Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 158 (2007) 583–591
FUZZY sets and systems
www.elsevier.com/locate/fss

A weighting function for improving fuzzy classification systems performance

Eghbal G. Mansoori*, Mansoor J. Zolghadri, Seraj D. Katebi

Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran

Received 2 July 2005; received in revised form 28 January 2006; accepted 6 October 2006
Available online 3 November 2006

Abstract

This paper considers the automatic design of fuzzy rule-based classification systems from labeled data. The classification accuracy and interpretability of generated rules are of major importance in fuzzy classification systems. We propose a weighting function for compatibility grade of patterns that improves the performance of fuzzy classification system without degrading the interpretability of fuzzy rules. Our approach divides the covering subspace of each fuzzy rule into two subdivisions based on a threshold. Any pattern with compatibility grade above this threshold should be classified truly so the weighting function enhances their association degree. For patterns below threshold, their compatibility grades remain unchanged. The splitting threshold for each rule (i.e. the compatibility grade of a specific pattern) is found using distribution of patterns in the covering subspace of that rule. We also show that how the proposed approach is applicable when fuzzy rules have certainty grades. Experiments on some well-known data sets are used to evaluate the performance of our approach.

Keywords: Fuzzy rule-based classification system; Fuzzy subspace; Weighting function

1. Introduction

Fuzzy rule-based expert systems are often applied to classification problems in various fields. The fuzzy if-then rules improve the interpretability of results and provide more insight into the classifier structure and decision making process [26]. Many approaches have been proposed for generating and learning fuzzy if-then rules from numerical data for classification problems. These include simple heuristic procedures [1,11], neuro-fuzzy techniques [18,19], clustering methods [2], fuzzy clustering in combination with fuzzy relations [24], fuzzy nearest neighbor [17], and genetic algorithms [25].

Traditionally, the design of fuzzy classification systems have focused either on the accuracy of the classifier or the interpretability of the fuzzy rules. Recently, some approaches that combine these properties have been reported. For example, deriving transparent models using fuzzy clustering [23], applying linguistic constraints to fuzzy modeling [26], rule extraction from neural networks [22], using genetic algorithm for iteratively developing fuzzy classifiers [21] and using evolutionary algorithms to learn the linguistic hedges and the parameters of t-norm connectives in fuzzy rules [5,7].

* Corresponding author. Tel./fax: +98 711 6271747.
E-mail address: mansoori@shirazu.ac.ir (E.G. Mansoori).

0165-0114/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2006.10.004

In this work, we do not attempt to modify the membership functions of the given linguistic values, as this will degrade the interpretability of the fuzzy rules. Instead, using weighting functions [6], we modify the compatibility grade of patterns to improve the classification accuracy, especially for very coarse partitioning where the performance is poor. Our approach specifies a positive pattern (i.e. pattern with true class) from the covering subspace of each fuzzy rule as splitting pattern and uses its compatibility grade as threshold. This pattern divides the covering subspace of each rule into two distinct subdivisions. All patterns having compatibility grade above this threshold are positive so any incoming pattern for this subdivision should be classified as positive. The proposed weighting function enhances the certainty grade of these patterns such that, using winner takes all reasoning method, they are classified by this rule. For other patterns, the weighting function makes no change.

Some approaches use certainty grades to improve the performance of fuzzy classification systems [10,20]. Adjustment of certainty grades is easier than the learning of membership functions of antecedent fuzzy sets. Also, the classification accuracy can be improved without modifying the membership function of each linguistic label. Considering these facts, our proposed weighting function is applicable for fuzzy rules having certainty grades. We obtain a general form of the weighting function that can also be used for this case.

The rest of this paper is organized as follows. In Section 2, designing a fuzzy rule-based classification system with reduced rule set is explained. Section 3 discusses the proposed weighting function. The effect of certainty grade on weighting function is explained in Section 4. In Section 5, we present the experimental results. Section 6 concludes this paper.

2. Designing fuzzy rule-based classification system

Fuzzy if-then rules for a pattern classification problem with n attributes can be written as

$$\text{Rule } R_j : \text{ If } x_1 \text{ is } A_{j1} \text{ and } \dots \text{ and } x_n \text{ is } A_{jn} \text{ then class } C_j, \text{ for } j = 1, 2, \dots, N, \quad (1)$$

where $X = [x_1, x_2, \dots, x_n]$ is the n-dimensional pattern vector, A_{ji} $(i = 1, 2, \dots, n)$ is an antecedent linguistic value such as *Small* or *Large*, C_j is the consequent class, and N is the number of fuzzy rules. Generally, for an M-class problem, m labeled patterns $X_p = [x_{p1}, x_{p2}, \dots, x_{pn}]$, $p = 1, 2, \dots, m$ is given. Usually, each attribute is first normalized to unit interval [0, 1]. Using the information provided by labeled patterns, the task of classifier design is to generate a set of fuzzy rules in the form of (1).

