Constructing Accurate Fuzzy Classification Systems: A New Approach Using Weighted Fuzzy Rules

S.M. Fakhr Ahmad [a], M. Zolghadri Jahromi [b]
[a] *Department of Computer Engineering, School of Engineering, Islamic Azad University of Shiraz, Shiraz, Iran*
[b] *Department of Computer Science &Engineering, School of Engineering, Shiraz University, Shiraz, Iran*
mfakhrahmad@cse.shirazu.ac.ir , zjahromi@shirazu.ac.ir

Abstract

Different approaches to design fuzzy rule-based classification systems can be grouped into two main categories: descriptive and accurate. In the descriptive category, the emphasis is on the interpretability of the resulting classifier. The classifier is usually represented by a set of short fuzzy rules (i.e., with a few number of antecedent conditions) that make it a suitable tool for knowledge representation. In the accurate category, the generalization ability of the classifier is the main target in the design process and no attempt is made to use understandable fuzzy rules in constructing the rule base. In this paper, we propose a simple and efficient method to construct an accurate fuzzy classification system. We use rule-weight as a simple mechanism to tune the classifier and propose a new method of rule-weight specification for this purpose. Through computer simulations on some data sets from UCI repository, we show that the proposed scheme achieves better prediction accuracy compared with other fuzzy and non-fuzzy rule-based classification systems proposed in the past.

Keywords--- **Fuzzy systems, Classification, Rule-weight, Generalization Accuracy**

1. Introduction

Fuzzy rule-based systems have been widely applied to various application areas such as classification. A Fuzzy Rule-Based Classification System (FRBCS) is a special case of fuzzy Rule-Based systems where the output of the system is crisp and discrete. Different classifiers can be grouped into two main categories: *descriptive* and *accurate*. The main advantage of *descriptive* classification systems in comparison with their counterparts is their interpretability. The classifier is usually represented by a set of short fuzzy rules (i.e., with a few number of antecedent conditions) that make it a suitable tool for knowledge representation. However, the main goal of many designers, who want to develop *accurate* classification systems (either fuzzy or non-fuzzy), is to maximize the generalization accuracy of the classifier. In these approaches no attempt is made to improve the understandability of the system.

The fuzzy if-then rules used for classification problems have commonly the following form:

Rule R_j: If x_1 is A_{j1} and ... and x_n is A_{jn} then Class C_j with CF_j , $j \in 1,2,...,N$, (1)

where $\mathbf{x} = (x_1,..., x_n)$ is an n-dimensional pattern vector, A_{ji} is an antecedent linguistic value such as *young* and *old* ($i \in 1,2,..., n$) , C_j is a consequent class (i.e., one of the given c classes), N is the number of fuzzy if-then rules, and CF_j is the certainty grade of the rule R_j which usually has a real value in the unit interval [0, 1] (i.e., $0 \leq CF_j \leq 1$).

It has already been shown that various methods of rule weighting have a significant effect on the classification performance of fuzzy and non-fuzzy systems [1,2].

Among non-fuzzy approaches, the k-nearest neighbor (K-NN) rule is a simple and *accurate* pattern classification algorithm. However, the problems emerge in cases that patterns of different classes overlap in some regions in the feature space. Many researchers have already developed various adaptive or discriminate metrics to improve the performance of the K-NN. In [3], Wang et al. showed that a simple adaptive distance measure significantly improves the performance of the k-NN.

In this paper, we propose a simple and efficient method to construct an *accurate* fuzzy classification system. We use rule-weight as a simple mechanism to tune the classifier and propose a new method of rule-weight specification for this purpose. Having an initial rule-base for a problem, our learning method assigns a weight to each rule in the rule-base. The base of the learning algorithm is a process that finds the optimal decision area for each rule according to the target classes of training patterns. In other words, it moves the boundaries of decision areas so as to the majority of the training patterns within an area be of the same class. This leads to the minimal error rate of the classifier over training data. Using this learning mechanism, the generalization accuracy of the classifier will also be improved, significantly.

In this approach, the number of rules does not grow exponentially as the number of features and fuzzy sets increase. The number of rules at most equals the number of training patterns and thus, no process for rule subset selection is needed.

Rule weighting is rarely used in research works on FRBCSs (e.g. [2]). Instead, in many cases, they modify membership functions of antecedent fuzzy sets using numerical data, which involves learning a number of parameter values for each membership function. Thus, rule weighting is a much easier approach that improves the performance of the classifier without changing the position of fuzzy sets given by domain experts.

