International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:1, No:1, 2007

Learning of Class Membership Values by Ellipsoidal Decision Regions

Leehter Yao and Chin-Chin Lin

***Abstract*—**A novel method of learning complex fuzzy decision regions in the n-dimensional feature space is proposed. Through the fuzzy decision regions, a given pattern's class membership value of every class is determined instead of the conventional crisp class the pattern belongs to. The n-dimensional fuzzy decision region is approximated by union of hyperellipsoids. By explicitly parameterizing these hyperellipsoids, the decision regions are determined by estimating the parameters of each hyperellipsoid. Genetic Algorithm is applied to estimate the parameters of each region component. With the global optimization ability of GA, the learned decision region can be arbitrarily complex.

***Keywords*—** ellipsoid, genetic algorithm, decision regions, classification

I. Introduction

PATTERN classification mainly deals with determination of decision regions based on the given prototypes. The information carried by the every prototype consists of the features associated with the prototype and the class the prototype belongs to. Since the information carried by each prototype are gathered by human beings, it is understood that uncertainty might exist within the information assigned to the prototype. Hence, fuzziness could be involved in feature space or in class assignment. For the ease of analysis and manipulation, most of the research in the field [1-7] gives crisp feature descriptions yet leave class assignment or classification fuzzily defined. Different from the deterministic classification [8-9] where each prototype is classified into one and only one class, membership degrees are employed in fuzzy classification defining degrees of belonging of each prototype to every class. There have been numerous approaches proposed for clustering [1, 10-14] based on the prototypes with fractional membership degrees belonging to different classes. However, not too many researches investigate learning of fuzzy decision regions based on the prototypes with fractional membership degrees belonging to every different class. The conventional fuzzy classification approach assigns each prototype's degree of belonging to different class by a real number between 0 and 1. The real number is called the class membership value. The training of conventional fuzzy classification approach is to find the model that determines the crisp classification based on the training prototype. For some classification applications such as medical diagnosis, geographical analysis or decision making, the classification aims to generate each pattern's class membership values for each class other than the crisp pattern class. For some applications, the crisp class that a given pattern is classified may not be as important as the class membership values since the class membership values will serve as important decision support data. In the paper, a novel approach is proposed to the train the decision region so that the class membership value of every class can be determined through the decision regions. A set of hyperellipsoids with adaptively tuned centers, orientations and sizes are employed to learn the fuzzy decision regions. If the prototypes are classified into c classes, there are in fact c decision regions to be learned since each prototype has crisply defined coordinates in the feature space and c respectively assigned membership degrees belonging to each of c classes. The proposed learning scheme for decision regions is basically nonparametric since no statistical information of the prototypes is assumed. The decision regions determined by the perceptron algorithm [15-16] or the least mean square algorithm [8-9] are compose of half spaces. Hyperellipsoids geometrically locates and cover the decision regions more precisely and yet requires more concise parameterizations.[17-18]

Since the decision region can be approximated by an union of a finite number of hyperellipsoids, learning of complex decision region is equivalent to the estimation of the parameters of hyperellipsoids. Multiple hyperellipsoids are respectively employed to approximate the decision regions in [19] and [20]. In [19], a multivariate Gaussian distribution function of prototypes is assumed. The locus of prototypes with constant probability density function forms a hyperellipsoid in the feature space. Therefore, [19] aims to learn the distribution function of prototypes, which is equivalent to learning the parameters of hyperellipsoids. The Genetic Algorithm (GA) is utilized to learn the hyperellipsoids in [20]. Both [19] and [20] consider only the prototypes with crisp classification, i.e., each prototype is assigned one and only one class. In this paper, the decision region for the prototypes with fuzzy classification will be investigated.

Similar to [20], the GA is also to be used in this paper to estimate the parameters of ellipsoids. The number of hyperellipsoids required to approximate the decision region is generally unknown in advance. More hyperellipsoids than

Manuscript received Nov. 19, 2004.