For this purpose, first the pattern space is partitioned into fuzzy subspaces and then, each partition is identified by a fuzzy rule if there are some patterns in that subspace [11]. In this paper we assume that partitioning of pattern space is provided in advance. We use triangular membership functions to partition each feature axis into K fuzzy subsets $\{A_1, A_2, \dots, A_k\}$. Fig. 1 shows this partitioning for $K = 2, 3, 4$ and 5.

Given an input partitioning of pattern space, one approach is to consider all possible combination of antecedent linguistic values and generate a fuzzy rule for each combination if there is a training pattern covered by this rule.

For high dimensional problems, this approach can generate too many rules which are practically impossible to handle. For example, for Wine data set [4] with 13 input variables, even with coarsest partitioning of pattern space $(K = 2)$, 2^{13} fuzzy rules may be generated with this method. One approach for handling this problem is to use a fuzzy rule evaluation measure to select a small subset of candidate rules [14].

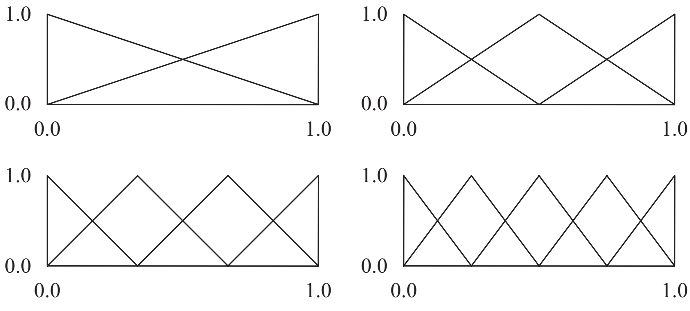

Fig. 1. Different partitioning of each feature axis.

The approach we take in this paper to handle high dimensional problems is to generate a fuzzy rule only if there is a training pattern in the decision subspace of the rule (i.e. classified by the rule). Using this method, for a problem with m training pattern, at most m rules can be generated. This is the case if each training pattern is located in the decision subspace of a different rule. In general, a single training pattern can generate 2^n fuzzy rules for an n-dimensional problem. Our rule generation method can be viewed as selecting one out of these 2^n rules (i.e. the rule having most compatibility with the training pattern).

The consequent class C_j of each fuzzy rule in Eq. (1) is determined by training patterns in the corresponding fuzzy subspace. The compatibility grade of each training pattern X_p is defined with the antecedent part $A_j = A_{j1} \times A_{j2} \times \cdots \times A_{jn}$ of the fuzzy rule R_j using the product operator as

$$\mu_j(X_p) = \mu_{j1}(x_{p1})\mu_{j2}(x_{p2}) \ldots \mu_{jn}(x_{pn}), \tag{2}$$

where $\mu_{ji}(\cdot)$ is the membership function of the antecedent fuzzy set A_{ji}. In order to select the consequent class of a fuzzy rule, we use the heuristic method proposed by Ishibuchi et al. [12], which is based on the confidence of association rules from the field of data mining. A fuzzy classification rule in (1) can be viewed as an association rule of the form $A_j \Rightarrow \mathit{class}\, C_j$, where A_j is a multidimensional fuzzy set representing the antecedent combination of the rule and C_j is a class label. In [13], a measure for evaluating the confidence of a fuzzy association rule is provided as

$$\mathit{Conf}(A_j \Rightarrow \mathit{Class}\, C_j) = \sum_{X_p \in \mathit{class}\, C_j} \mu_j(X_p) \Bigg/ \sum_{p=1}^{m} \mu_j(X_p)\ . \tag{3}$$

The consequent class C_j of fuzzy rule R_j is specified by identifying the class with the maximum confidence.

The most popular fuzzy reasoning method in fuzzy rule-based classification systems is the reasoning based on a single winner rule [9]. This method is simple and intuitive for human users. Other fuzzy reasoning methods are studied in [3,6,9]. In this way, a new pattern $X_p = [x_{p1}, x_{p2}, \ldots, x_{pn}]$ is classified by the single winner rule $R_{\hat{j}}$ defined as

$$\mu_{\hat{j}}(X_p) = \max\{\mu_j(X_p),\ j = 1, 2, \ldots, N\}, \tag{4}$$

where $\mu_j(X_p)$ is the compatibility grade of fuzzy rule R_j with X_p using (2).