The rest of this paper is organized as follows. In Section 2, a brief introduction for FRBCS is given. In Section 3, the rule weight learning mechanism is discussed. Section 4 is devoted to the method of finding the best boundaries for the decision area of a typical rule. The experimental results are presented in Section 5. Finally, a conclusion is given at the end of the paper.

2. Fuzzy Rule-Based Classification Systems (FRBCS)

A Fuzzy Rule-Based Classification System (FRBCS) is composed of three main conceptual components: database, rule-base and reasoning method. The database describes the semantic of fuzzy sets associated to linguistic labels. Each rule in the rule-base specifies a subspace of pattern space using the fuzzy sets in the antecedent part of the rule. The reasoning method provides the mechanism to classify a pattern using the information from the rule-base and database. Different rule types have been used for pattern classification problems [3]. We use fuzzy rules of the following type for an n-dimensional problem.

Rule R_j: If x_1 is A_{j1} and … and x_n is A_{jn} then class h with CF_j (2)

where, $X=[x_1, x_2, \ldots, x_n]$ is the input feature vector, $h \in [C_1, C_2 \ldots, C_M]$ is the label of the consequent class, A_{jk} is the fuzzy set associated to x_k , CF_j is the certainty grade (i.e. rule weight) of rule R_j and N is the number of fuzzy rules in the rule-base.

In order to classify an input pattern $X_t = [x_{t1}, x_{t2}, \ldots, x_{tn}]$, the degree of compatibility of the pattern with each rule is calculated (i.e., using a T-norm to model the "and" connectives in the rule antecedent). In case of using product as T-norm, the compatibility grade of rule R_j with the input pattern X_t can be calculated as:

$$\mu_j(X_t) = \prod_{i=1}^{n} \mu_{A_{ji}}(x_{ti}) \tag{3}$$

In the case of using single winner reasoning method, the pattern is classified according to consequent class of the winner rule R_w. With the rules of form (1), the winner rule is specified using:

$$w = \arg\max \{\mu_j(X_t) CF_j ,\ j = 1,\ldots,N\} \tag{4}$$

Note that the classification of a pattern not covered by any rule in the rule-base is rejected. The classification of a pattern X_t is also rejected if two rules with different consequent classes have the same value of $\mu(X_t).CF$ in equation (3).

In case of using weighted vote [4] as the reasoning mechanism, each fuzzy rule gives a vote for its consequent class. The strength of the vote given by each rule can be defined as the product of compatibility grade and certainty grade. The total strength of the vote for each class can be calculated as follows.

$$\sigma_{Class\ T}(X_t) = \left\{ \sum_{j=1}^{N} \mu_j(X_t) CF_j \mid R_j \in S, Consequent(R_j) = ClassT \right\} \tag{5}$$

Where, S represents the rule-base. In this case, a test pattern X_t is classified as the class having maximum total strength.

3. Rule-base construction

For an M-class problem in an n-dimensional feature space, assume that m labeled patterns $X_p=[x_{p1}, x_{p2}, \ldots, x_{pn}]$, $p=1, 2, \ldots, m$ from M classes are given. A simple approach for generating fuzzy rules is to partition the domain interval of each input attribute using a pre-specified number of fuzzy sets (i.e., grid partitioning), denoted by k. Some examples of this partitioning (using triangular membership functions) are shown in Fig. 1.

Given a partitioning of pattern space, one approach is to consider all possible combination of antecedents to generate the fuzzy rules. The selection of the consequent class for an antecedent combination (i.e. a fuzzy rule) can be easily expressed in terms of confidence of an association rule from the field of data mining [5]. A fuzzy classification rule can be viewed as an association rule of the form $A_j \Rightarrow class\ C_j$, where, A_j is a multi-dimensional fuzzy set representing the antecedent conditions and C_j is a class label. Confidence (denoted by C) of a fuzzy association rule R_j is defined as [6]:

$$C(A_j \Rightarrow class\ C_j) = \frac{\sum_{X_p \in class\ C_j} \mu_j(X_p)}{\sum_{p=1}^{m} \mu_j(X_p)} \quad (6)$$

Where, $\mu_j(X_p)$ is the compatibility grade of pattern X_p with the antecedent of the rule R_j, m is the number of training patterns and C_j is a class label. The consequent class C_q of an antecedent combination A_j is specified by finding the class with maximum confidence. This can be expressed as:

$$q = \arg\max\{C(A_j \Rightarrow class\ h \mid h = 1,2,...,M\} \quad (7)$$

Note that, when the consequent class C_q can not be uniquely determined, the fuzzy rule is not generated.

