy sets. For each class, one optimal cluster with its centroid, radius, and weighting factors is searched by differential evolution (DE), and then feature vectors contained in the cluster are removed in the feature vector set. For the remaining feature vectors, this process is continued until a predefined maximum cluster number is reached. For each class, the clusters identified by ECS are converted to Gaussian fuzzy sets, which are joined by an "and" connective in the antecedent part of the classification rule. The fuzzy sets that have the form of unique variance in each dimension improve classification by allowing the use of overlapping class definitions. The fuzzy classifier is composed of as many classification rules as classes. The proposed approach is applied to six classification problems, and comparative results with other techniques are given.

2. FUZZY CLASSIFIER STRUCTURE

Let's consider fuzzy classification rules that each describe one of n_{pc} classes in the feature vector set. The antecedent of the fuzzy rule is a fuzzy description in the d-dimensional feature space and the consequent is a crisp class label from the set $\{1,2,\cdots,n_{pc}\}$. The degree of fulfillment of each rule relates to truth-

Manuscript received May 16, 2002; revised October 11, 2002; accepted December 7, 2002. Recommended by Editor Sung-Kwan Oh.

Hee-Soo Hwang is with the School of Electrical Engineering, Halla University, San 66, Heun-gup, Wonju, Kangwon, 220-712, Korea (e-mail: hshwang@halla.ac.kr).

value, i.e., the membership grade of a pattern to the rule's class. The rules have the form of (1).

$$R_i : \textit{If } \mathbf{x} \text{ is } A_{i1} \text{ and } \mathbf{x} \cdots \text{ is } A_{ij} \text{ and } \cdots \mathbf{x} \text{ is } A_{ij}, \quad \textit{then } class_i \tag{1}$$

where $x = \{x_1, x_2, \cdots x_d\}$ is a d-dimensional input feature vector. A fuzzy singleton output of the i th rule, $class_i$ is one of values $\{1, 2, \cdots n_{pc}\}$. The "and" connective is modeled by a max operator. i_{nc} is the number of antecedent fuzzy sets in the i th rule. A_{ij} denotes the j th Gaussiane fuzzy set in the i th rule and is defined by

$$A_{ij}(X) = \exp(-h_{ij}(X)), \tag{2}$$

$$h_{ij}(X) = \beta_{ij} \cdot d_{ij}(X), \tag{3}$$

$$d_{ij}(\mathbf{x}) = \sqrt{\alpha_{ij}^1 \cdot \left(x_1 - c_{ij}^1\right)^2 + \cdots + \alpha_{ij}^k \cdot \left(x_d - c_{ij}^k\right)^2 + \cdots + \alpha_{ij}^d \cdot \left(x_d - c_{ij}^d\right)^2}, \tag{4}$$

where $d_{ij}(x)$ is the weighted distance between $\mathbf{x}$ and c_{ij}. $\alpha_{ij}^k (\geq 0)$ is a weighting factor for the k-th input feature, $h_{ij}(x)$ is a tuned distance, and $\beta_{ij}(>0)$ is a width parameter for the membership function.

The degree of activation of the i th rule is calculated as in (5).

$$\omega_i(\mathrm{x}) = \Pi_{j=1}^{i_{nc}} \left|A_{ij}(x)\right|, \quad i = 1, 2, \cdots, n_r \tag{5}$$

where Π is a max operator. The classifier output is determined by the rule that has the highest degree of activation as in (6). A crisp decision is made by taking the class belonging to the fuzzy-rule with the maximum degree of activation.

$$y = class_i^*, \quad i^* = \underset{1 \leq i \leq n_r}{\arg\max\, \omega_i}. \tag{6}$$

In the following, we assume that the number of rules is equal to that of classes, i.e., $n_r = n_{pc}$. The certainty degree of the decision is given by the normalized degree of firing of the rule as in (7).