Leether Yao is with Dept. of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan. (phone : +886-2-2751-0623; fax : +886-2-2751-3892; e-mail : ltyao@ntut.edu.tw).

Chin-Chin Lin is with Dept. of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan. (e-mail : s0669013@ntut.edu.tw).

actually necessary are assigned for the learning. It will be shown that by appropriately defining the criterion function for minimization, GA is not only able to learn the parameters of each hyperellipsoids but also tends to minimize the total volume of hyperellipsoids. An efficient method will be proposed along with the parameter learning process by GA to identify redundant hyperellipsoids so that the redundant hyperellipsoids can be eliminated. With the global optimization ability of GA, the decision region to be learned can be arbitrarily complex including linearly inseparable, nonconvex and disconnected ones. The GA is implemented to learn the decision regions for all classes in parallel. However, the optimization of parameters to learn the decision region for each class is not performed separately since membership degrees belonging to different classes learned in every generation of GA are closely related. The optimization of GA is performed adjoining the constraints for membership degrees of all classes. By use of the fuzzy partition techniques in fuzzy c-means clustering algorithm [1], membership degrees are determined by minimizing the objective function adjoining the constraint by means of Lagrange multipliers. Although the learning of decision regions proposed in this paper is a supervised learning approach, the unsupervised learning scheme such as fuzzy c-means algorithm is employed along with GA.

The organization of this paper is as follows. The problem and some preliminaries are stated in section II. In section III, the object function of GA is described and analyzed. The ways of identifying redundant hyperellipsoids following the learning process by GA is also introduced. In section IV, the implementation of GA to learn the decision regions is described. The simulation results are shown in section V. Finally, conclusions are drawn in section VI.

II. Problem Statement

Fuzzy partition of pattern space is employed for the classification discussed in this paper so that membership degrees of respectively belonging to c classes are defined for every prototype. Let $W \subset R^n$ be the feature space and F be the set of m prototypes such that for every prototype $(\boldsymbol{x}_j, u_{ij}) \in F$, $\boldsymbol{x}_j \in W$ and $u_{ij} \in [0, 1]$, $i = 1...c$, $j = 1...m$. Two additional constraints need to be defined for the values of u_{ij}. The first one is to assume that membership degrees of $\boldsymbol{x}_j$ belonging to all c classes are equal to 1; that is,

$$\sum_{i=1}^{c} u_{ij} = 1. \forall j = 1...m. \tag{1}$$

The second constraint is to assume that every class is nonempty and different from the entire pattern space; that is,

$$0 < \sum_{j=1}^{m} u_{ij} < m, \forall i = 1...c. \tag{2}$$

The fuzzy classification problem investigated in this paper can be considered as determining c fuzzy decision regions G_1, G_2, ..., G_c. For every prototype $\boldsymbol{x}_j$, the degree of belonging to the decision region G_i is defined by the value of u_{ij}. Every fuzzy decision region is to be approximated by the union of a finite number of hyperellipsoids. The candidate hyperellipsoid is called the region component. Let S_{ip} be the p-th region component approximating G_i, S_{ip} can be expressed by the function f_{ip} as:

$$f_{ip}(\boldsymbol{x}; \boldsymbol{v}_{ip}, \boldsymbol{A}_{ip}) = (\boldsymbol{x} - \boldsymbol{v}_{ip})^T \boldsymbol{A}_{ip} (\boldsymbol{x} - \boldsymbol{v}_{ip}) \tag{3}$$

where $\boldsymbol{v}_{ip} \in W$ denotes the center of S_{ip} and $\boldsymbol{A}_{ip} \in R^{n \times n}$ is the matrix corresponding to the topological structure of the data. The matrix $\boldsymbol{A}_{ip}$ can be defined as

$$\boldsymbol{A}_{ip} = \boldsymbol{\Theta}_{ip} \boldsymbol{\Lambda}_{ip} \tag{4}$$