The problem with grid partitioning is that for an n-dimensional problem, k^n antecedent combinations should be considered. It is impractical to consider such a huge number of antecedent combinations when dealing with high dimensional problems.

In order to prevent from exponential growth of the rule-base, we just generate rules that have at least one training pattern in their decision areas. Since the decision areas of different rules do not overlap, the number of rules will be at most equal to the number of training patterns.

3.1. The proposed method for rule weighting

Initially, all rules are assumed to have a weight of one (i.e. CF_k=1, K=1,2,...,N). In this section we propose an algorithm to assign some real numbers (in the interval [1,∞)) as the rule weights using the training patterns. The rule-weighting process for a typical rule, R_i, can be organized into the following steps:

1. Specify the center of the rule's covering area, sort the training patterns within this area in ascending order of their distance from the center (The first pattern in the sorted list is the most compatible one with the rule).
2. Scan the patterns through the sorted list until a pattern (X_n) from the negative class (any class except the rule's target class) is met (The enemy pattern with maximum compatibility degree is found).
3. Call the pattern just before the enemy pattern in the list X^* and Find its compatibility with the rule R_i ($\mu_i(X^*)$).
4. Compute the rule's weight (CF_i) using the following equation:

$$CF_i = 1/\mu_i(X^*) \quad (8)$$

This algorithm obtains a real number in the interval [1,∞) as the weight of each rule. However, in this issue, two exceptional cases may occur:

1. The first pattern in the sorted list is an enemy pattern. For this case, we set the value of 1 to $\mu_i(X^*)$ and thus the rule's weight will not change from 1.
2. There is no enemy pattern in the covering area of the rule (i.e., an interesting case). For this case, we chose the compatibility degree of the last pattern in the sorted list for $\mu_i(X^*)$, i.e., $\mu_i(X^*)$ = μ_i(last pattern). Since the last pattern has the minimum compatibility with the rule, a higher weight is given to such rules.

In this method, no rule is given a weight of 0. Thus, the number of rules does not change through this weighting process. As the number of partitions of each feature increases, the performance of the system approaches the performance of weighted K-NN method, while having the extra advantage of interpretability, especially for low-dimensional data sets.

4. Finding the optimal decision boundaries

The method proposed in Section 3, increases the classification accuracy of each fuzzy rule over training data up to 100%. This is accomplished by tuning the boundaries for the decision area of each rule through assigning a weight to it. It can be predicted that the generalization ability of the classifier will be improved, too. However, there is really no reason that we will get the optimal results for generalization accuracy, through this issue. The main reason refers to some probable noisy or exceptional patterns or in case of data sets with highly ovelapped classes. Our proposed method can easily become more flexible (for noisy data) by making a small change to it. After determining a threshold for the rule accuracy, we do not stop the scanning of the sorted list when meeting the first enemy pattern. Instead, we extend the decision boundary of the rule until we reach an enemy pattern that makes the rule's accuracy become less than the specified threshold. In other words we let a few number of enemy patterns to exist in the decision area of each rule. This can be more effective for noisy-nature data sets.

To find the best accuracy threshold for a typical data set, different threshold values can be tested through the discussed method and the optimal value will be obtained by comparing the generalization accuracies of different cases.

5. Experimental results

In order to evaluate the performance of the proposed scheme, we used the data sets shown in Table 1 available from UCI ML repository. To construct an initial rule-base for a specific data set, a number of equi-length rules (number of antecedents equaling to the number of features), having at least one training pattern in their decision areas were generated. In order to assess the effect of the proposed scheme in comparison with its

Computer Graphics, Imaging and Visualisation (CGIV 2007)
0-7695-2928-3/07 $25.00 © 2007 IEEE

alternatives, we used 10CV technique which is a case of n-fold cross validation.

In the first part of the experiment, the generalization accuracy of the initial rule-base (before rule weighting) was measured.