$$CF = \frac{\omega_i^*}{\sum_{i=1}^{n_r} \omega_i}. \tag{7}$$

3. IDENTIFYING FUZZY CLASSIFICATION RULES

3.1. Overview of evolutionary algorithm based hard clustering

Clustering is a traditional machine learning problem. The most popular hard clustering method is the well-known K-means algorithm. However, a number of good reasons exist for considering other clustering methods as well [11]. One alternative to the K-means clustering algorithm is to consider an evolutionary algorithm based clustering method, where the evolutionary algorithm determines the cluster centers to reduce the cluster dispersion measure or any other measure related to cluster performance. In the clustering a collection of feature cases is partitioned into classes by minimizing the objective function of (8).

$$J_c = \sum_{k=1}^{n_{pc}} \sum_{i=1}^{n} \delta_{ik} \cdot \|x_i - c_k\| \tag{8}$$

here J_c is a cluster dispersion measure to be minimized, n is the number of feature vectors, n_{pc} is the number of classes (groups) to be classified, δ_{ik} is 1 when case i belongs to cluster k, 0 otherwise, x_i is the feature vector for case i, and c_k is the vector for the k th cluster center.

Implementing an evolutionary algorithm for searching the cluster centers to minimize the objective function as in (8) is straightforward. Note that so far the number of clusters was predefined. Extending the evolutionary algorithm driven clustering to allow for a varying number of clusters [1, 11] is now possible. After starting out with a relatively large prescribed number of clusters, we can let the number of clusters vary by adding a penalty function to the cluster dispersion as in (9). This approach helps to reduce the amount of over-fitting. In this regard, the additive clustering approach, i.e., introducing a new cluster gradually, seems to be more favorable.

$$J = J_c + r \cdot J_e. \tag{9}$$

Here γ is a penalty factor and its value is problem dependent. J_e is the number of empty clusters. An empty cluster, i.e., a dummy cluster, has no data member.

3.2. Evolutionary supervised clustering

An evolving weighted Euclidean distance-based clustering method, called evolutionary supervised clustering (ESC) is proposed to induce Gaussian fuzzy membership functions, which constitute the antecedents of fuzzy classification rules. The basic concept of ESC is no more than a traditional evolutionary clustering approach, but its particular implementation details are different. In traditional additive clustering, the maximum distance between a feature vector and its corresponding cluster center, or misclassification rate, can be used as a criterion for introducing a new cluster. However, a threshold value for the criterion is problem-dependent and has an influence on clustering performance.

The important advantage of ESC is that no priori knowledge is required. ESC constructs clusters one by one for each pattern class. Before starting ESC, we have to set $n_{c_{\max}}$ and $n_{d_{\min}}$, which are the maximum number of clusters and the minimum number of feature vectors to be contained in one cluster, respectively. A practical upper bound of seven can be placed on $n_{c_{\max}}$ since empirical evidence suggests that most classification problems require fewer than seven clusters per class. The value of $n_{d_{\min}}$ is equal to 2. For each class, the algorithm searches one cluster in the feature vectors set. After the search, the feature vectors contained in the cluster are eliminated in the feature vectors set. If there exist more than $n_{d_{\min}}$ eature vectors to classify in the feature vectors set, the search process is continued. The same process is repeated for all the remaining classes, i.e., the process is iterated $n_{c_{\max}}$ times. Fig. 1 illustrates how ESC is processed, in which 13 feature vectors for class 1 are marked by circles and 14 feature vectors for class 2 are marked by triangles. Part (a) is the initial state. Part (b) shows the first identified cluster, C_{11} for class 1. After removing the feature vectors contained in, C_{11} the second cluster, C_{12}, is identified as shown in Part (c). No feature vectors are available for further clustering for class 1, so from Part (a) the clustering is continued for class 2. Part (d) shows the first identified cluster, C_{21}, for class 2. After removing the feature vectors contained in, C_{21} the second cluster, C_{22}, is identified as shown in Part (e). The same procedure is applied for Part (f). The number of feature vectors contained in each cluster must be greater than $n_{d_{\min}}$. If the value of is set to 3, the clustering for C_{23} will not be performed.

A cluster is defined by three parameters, i.e., its center, radius, and weighting factors, as shown in (4). If the distance between a feature vector and a cluster center calculated by (4) is not greater than the radius of the cluster, the feature vector belongs to the cluster.