where $\boldsymbol{\Theta}_{ip}$ denotes the rotation of the region component in the n dimensional pattern space while $\boldsymbol{\Lambda}_{ip}$ is a diagonal matrix with the length of the region component's axes in each direction on the diagonal. If θ_{ipk}, is angle of rotation with respect to the k-th axis for S_{ip}, $k = 1...n$, , define the rotation element as

$$\Phi(\theta_{ipk}) = \begin{bmatrix} \cos\theta_{ipk} & -\sin\theta_{ipk} \\ \sin\theta_{ipk} & \cos\theta_{ipk} \end{bmatrix}, \tag{5}$$

and the rotation matrix $\boldsymbol{\Psi}(\boldsymbol{\theta}_{ipk}) \in R^{n \times n}$as

$$\Psi(\theta_{ipk}) = diag(1,...,1,\Phi(\theta_{ipk}),1,...,1) \equiv \begin{bmatrix} 1 & & & & & & 0 \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & \Phi(\theta_{ipk}) & & & \\ & & & & 1 & & \\ & & & & & \ddots & \\ 0 & & & & & & 1 \end{bmatrix} \tag{6}$$

where $\boldsymbol{\Phi}(\boldsymbol{\theta}_{ipk})$ is located in the k-th and $(k+1)$-th rows and columns of $\Psi(\theta_{ipk})$. Note, however, that

$$\Psi(\theta_{ipn}) = \begin{bmatrix} \cos\theta_{ipn} & 0 & \cdots & 0 & -\sin\theta_{ipn} \\ 0 & 1 & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & 0 \\ \sin\theta_{ipn} & 0 & \cdots & 0 & \cos\theta_{ipn} \end{bmatrix}. \tag{7}$$

Referring to (5) and (6), the matrix, $\boldsymbol{\Theta}_{ip}$, denoting the rotations of S_{ip} in the pattern space can be defined as

$$\Theta_{ip} = \prod_{k=1}^{n} \Psi(\theta_{ipk}) \; for \quad n \geq 3; \tag{8}$$

and $\Theta_{ij} = \Psi(\theta_{ip1})$ for $n = 2$. (9)

As for the matrix Λ_{ip}, if d_{ipk}, is the k-th axis of S_{ip}, $k = 1 \ldots n$, then Λ_{ip} can be defined as

$$\Lambda_{ip} = \begin{bmatrix} 1/d_{ip1}^2 & 0 & \cdots & 0 \\ 0 & 1/d_{ip2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/d_{ipn}^2 \end{bmatrix}. \tag{10}$$

In this paper, GA is utilized to learn the parameterizations of region components as shown in (3)-(10) based on the information carried by prototypes. For any fuzzy decision region, G_i, the number of region components required to approximate G_i is generally unknown a priori. More than enough candidate region components are thus assigned to learn to fuzzy decision region based on the prototypes $(\boldsymbol{x}_j, u_{ij})$, $j = 1...m$. If h_i region components are necessary to approximate G_i by intuitive assessment or experience, b_i $(>h_i)$ region components with different parameterizations as in (3)-(10) are set to learn G_i. Let Q_i be the set of b_i candidate region components to approximate the fuzzy decision region G_i. Then,

learning fuzzy decision region is equivalent to tuning the parameters of region components in Q_i so that

$$Q_i \equiv \bigcup_{p=1}^{b_i} (S_{ip} \cap W) \approx G, i = 1...c_i \tag{11}$$

III. Learning of Fuzzy Decision Regions

A. Fitness Function of GA

GA learns different sets of region components $Q_1, Q_2, ..., Q_c$ to approximate the fuzzy decision regions $G_1, G_2, ..., G_c$. The approximation is performed in the sense of minimum misclassification errors as well as minimum total volume of region components. GA is implemented to learn the fuzzy decision regions $G_1, G_2, ..., G_c$, in parallel while satisfying the constraints of fuzzy partitions in (1) and (2) at the same time.