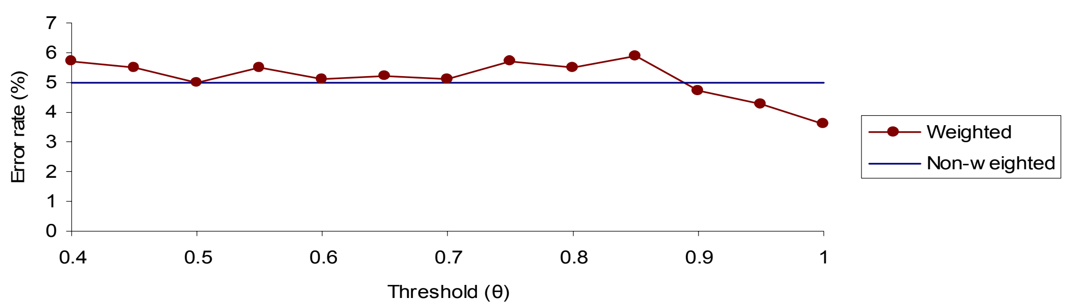
In the second part, our rule-weighting method was evaluated without considering any accuracy threshold, as discussed in Section 4. In other words, the threshold was set to 1. The results, shown in Table 2 narrate from a positive effect for the weighting method over the generalization ability.

Finally, in the third part, for each data set, we tried different values of the accuracy threshold by changing it from 1 down to 0.4 (by the step size of 0.05). Using the LVO (Leave One Out) technique, in each case, the error rate was measured using training data. The best threshold (leading to the best result) was then selected to evaluate the generalization ability over that data set using the 10CV technique. The results of this part are shown in Figures 1.a through 1.f. These figures indicate the error rates of the classifier for each data set through different values of the threshold (The threshold is denoted by θ).

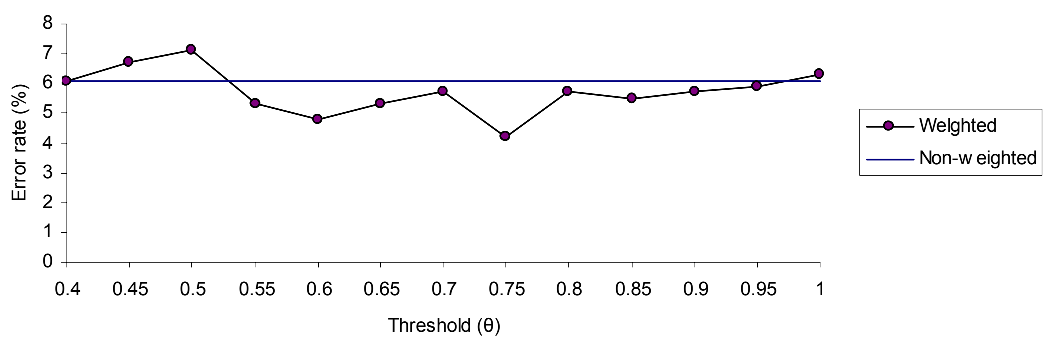
The straight line in each figure represents the error rate of the classifier without rule weighting and has been used to easily see the effect of rule weights in different cases. The error rates of the classifier using the optimal value of θ for each data set are presented in Table 2. In this table, our proposed method is also compared with another successful rule-based method as benchmark results called C4.5 reported by Elomaa and Rousu [7]. As shown in Table 2, except in one case, the proposed classifier in this paper results in better classification rates, compared to the best results already achieved by C4.5.

Table 1 Some statistics of the data sets used in our computer simulations

Data set	Number of attributes	Number of patterns	Number of classes
Iris	4	150	3
Wine	13	178	3
Thyroid	5	215	3
Sonar	60	208	2
Bupa	6	345	2
Pima	8	768	2
Glass	9	214	6