During the evolution, the weighting factors C_{12} for each cluster are searched in the range of real values. The negative weighting factors are considered to be zero, so the corresponding dimensions are neglected in the calculation of (4). If all the weighting factors of a

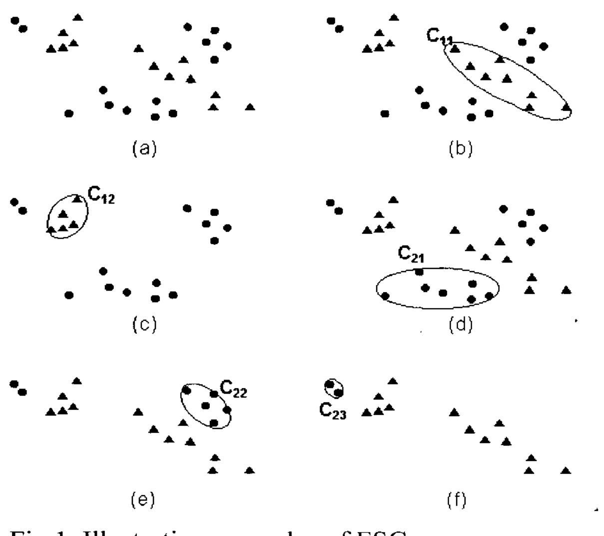

Fig.1. Illustrative examples of ESC.

cluster have the same values, the cluster has a hyper-spherical region; otherwise, it has an ellipsoidal region. The purpose of the weighting factors is to uncover the hidden structure of the cluster for classifying the feature vectors. In the case of a hyper-dimensional system whose underlying structure is unknown, the role of the weighting factors becomes of utmost importance.

During ESC, iterative clustering is achieved by not only maximizing classification defined by (10), but also by minimizing the cluster dispersion measure and the cluster radius defined by (11) and (12), respectively.

$$f_{classify} = \frac{1}{n} \cdot \sum_{i=1}^{n} B(x_i). \tag{10}$$

Here n is the number of feature vectors; $B(\mathbf{x}_i)$ is 1 if the i th feature vector $\mathbf{x}_i$ is successfully classified, otherwise 0.

$$f_{disperse} = \frac{1}{n_c} \sum_{i=1}^{n_c} \|x_i - c\|. \tag{11}$$

Here n_c is the number of feature vectors contained in the cluster $\mathbf{c}$. $\|\cdot\|$ is defined as in (4).

$$f_{radius} = r. \tag{12}$$

Here r is the radius of the cluster $\mathbf{c}$.

(10), (11), and (12) have different objectives, and they must be combined into one. (13) will achieve this combination.

$$f_{goal} = r_1 \cdot f_{classify} - r_2 \cdot f_{disperse} - r_3 \cdot f_{radius}. \tag{13}$$

Here f_{goal} is the final goal function to maximize and r_i is the i th weighting factor.

3.3. Differential evolution

As a search algorithm of ESC, real-valued differential evolution (DE) is used. DE has been proven to be a promising candidate for minimizing real-valued, multi-modal objective functions [14]. Besides its good convergence properties, DE is simple to understand and to implement. DE is a parallel direct search method that utilizes n_p d-dimensional parameter vectors as a population. The initial population is chosen randomly. DE generates a new vector by adding the weighted difference between the population vectors to a third vector, and then a crossover operation is performed. If the resulting vector yields a higher/lower objective function value in the maximization/minimization problem than a predetermined population member (old vector), the newly generated vector replaces the old vector otherwise the old vector is retained. Several variants of DE exist. The

DE/rand1 scheme is to be used in this paper. In this scheme, a new vector is generated as in (14).

$$v_{i,g+1} = p_{r_{1,g}} + F \cdot \left(p_{r_{2,g}} - p_{r_{3,g}}\right), \quad i = 1, 2, \cdots, n_p. \quad (14)$$

Here $v_{i,g+1}$ is the i th new vector, n_p is a population size, g is an iteration (generation) number, $p_{ri,g}$ is the r_i th individual (vector), $r_1, r_2, r_3 \in [1, n_p]$ are mutually different integers, and F is a positive real diversity factor.