Since the volume of a region component S_{ip} is proportional to the determinant of $\boldsymbol{\Lambda}_{ip}^{-1}$, the fitness function for GA to learn the parameters of region components approximating G_i is given as

$$e = e_a + \gamma \sum_{i=1}^{c} \sum_{p=1}^{b_i} \prod_{k=1}^{n} d_{ipk}^2 \tag{12}$$

where e_a denotes the misclassification errors due to the set of region components; and γ weights the total volume of region components with respect to the misclassification errors. Based on the misclassification errors as well as the total volume of region components, GA is able to tune the parameters of each region component to minimize the misclassification errors as well as to adjust the sizes and orientations of region components to geometrically approximate the fuzzy decision region in parallel.

Referring to (3)-(10), the parameters of a region component S_{ip} to be learned by GA are the coordinates of center $\boldsymbol{v}_{ip} \equiv [v_{ip1}, v_{ip2}, ..., v_{ipn}]^T$, the angles of rotations in each direction $\boldsymbol{\theta}_{ip} \equiv [\theta_{ip1}, \theta_{ip2}, ..., \theta_{ipn}]^T$ and the length of each axes of the ellipsoid $\boldsymbol{d}_{ip} \equiv [d_{ip1}, d_{ip2}, ..., d_{ipn}]^T$, $i = 1...c$, $p = 1...b_i$. In order to learn the fuzzy decision region G_i, it is shown in (11) that more than enough region components S_{ip} are set to learn by GA. The S-norm of an union in (11) is operated by *max(·)*. Referring to (11) and (3), the distance between the *j-th* prototype and the region component Q_i can be calculated by:

$$\begin{aligned} y_{ij} &= \bigvee_{p=1}^{b_i} f_{ip}(\boldsymbol{x}_j; \boldsymbol{v}_{ip}, \boldsymbol{\theta}_{ip}, \boldsymbol{d}_{ip}) \\ &= \bigvee_{p=1}^{b_i} (\boldsymbol{x}_j - \boldsymbol{v}_{ip})^T \boldsymbol{\Theta}_{ip} \boldsymbol{\Lambda}_{ip} (\boldsymbol{x}_j - \boldsymbol{v}_{ip}) \end{aligned} \tag{13}$$

where $\vee$ denotes the operation of maximization. Similar to the fuzzy partition techniques utilized in fuzzy c-means algorithm, membership degrees are determined by minimizing the following objective function adjoining the constraint in (1):

$$J = \sum_{i=1}^{c} \sum_{j=1}^{m} \hat{u}_{ij}^{\eta_i} y_{ij} + \sum_{j=1}^{m} \lambda_j (1 - \sum_{i=1}^{c} \hat{u}_{ij}) \tag{14}$$

where $\hat{u}_{ij}$ is the estimated membership degree associated with y_{ij} in (13), η_i is a weighting exponent which determines the fuzziness of membership degrees $\hat{u}_{ij}$, and λ_j is the Lagrange multiplier. Membership degrees $\hat{u}_{ij}$ are determined by minimizing the objective function in (14); they thus can be obtained by setting

$$\frac{\partial J}{\partial \hat{u}_{ij}} = 0, \tag{15}$$

and

$$\frac{\partial J}{\partial \hat{u}_{ij}} = 0, \tag{16}$$

Solving $\hat{u}_{ij}$ in (15) and (16), gives

$$\hat{u}_{ij} = \frac{(\frac{1}{y_{ij}})^{\frac{1}{\eta_i - 1}}}{\sum_{i=1}^{c} (\frac{1}{y_{ij}})^{\frac{1}{\eta_i - 1}}}, i = 1...c, j = 1...m \tag{17}$$

The total misclassification errors e_a in (12) can be calculated by

$$e_a = \sum_{i=1}^{c} \sum_{j=1}^{m} (u_{ij} - \hat{u}_{ij})^2 \tag{18}$$

In order to calculate the misclassification error for each prototype $(\boldsymbol{x}_j, u_{ij})$, the distance between $\boldsymbol{x}_j$ and region components $Q_1, Q_2, ..., Q_c$, denoted by $y_{1j}, y_{2j}, ..., y_{cj}$, are first calculated by (13). The membership degrees of the given prototype belonging to each class are then calculated by (17) based on $y_{1j}, y_{2j}, ..., y_{cj}$. Finally, the misclassification error corresponding to the given prototype is calculated by (18). Therefore, although region components $Q_1, Q_2, ..., Q_c$ are learned in parallel to approximate the fuzzy decision regions $G_1, G_2, ..., G_c$, the learning for $Q_1, Q_2, ..., Q_c$ are cross correlated.