3. The proposed weighting function

Cordon et al. [6] present a general model of fuzzy reasoning to combine information provided by different rules. Their model is an extension of the fuzzy classifier definition presented by Kuncheva [15]. In their model, after calculating the compatibility grade of patterns according to each rule, a weighting function is applied. This function modifies the association degrees in order to increase the classification accuracy of the system.

In this work, we define a piecewise linear weighting function based on a splitting threshold. This function enhances the compatibility grade of patterns above this threshold such that they are classified truly. For patterns below this threshold, their compatibility grades remain unchanged.

Considering the distribution of training patterns in the covering subspace of each fuzzy rule, we specify a splitting pattern which divides the covering subspace of the rule into two subdivisions. Given this pattern, its compatibility grade is used as splitting threshold. All training patterns having association values above this threshold are positive (i.e. patterns with true class). Intuitively, any incoming pattern within this subdivision must be classified truly. To classify these patterns as positive, their compatibility grade is set to one, ensuring that these patterns will be classified by the current rule. However, for some new patterns, the splitting threshold of more than one fuzzy rule might be over passed. In this case, the compatibility grade of new pattern in two or more fuzzy rules would be set to one. Selecting the appropriate rule from these competing rules is impossible. To overcome this difficulty, we add to one the initial compatibility grade of the pattern in each rule. Fig. 2 shows the input–output mapping of weighting function. In this Figure, t_j is the splitting threshold of rule R_j, μ_j^{in} is the compatibility grade of input pattern X_p with this rule, and μ_j^{out} is the enhanced compatibility grade.

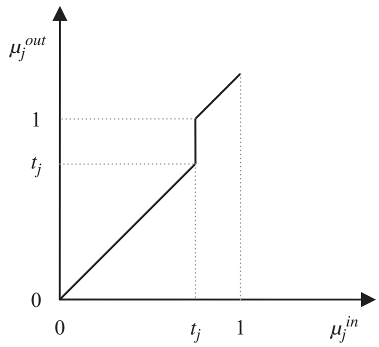

Fig. 2. Input–output mapping of weighting function.

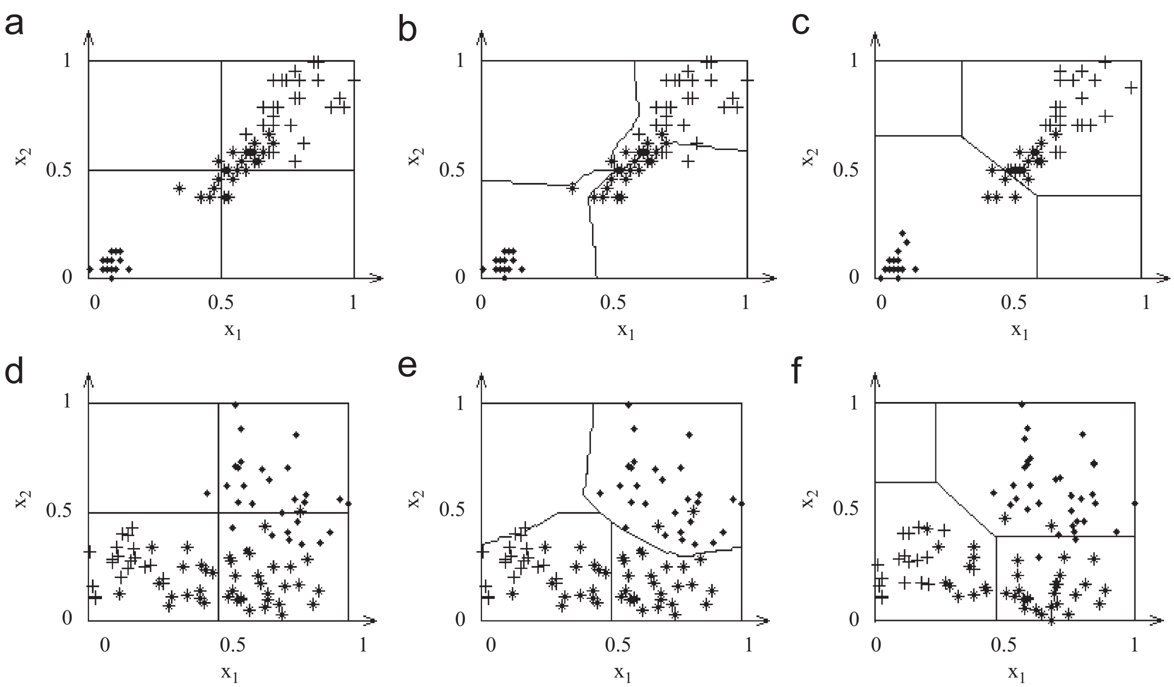

Fig. 3. Comparing the decision area of fuzzy rules; (a), (d) no weighting function and certainty grade; (b), (e) using weighting function; (c), (f) using certainty grade.