To increase the potential diversity of the perturbed vectors of (14), crossover is introduced as in (15).

$$p_{ji,g+1} = \begin{bmatrix} v_{ji,g+1} & \text{if} & rand \geq c_r \\ p_{ji,g} & & otherwise \end{bmatrix} \quad (15)$$
$$for\ j = 1, 2, \cdots, d$$

Here $p_{ji,g+1}$ is the j th component of the i th individual, $v_{ji,g+1}$ is a new vector generated by (14), $rand \in [0 \quad 1]$ is a random number, c_r is a crossover rate, and d is a vector dimension.

The evolutionary search optimizes the clustering process by evolving three independent parameters: the position and the radius of the cluster and the weighting factors of the input variables. During evolution, these parameters are allowed to vary continually by (15). Each individual p_{r_i} $(i = 1, \cdots, n_p)$ for DE is constructed as shown in Fig. 2. If the dimension of the feature vector is, d the number of parameters constituting an individual is equal to $2 \cdot d + 1$. The effectiveness of an individual is measured by its fitness. Since the objective is classification, the measure of its fitness is the ability to correctly classify the feature vectors. (13) is used as a fitness function.

According to the above discussion, the summary of the ESC algorithm follows.

Step 1: Define the maximum number of clusters, $n_{c_{\max}}$, and the minimum number of feature vectors to be contained in one cluster, $n_{d_{\min}}$. Initialize the parameters of DE, such as population size n_p; maximum generation number; crossover rate, c_r; and diversity factor, F. The feature vectors are scaled to the range of [0,1] to assist the optimization routines. Scaling does not change the topology of the feature vector space, but giving each variable the same range allows similar search steps for each dimension.

Step 2: For a pattern class, DE searches an optimal cluster maximizing (13).

Step 3: When the maximum generation number is reached, DE terminates the process of searching the cluster. If the searched best cluster contains feature vectors less than $n_{d_{\min}}$ it is deleted; otherwise, the radius, the weighting factors, and the center points of the cluster are saved. The feature vectors contained in the cluster are removed in the feature vector set.

Step 4: If there exist feature vectors more than $n_{d_{\min}}$ to classify in the feature vector set, go to step 2.

Step 5: Iterate step 2 to step 4 for all the remaining classes.

3.4. Conversion of clusters to fuzzy membership functions

The clusters uncovered by ESC can be converted to Gaussian fuzzy membership functions, whose variance reflects the tightness of the cluster. The membership function of (2) requires the width parameter of (3) and the weighting factors and the center points of (4). Since the latter parameters are obtained from each cluster, the only width parameter whose role is to adjust the width of the membership function is unknown. The width is determined to give a degree of membership of 0.5 for the boundary of the cluster as illustrated in Fig. 3. The width parameter, β_{ij}, of the membership function $A_{ij}(\mathbf{x})$ for each cluster is calculated by (16).

$$\beta_{ij} = -\frac{\ln 0.5}{r_{ij}}. \quad (16)$$

Here r_{ij} is the radius of the j th cluster for the i th pattern class.

The determination of width parameters transform clusters into Gaussian membership functions, which constitute the antecedent fuzzy sets in the fuzzy classification rules of (1).

4. TEST EXAMPLES

Six test examples described in this section are from the UCI repository databases [19]. For brevity the fuzzy classifier constructed by ESC is hereinafter called ESCGFCS. The results obtained by the ESCGFCS are compared with those of other fuzzy classifiers. They are also compared with those of the commercial tool See5 [17], which is an extremely robust algorithm. The tests were run under the same initial setting for all parameters of the ESCGFCS. The initial settings are summarized in Table 1. All the random numbers are generated with uniform distribution.

The Fisher Iris data consist of 150 data with four input features and three classes [19]. The Wine data contains the chemical analysis of 178 wines produced in the same region in Italy but derived from three different cultivators [19]. The problem is to classify the three different types based on thirteen continuous attributes derived from the chemical analysis. To obtain an estimate for the classifier performance, we carried

radius	weighting factors					cluster center				
v	x^1	$\cdots$	x^1	$\cdots$	x^1	c^1	$\cdots$	c^1	$\cdots$	c^t

Fig. 2. Individual for iterative clustering in ESC.