Note that since the weighting exponent η_i (≥ 1) has significant influence on the fuzziness of fuzzy partition, it is also learned by GA along with the parameters of each region component, i.e., $\boldsymbol{v}_{ip}$, $\boldsymbol{\theta}_{ip}$ *and* $\boldsymbol{d}_{ip}$, $i = 1...c$, $p = 1...b_i$, based on the fitness function in (12) and (18).

In fuzzy c-means algorithm, cluster center is iteratively adjusted so that the total distance norms between the prototype and every cluster center are minimized. In (13), the Mahalanobis norm is used for measuring the distance between the prototype and the center of Q_i, S_{ip}, $i = 1...c$, $p = 1...b_i$. Similar to fuzzy c-means algorithm, a virtual cluster center $\boldsymbol{v}_i'$ and the virtual distance norm D_{ijA}^2 are induced from (13) for each fuzzy decision region G_i corresponding to the prototypes $(\boldsymbol{x}_j, u_{ij})$, $i = 1...c$, $j = 1...m$, i.e.,

$$D_{ijA}^2 = \left\| \boldsymbol{x}_j - \boldsymbol{v}_i' \right\|_A^2 = \bigvee_{p=1}^{b_i} (\boldsymbol{x}_j - \boldsymbol{v}_{ip})^T \boldsymbol{\Theta}_{ip} \boldsymbol{\Lambda}_{ip} (\boldsymbol{x}_j - \boldsymbol{v}_{ip}) \tag{19}$$

Instead of iteratively determining the cluster center as in fuzzy c-means algorithm, centers as well as other parameters of Q_i are iteratively determined by GA in this paper. Since the distance norm in (19) is not a linear function, centers of Q_i cannot be calculated by differentiating the objective function with respect to the parameters of region component center ($\boldsymbol{v}_{ip}$) as in the regular fuzzy c-means algorithm. GA is thus utilized instead as the tool of optimization. Both fuzzy c-means algorithm and the method proposed in this paper, aim to minimize the total distance norms as in (19) between prototypes and each cluster center. For fuzzy c-means algorithm, the minimization of distance norms are performed by iteratively tuning cluster center as the weighted topological mean among the given

prototypes, where the weights are the membership degrees calculated based on the distance norms between prototypes and cluster center. The minimization of distance norms proposed in this paper is nevertheless achieved by iteratively tuning a set of candidate hyperellipsoids to cover the prototypes based on the membership degree associated with each prototype.

B. Trimming of Redundant Region Components

Since suitable number of region components required to learn the fuzzy decisions is generally unknown. To begin with, more than enough region components are assigned to learn the fuzzy decision regions. It might be possible that GA tunes the parameters of region components so that less number of region components are well conglomerated to approximate the fuzzy decision region. In other words, redundant region components might exist after the learning process by GA. It is thus necessary to design a scheme to trim off those redundant region components to increase the classification accuracy. In order to determine the redundant region components, the contribution degree of each estimated region component is defined. Referring to (13), b_i region components are set to conglomerate as Q_i to approximate the fuzzy decision region G_i. In the feature space $W \subset R^n$, there is one and only one region component that is closest to a prototype $\boldsymbol{x}_j$, $j = 1...m$. For any p-th region component learning the i-th fuzzy decision region, S_{ip}, $p \in [1, b_i]$, define the threshold function $T(\cdot)$ as

$$T(\boldsymbol{x}_j, S_{ip}) = \begin{cases} 1, & \text{if } f_{ij}(\boldsymbol{x}_j; \boldsymbol{d}_{ip}, \boldsymbol{v}_{ip}, \boldsymbol{\theta}_{ip}) > f_{ij}(\boldsymbol{x}_j; \boldsymbol{d}_{iq}, \boldsymbol{v}_{iq}, \boldsymbol{\theta}_{iq}), \forall q \in [1, b_i], q \neq p; \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