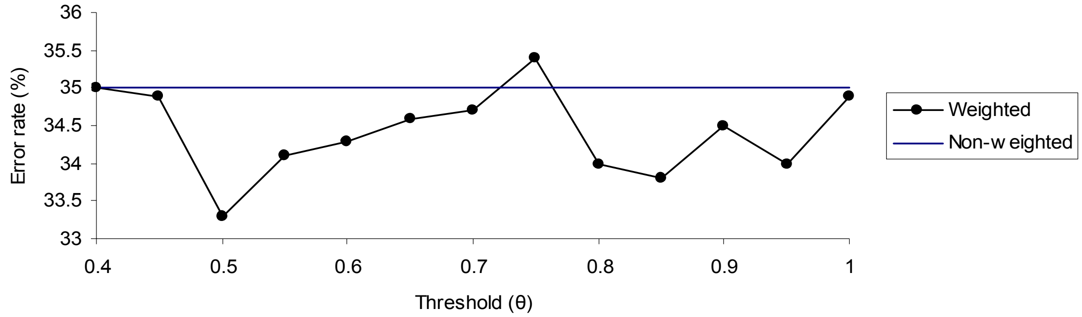


a) Iris

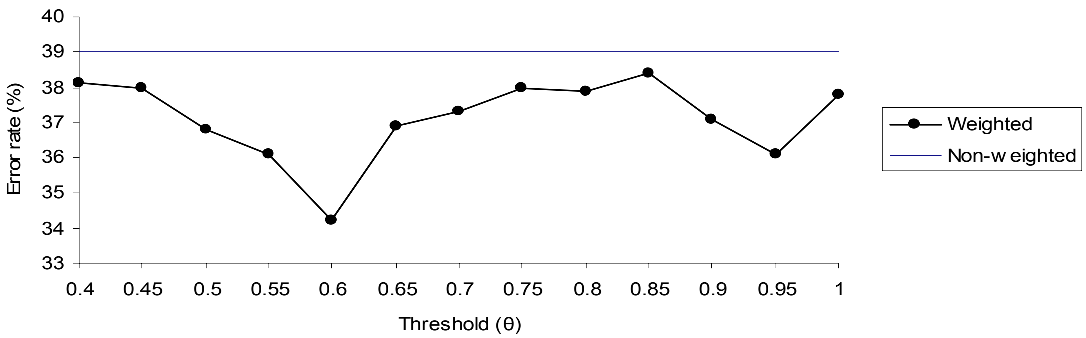


b) Wine

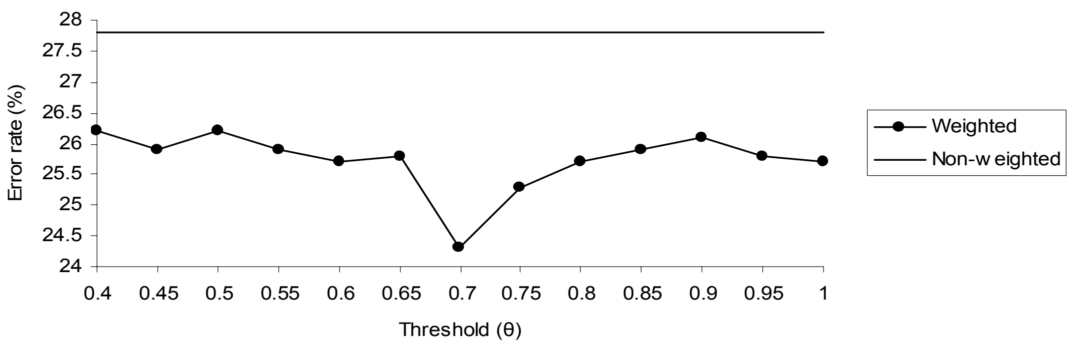
Computer Graphics, Imaging and Visualisation (CGIV 2007)
0-7695-2928-3/07 $25.00 © 2007 IEEE



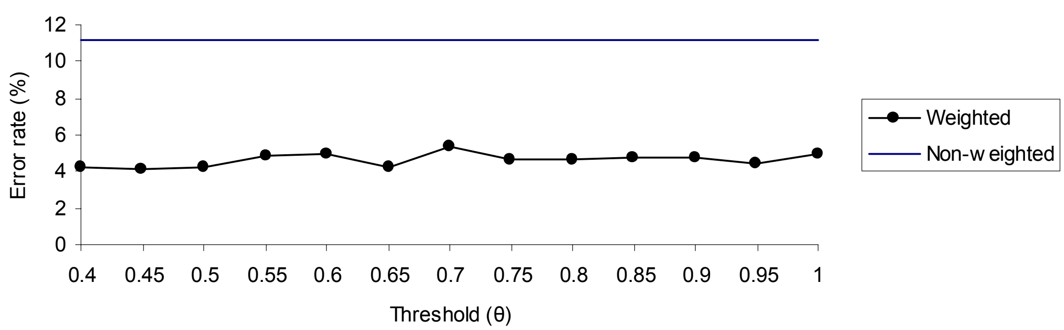
c) Glass



d) Bupa



e) Pima



f) Sonar

Figure 1 Error rates of the proposed classifier using different threshold values and comparison with non-weighted rule-base classifier for data sets of Table 1

Table 2 Classification Error rates of the proposed classifier using the optimal threshold values and comparison with threshold of 1, non-weighted rule-base classifier and the C4.5 method for data sets of Table 1