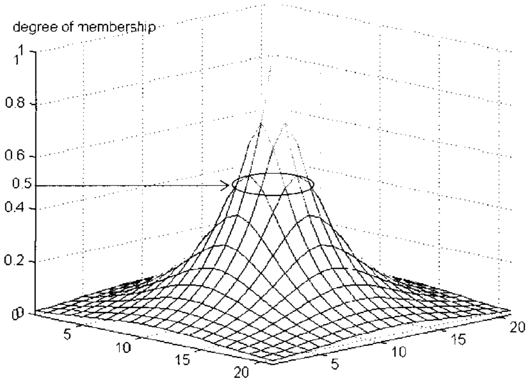

Fig. 3. Conversion of hard-cluster to Gaussian-membership function.

out ten independent runs and averaged the classification rates. Table 2 show performance comparison with other fuzzy classifiers on these data sets.

The classification rate gives the number of correctly classified data expressed as a percentage. In this case, all the data are used for the training, i.e., there is no separate test data set. It is done for the comparison with other methods. As an equivalent measure, the total number of membership functions used in the fuzzy classifiers is expressed with the number of rules. In case of the Iris data, the ESCGFCS with six membership functions and two cases of FMC with twelve membership functions were able to identify the four outliers, which limited the classification rate to 97.33%. In case of the Wine data, the ESCGFCS obtains the best accuracy with far fewer membership

Table 1. Initial parameter setting for ECS.

Parameter	Value
Population size (n_p)	30
Generations	300
Crossover rate (c_r)	0.5
Diversity factor (F)	0.5
Maximum number of clusters ($n_{c_{\max}}$)	7
Minimum number of data to be contained in cluster ($n_{d_{\min}}$)	2
Weighting factors (α_{ij}^k) of (4)	random number in the range of [-0.5,0.5]
Center points (c_{ij}^k) of (4)	random number in the range of [0,1]
Radii (r_{ij})	random number in the range of [ε,1]
Weighting factors of (13)	$r_1 = 1$, $r_2 = 0.01$, $r_2 = 0.01$

Table 2. Performance comparison (all data is used for the training).

Data	Algorithm	Parameters	Classification rate
Iris	ESC	6 clusters {1,3,2}	97.93
	ESCGFCS	6 membership functions, 3 rules	98.5
	Fuzzy Model Classifier [8]	12 membership functions	97.33, 98, 99.4
Wine	ESC	4 clusters {1,2,1}	98.7
	ESCGFCS	4 membership functions, 3 rules	99.4
	Fuzzy Classifier [7]	60 rules	98.5
	Fuzzy Classifier [9]	11 membership functions, 3 rules	98.3, 99.4

Table 3. Performance comparison (50% Jack-knife test).

Data	Algorithm	Parameters	Classification rate
Iris	ESC	4 clusters	91.33
	ESCGFCS	4 membership functions, 3 rules	96.7
	Trainable Fuzzy Classifier [12]	6 membership functions	96

functions. Fig. 4 shows the identified fuzzy classier for the Wine data, in which the hidden structure of the thirteen feature variables has been uncovered through the evolutionary search of the corresponding weighting factors.

Table 3 shows the performance comparison with other fuzzy classifiers on the Iris data. A 50% Jack-knife test was carried out. The ESCGFCS with fewer rules had better performance.

Table 4 shows the performance comparison with the commercial tool See5. To measure the performance, a 10-fold cross-validation was carried out. The number of rules for See5 are counted by summing the leaves on the tree or applying the expression $(s+1)/2$, where is the size of the tree. Even though the ESCGFCS needs much more computation time, it shows a better classification rate with fewer membership functions. For all cases, the ESCGFC is superior to ESC due to the overlapping class definitions by Gaussian fuzzy sets. As a result of ESC, Breast Cancer and Pima Indian have more than four clusters per class. For these two cases, another 10-fold cross-validation was carried out with the maximum number of clusters, $n_{c_{\max}}$ = 4. The result is shown in Table 5.

$\boldsymbol{R}_1$: **If x is** $A_{11}(x)$ then $class_2$.