The contribution degree of an estimated region component S_{ip} can be defined as the number of the prototypes that are closest to it. Let $g(S_{ip})$ be the contribution degree of S_{ip}, $i = 1...c$, $p = 1...b_i$, then

$$g(S_{ip}) = \sum_{j=1}^{m} T(\boldsymbol{x}_j, S_{ip}) \tag{21}$$

With the contribution degree given as in (21), the redundant region components can thus be defined as the ones with contribution degree less than a preset value. Within Q_i, let the redundant region components be $\overline{S}_{ir}$, $r \in [1, b_i]$, then

$$g(\overline{S}_{ir}) < \alpha \tag{22}$$

where α is a preset threshold value determining whether the evaluated region component is considered to be redundant. If β_i region components are determined to be redundant, trimming off these redundant region components from Q_i will thus increase the classification efficiency since fewer coefficients are required to parameterize fuzzy decision regions. Let $\hat{Q}_i$ be the refined set of candidate region components to approximate fuzzy decision region G_i, then referring to (11),

$$\hat{Q}_i = Q_i - \bigcup_{i=1}^{\beta_i} (\hat{S}_{ip} \cap W), \quad i = 1...c \tag{23}$$

IV. Implementation of the Genetic Algorithm

Each n-dimensional region component S_{ip} is parameterized by the coordinates of center $\boldsymbol{v}_{ip}$, the angles of rotations in each direction $\boldsymbol{\theta}_{ip}$ and the length of each axes $\boldsymbol{d}_{ip}$, $i = 1...c$, $p = 1...b_i$. Along with the weighting exponent η_i, there are $(3n+1)$ parameters to be learned by GA for each region component. Every estimated parameter is encoded as a string of binary digits. The binary strings are then cascaded to form a chromosome. If c decision regions are to be determined from the information carried by prototypes, only (c-1) sets of parameters need to be learned since the c-th decision region can be determined based on the constraint in (1). That is, membership degree of the c-th decision region is determined by subtracting all the membership degrees belonging to other classes from 1. Therefore, total number of parameters encoded in one chromosome is given by

$$\kappa = (3n+1) \sum_{i=1}^{c-1} b_i \tag{24}$$

Referring to (12) and (18), fuzzy decision regions G_1, $G_2, ..., G_c$ are learned by optimizing $\boldsymbol{v}_{ip}$, $\boldsymbol{\theta}_{ip}$, $\boldsymbol{d}_{ip}$ and η_i via GA, i.e.

$$(\boldsymbol{v}_{ip}, \boldsymbol{\theta}_{ip}, \boldsymbol{d}_{ip}, \eta_i)\big|_{i=1..c} = \underset{i=1..c,\, p=1..b_i}{\arg\min} \left(\sum_{i=1}^{c} \sum_{j=1}^{m} (u_{ij} - \hat{u}_{ij})^2 + \gamma \sum_{i=1}^{c} \sum_{p=1}^{b_i} \prod_{k=1}^{n} d_{ipk}^2 \right) \tag{25}$$