Data sets	Error Rates (%)			
	No. Weight	Weighted rules		C4.5 (best results)
		$\theta = 1$	Optimal θ	
Iris	5	3.6	3.6	5.1
Wine	6.1	6.3, 5.2	5.1	5.6
Pima	27.8	25.7	24.3	25
Bupa	39	37.8	37.8	38.2
Thyroid	8.5	4.1	4.1	6.7
Glass	35	34.9	33.3	27.3
Sonar	11.2	5	4.1	23.3

Conclusions

In this paper, we proposed a simple and efficient method to construct an *accurate* FRBCS. We proposed a new method of rule-weight specification in order to tune the classifier. In this method, the number of generated rules at most equals the number of training patterns and thus, no process for rule subset selection is needed. This number does not change through the weighting process, since no rule is given the weight of zero. As the number of partitions of each feature increases, the generalization ability of the system competes and even precedes the weighted K-NN method. Moreover, the proposed scheme is a FRBCS and has the advantage of interpretability, especially for low-dimensional data sets.

We also proposed a mechanism to find the optimal rule weights, which is much more useful in case of noisy or highly overlapped data sets, in order to prevent from overfitting of the learned classifier.

We used seven data sets from UCI-ML repository to assess the performance of the learning scheme. Simulation results on thsese data sets showed that the method can be used to construct a rule-base with a good generalization ability. The effect of rule weights could be seen, clearly, through this experiment. We also showed that the proposed method is more effective in reducing the error rate of the classifier in comparison with C4.5 as a successful rule-based method.

References

[1] S. Abe, R. Thawonmas, A fuzzy classifier with ellipsoidal regions, IEEE Trans. on Fuzzy Systems 5 (3) (1997) 358-368.

[2] D. Nauck, R. Kruse, A neuro-fuzzy method to learn fuzzy classification rules from data, Fuzzy Sets and Systems 89 (3) (1997) 277-288.

[3] O. Cordon, M. J. del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule- based classification systems, International Journal of Approximate Reasoning 20 (1999) 21-45.

[4] Ishibuchi, H., Nakashima, T., and Morisawa, T.: Voting in Fuzzy Rule-Based Systems for Pattern Classification problems, *Fuzzy Sets and Systems* 103 (1999) 223-238.

[5] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proc. 20th International Conference on Very large Databases, 1994, pp. 487-499.

[6] H. Ishibuchi, T. Yamamoto, Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining, Fuzzy Sets and Systems 141 (1) (2004) 59-88.

[7] Elomaa, T. and J. Rousu (1999). “General and efficient multisplitting of numerical Attributes,” m*achine Learning* 36, 201-244.

[8] Arima, M., Hara, E.H., Katzberg, J.D., *"A fuzzy logic and rough sets controller for HVAC systems"*, Proceedings of the IEEE WESCANEX, New York, pp. 133–138, 1995.

[9] Cordón, O., Herrera, F., Peregrín, A., "Applicability of the fuzzy operators in the design of fuzzy logic controllers", Fuzzy Sets and Systems, Vol. 86, No. 1, pp. 15–41, 1997.

[10] Glorennec, P.Y., "Application of fuzzy control for building energy management. In: Building Simulation": International Building Performance Simulation Association 1. Sophia Antipolis, France, pp. 197–201, 1991.

[11] Bárdossy, A., Duckstein, L., *"Fuzzy rule-based modeling with applications to geophysical"*, biological and engineering systems, CRC Press,1995.

[12] Bezdek, J.C., Pal, S.K., "Fuzzy Models for Pattern Recognition, Methods that Search for Structures in Data", IEEE Press, Boca Raton, 1992.

[13] Chi, Z., Yan, H., Pham, T., "Fuzzy algorithms with applications to image processing and pattern recognition", World Scientific, New York, 1996.

[14] S. Abe, M. S. Lan, A method for fuzzy rules extraction directly from numerical data and its application to pattern classification, IEEE Trans. on Fuzzy Systems 3 (1995) 18-28.

[15] H. Ishibuchi, K. Nozaki, H. Tanaka, Distributed representation of fuzzy rules and its application to pattern classification, Fuzzy Sets and Systems 52 (1) (1992) 21-32.

[16] S. Mitra, L. I. Kuncheva, Improving classification performance using fuzzy MLP and two-level selective partitioning of the feature space, Fuzzy Sets and Systems 70 (1) (1995) 1-13.