$\boldsymbol{R}_2$: If **x** is $A_{21}(x)$ and **x** is $A_{22}(x)$ then $class_2$.

$\boldsymbol{R}_3$: If **x** is $A_{31}(x)$ (**x**) then. $class_2$

$$A_{11} = e^{-1.566\sqrt{0.379(x_1-0.839)^2+0.848(x_5-0.405)^2+0.505(x_7-0.618)^2+0.209(x_8-0.19)^2+0.68(x_{11}-0.521)^2+0.798(x_{12}-0.625)^2+0.571(x_{13}-0.677)^2}}$$

$$A_{21} = e^{-2.02\sqrt{0.156(x_7-0.467)^2+0.6(x_{10}-0.057)^2+0.045(x_{11}-0.885)^2+0.584(x_{13}-0.068)^2}}$$

$$A_{22} = e^{-6.652\sqrt{0.723(x_{11}-0.743)^2}}$$

$$A_{31} = e^{2.686\sqrt{0.103(x_6-0.276)^2+1.881(x_7-0.074)^2+0.2(x_{10}-0.598)^2+0.839(x_{11}\ 0.102)^2+0.6(x_{12}-0.328)^2}}$$

Fig. 4. The identified Gaussian fuzzy classification rules for the Wine data.

Table 4. Performance comparison (10-fold cross-validation).

Data	Algorithm	Parameters	Classification rate
Breast Cancer	ESC	11.4 clusters	93.6
	ESCGFCS	11.4 membership functions, 2 rules	96
	See5	14.4 rules	95.7
Iris	ESC	4.7 clusters	91.9
	ESCGFCS	4.7 membership functions, 3 rules	95.6
	See5	4.7 rules	95.3
Lenses	ESC	4 clusters	76.7
	ESCGFCS	4 membership functions, 3 rules	88.3
	See5	3.7 rules	84.5
Pima Indian	ESC	14 clusters	59
	ESCGFCS	14 membership functions, 2 rules	75.4
	See5	28.8 rules	74.5
Wine	ESC	4.4 clusters	85.3
	ESCGFCS	4.4 membership functions, 3 rules	95.5
	See5	5.4 rules	93.8
Zoo	ESC	7.9 clusters	84.5
	ESCGFCS	7.9 membership functions, 7 rules	94.3
	See5	8.4 rules	93.5

For both cases, the training classification rates of $n_{c_{\max}} = 7$ are higher those of $n_{c_{\max}} = 4$.
However, the testing classification rates make only a minor difference. The ESCGFCS results in a higher testing classification rate with $n_{c_{\max}} = 4$, i.e., the maximum four cluster are enough to classify patterns correctly.

Table 5. Performance comparison for different values of $n_{c_{\max}}$ (10-fold cross-validation).

Data	Algorithm	$n_{c_{\max}} = 7$		$n_{c_{\max}} = 4$	
		training	testing	training	testing
Breast Cancer	ESC	97.6	93.6	96.6	93.3
	ESCGFCS	98.4	96	97.8	96.1
Pima Indian	ESC	66.1	59	63.7	54.7
	ESCGFCS	79.9	75.4	77.3	75.8

5. CONCLUSION

Since both classification performance and discovering hidden structure are of major importance, focus is given to identifying clusters of simple structure through the proposed ESC and transforming them into fuzzy classifiers with better performance. ESC requires more computation time due to its iterative characteristics. However, both the parallel runs of DE for all the classes and the parallel implementation of DE itself can overcome this disadvantage. The optimal clusters identified by ESC are converted to Gaussian membership functions, and they constitute antecedent fuzzy sets in a set of fuzzy rules, where each rule classifies feature vectors into a class. The presented approach was tested by six classification problems and compared with results from different classifiers. The results show that the identified fuzzy classifiers have a better average performance with regard to accuracy. The ESCGFCS brings forth as many rules as pattern classes, and at most four membership functions in each rule is enough. Conclusively, the approach has the potential to improve classification accuracy at the cost of more computation time.