V. Numerical Simulation

In this section, a numerical example is presented verifying the proposed algorithm. It will be shown that the number of region components required to approximate the decision region for every class are unknown a priori. Although more than enough region components are assigned to learn the decision region, the redundant region components can be easily identified and trimmed off based on the learning results. The weighting coefficient γ in (12) is set to be 0.15. In order to assess the learning efficiency and accuracy, the prototypes are divided into two parts; one part for the learning and the other part for the test. In this example, 695 prototypes with 10% noise are set for the learning. The membership degrees for class 1, 2 and 3 of these 695 prototypes are shown in Fig. 1, respectively. The distribution of prototypes in this example is not linearly separable. The decision region of every class is more complicated and more difficult to learn. In this example, 12 region components are assigned, respectively, to learn each of 3 classes. The learning results that 6 out of 12 candidate region components for both class 1 and 2, 5 out of 12 candidate region components for both class 3 remain based on contribution degree of each region component. The decision regions for class 1, 2 and 3 are shown in Fig. 2, respectively. As the decision regions are learned based on the prototypes, another set of 695 data is employed for test. The average misclassification error $\overline{e}$ is calculated to be 0.00881. The learning of decision regions is still accurate and efficient.

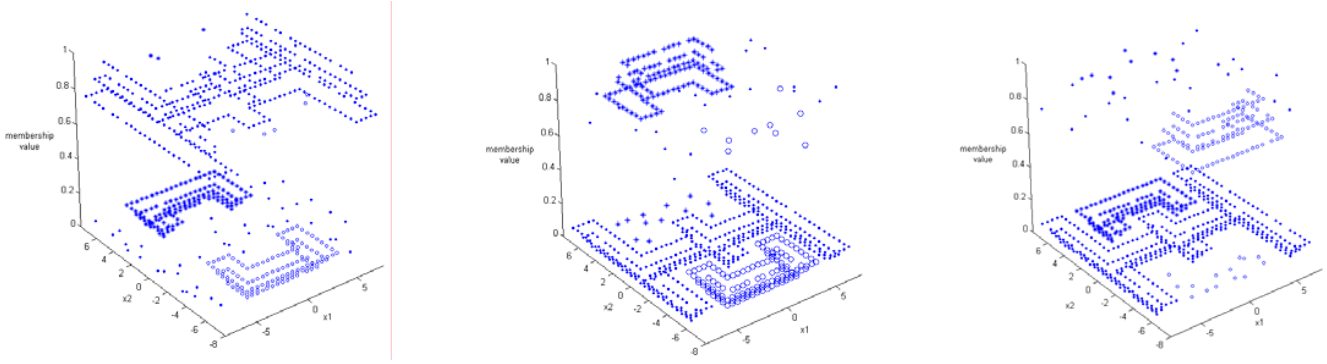
VI. Conclusion

It has been shown in this paper that the desired fuzzy decision region is approximated by a finite number of ellipsoids. By appropriately parameterizing the hyperellipsoids, the GA is applied to estimate the associated parameters, and thus to learn

the decision region. Since the minimization criterion of GA is defined to be a combination of misclassification error and the sum of the volume of the estimated hyperellipsoids, the learning hyperellipsoids tends to agglomerate together with the least total volume to approximate the decision regions. Compared to the traditional methods such as statistical approaches or artificial neural networks that approximate decision regions with half spaces, the proposed method locates and approximates the decision regions more precisely and yet employs more concise parameterizations.