REFERENCE

[1] A. Demiriz, K. P. Bennett, and M. J. Embrechts, "Semi-supervised scaled clustering using genetic algorithms," *Intelligent Engineering Systems*

through Artificial Neural Networks, vol. 9, pp. 809-814, ASME Press, 1999.
[2] A. L. Corcoran and S. Sen, "Using real-valued genetic algorithm to evolve rule sets for classification," *Proc. of the First IEEE Conference on Evolutionary Computation*, vol. 1, pp. 120-124, 1994.
[3] B. P. Buckles, F. E. Petry, D. Prabhu, R. George, and R. Srikanth, "Fuzzy clustering with genetic search," *Proc. of the First IEEE Conference on Evolutionary Computation*, vol. 1, pp. 46-50, 1994.
[4] B. Yuan, G. J. Klir, and J. F. Swan-Stone, "Evolutionary fuzzy C-means clustering algorithm," *Proc. of the Fourth IEEE International Conference on Fuzzy Systems*, vol. 4, pp. 2221-2226, 1995.
[5] D. Mauck and R. Kruse, "Obtaining interpretable fuzzy classification rules from medical data," *Artificial Intelligence in Medicine*, vol. 16, pp. 149-169, 1999.
[6] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classifier systems for multidimensional pattern classification problems," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 29, no. 5, pp. 601-618, 1999.
[7] H. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. on Fuzzy Systems*, vol. 9, no. 4, pp. 516-524, 2001.
[8] H. Roubos, M. Setnes, and J. Abonyi, "Learning fuzzy classification rules from data," *Proc. Recent Advances in Soft Computing*, 2000.
[9] J. V. D. Oliveira, "Semantic constraints for membership function optimization," *IEEE Trans. on Systems, Man and Cybernetics, Part A*, vol. 29, no. 1, pp. 128-138, 1999.
[10] K. Nozaki, H. Ishibuchi, and H. Tanaka, "Trainable fuzzy classification systems based on fuzzy if-then rules," *Proc. of the Third IEEE Conference on Fuzzy Systems*, vol. 1, pp. 498-502, 1994.
[11] L. I. Kuncheva, and J. C. Bezdek, "Nearest prototype classification: clustering, genetic algorithm, or random search," *IEEE Trans. on System, Man and Cybernetics, Part C*, vol. 28, no. 1, pp. 160-164, 1998.
[12] L. O. Hall, J. C. Bezdek, S. Boggavarapu, and A. Bensaid, "Genetic fuzzy clustering," *Proc. of the First International Joint Conference of the North American Fuzzy Information Processing Society Biannual Conference*, pp. 411-415, 1994.
[13] R. Babuska, *Fuzzy Modeling for Control*, Kluwer Academic Press, 1998.
[14] R. Storn, "System design by constraint adaptation and differential evolution," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 1, pp. 22-34, 1999.
[15] S. Abe, R. Thawonmas, and M. Kayama, "A fuzzy classifier with ellipsoidal regions for diagnosis problems," *IEEE Trans. on Systems, Man and Cybernetics, Part C*, vol. 29, no. 1, pp. 140-148, 1999.
[16] S. Nascimiento and F. Moura-Pires, "A genetic approach for fuzzy clustering with a validity measure function," *Proc. of the Second International Conference on Intelligent Data Analysis*, pp. 325-335, 1997.
[17] See5/C5.0, http://www.rulequest.com/index.html.
[18] T. Van Le, "Evolutionary fuzzy clustering," *Proc. of the Second IEEE Conference on Evolutionary Computation*, vol. 2, pp. 753-758, 1995.
[19] UCI Repository of Machine Learning Databases, http:// www.ics.uci.edu/~mlearn/.

Heesoo Hwang received the B.S., M.S., and Ph.D. degrees in electrical engineering from Yonsei University in 1986, 1988, and 1993, respectively. From 1993 to 2000, he served as a senior and a principal engineer for the Korea High-Speed Rail Development project of Korea Railroad Research Institute and Korea High Speed Railway Construction Authority, respectively. Since 2001, he has been an assistant professor in the School of Electrical Engineering at Halla University. His research interests include evolutionary design, pattern classification, event prediction, and surveillance using fuzzy logic, neural networks, and evolutionary algorithms.