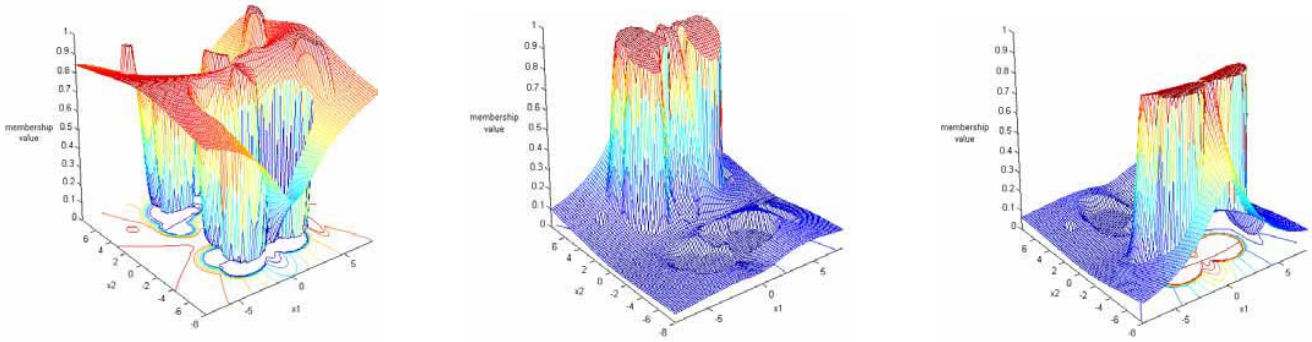
REFERENCES

[1] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum Press, 1981.
[2] J. C. Bezdek, S. K. Pal, eds., *Fuzzy Models for Pattern Recognition*, New York: IEEE Press, 1992
[3] J. C. Bezdek, " Computing with uncertainty, " *IEEE Commun. Mag*., vol. 30, Sept. 1992, pp. 24–37
[4] S. K. Pal, D. K. Dutta Majumber, *Fuzzy Mathematical Approach to Pattern Recognition*, New York: John Wiley, 1986.
[5] S. K. Pal, S. Mitra, Multilayer perception, fuzzy sets and classification, *IEEE Trans. Neural Networks*, vol. 3, May 1992, pp. 683–697
[6] W. Pedrycz, "Fuzzy sets in pattern recognition: methodology and methods," *Pattern Recognition*, vol. 23, 1990, pp. 121-146
[7] W. Pedrycz, "Fuzzy neural networks with reference neurons as pattern classifiers," *IEEE Trans. Neural Networks*, vol. 3, May 1992, pp. 770–775
[8] S. T. Bow, *Pattern Recognition Application to Large Dataset Problems*, New York: Marcel Dekker, 1980.
[9] R. S.chalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches*, New York: John Wiley & Sons, 1992.
[10] J. C. Bezdek, J. M. Keller, R. Krishnapuram, N. R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Norwell, MA: Kluwer, 1999.
[11] R. J. Hathaway, J. C. Bezdek, Y. Hu, "Generalized fuzzy c-means clustering strategies using L_p norm distances," *IEEE Trans. Fuzzy Syst.,* vol. 8, May 2000, pp. 576–582
[12] R. Krishnapuram, J. M. Keller, "A possibility approach to clustering," *IEEE Trans. Fuzzy Syst.,* vol. 1, Mar. 1993, pp. 98-110
[13] L. Zhao, Y. Tsujimura, M. Gen, "Genetic algorithm for fuzzy clustering," *Proceeding of IEEE International Conference on Evolutionary Computation*," 1996, pp. 716-719.
[14] B. P. Buckles, F. E. Petry, D. Prabhu, R. George and R. Srikanth, " Fuzzy clustering with genetic search," *Proceeding of IEEE World Congress on Coimputational Intelligence*, 1994, pp. 46-50.
[15] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.,* April 1987, pp. 4-22.
[16] T. Khanna, *Foundations of Neural Networks*, MA: Addison-Wesley, 1990.
[17] H. I. Avi-Itzhak, J. A. Van Mieghem, L. Rub, "Multiple subclass pattern recognition: a maximum correlation approach," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 17, April 1995, pp. 418-431.
[18] Q. Zhu, Y. Cai, "A subclass model for nonlinear pattern classification," *Pattern Recognition Lett.*, vol. 19, Feb. 1998, pp. 19-29.
[19] Q. Zhu, Y. Cai, L. Liu, "A multiple hyper-ellipsoidal subclass model for an evolutionary classifier," *Pattern Recognition*, vol. 34, March 2001, pp. 547-560.
[20] L. Yao, "Nonparametric learning of decision regions via the genetic algorithm," *IEEE Trans. System, Man, and Cybernetics*, vol. 26, Feb. 1996, pp. 313-321.



(a) (b) (c)

Fig. 1. The distribution of prototypes. (a) class 1; (b) class 2; (c) class3.



(a) (b) (c)

Fig. 2. The decision regions for (a) class 1; (b) class 2; (c) class3.