Thus, before selecting the winner rule for new pattern X_p, its compatibility grade with rule R_j is modified as

$$\mu_j^{\text{out}}(X_p) = \begin{cases} \mu_j^{\text{in}}(X_p) & \text{if } \mu_j^{\text{in}}(X_p) < t_j, \\ 1 - t_j + \mu_j^{\text{in}}(X_p) & \text{if } \mu_j^{\text{in}}(X_p) \geqslant t_j. \end{cases} \quad (5)$$

Specifying the splitting pattern is simple and straightforward. We rank (in descending order) the training patterns in the covering subspace of the rule based on their compatibility grade. The last positive pattern before the first negative one is selected as splitting pattern and its grade of compatibility is used as threshold.

Fig. 3 illustrates the effect of weighting function on decision area of fuzzy rules applied to two 2-dimensional pattern classification examples. Using no weighting function and certainty grade, the fuzzy rules have rectangular decision areas and the classification boundaries are always linear and parallel to the axes of the pattern space [16]. As claimed in [10], fuzzy rules with certainty grades have decision areas of various shapes and the classification boundaries are not always parallel to the axes of the pattern space. On the other hand, when weighting function is applied, the decision area of rules is not necessarily rectangular and the classification boundaries are nonlinear.

4. Effect of certainty grade on weighting function

As shown in [10], fuzzy rule-based systems can generate various classification boundaries by adjusting the certainty grade (i.e. rule weight) of each fuzzy rule even when fixed membership functions are used. Using certainty grade, each fuzzy if-then rule in (1) can be written as

$$\text{Rule } R_j: \text{ If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{and } x_n \text{ is } A_{jn} \text{ then class } C_j \text{ with } w_j, \text{ for } j = 1, 2, \ldots, N, \tag{6}$$

where w_j is the certainty grade of rule R_j. In this case, the winner rule $R_{\hat{j}}$ for a new pattern $X_p = [x_{p1}, x_{p2}, \ldots, x_{pn}]$ is defined by

$$\mu_{\hat{j}}(X_p).w_{\hat{j}} = \max\{\mu_j(X_p).w_j, \quad j = 1, 2, \ldots, N\}. \tag{7}$$

In order to assign a suitable certainty grade to each fuzzy rule, several heuristic measures are proposed in the literature [13]. Since we use the single winner-based method, the following definition of certainty grade from Ishibuchi and Yamamoto [13] is appropriate for multi-class problems:

$$w_j = Conf(A_j \Rightarrow Class\, C_j) - Conf(A_j \Rightarrow Class\, C_{2\text{nd}}), \tag{8}$$

where class $C_{2\text{nd}}$ is the class with the second largest confidence for the antecedent part A_j. This definition has been used in many fuzzy rule-based classification systems [8] where satisfactory results have been obtained.

Using certainty grade for fuzzy rules in classification process, some modification on weighting function in (5) is needed. We obtain a general form of weighting function that can also be used for fuzzy classification rules having certainty grades. Assigning certainty grade w_j to rule R_j, changes the slope of piecewise linear weighting function in Fig. 2 from 1 to w_j. This is equal to rotating the input–output mapping about the origin up to $\alpha_j = \tan^{-1} w_j - (\pi/4)$ radians as shown in Fig. 4.

Generally, rotating any point (x, y) about the origin through an angle θ radians transforms this point to (x', y') using Eq. (9):

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix}. \tag{9}$$

From $\alpha_j = \tan^{-1} w_j - (\pi/4)$, we obtain

$$\begin{cases} \sin\alpha_j = (w_j - 1)/\sqrt{2w_j^2 + 2}, \\ \cos\alpha_j = (w_j + 1)/\sqrt{2w_j^2 + 2}. \end{cases} \tag{10}$$

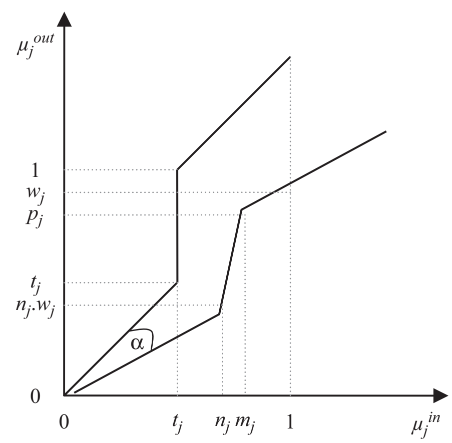

Fig. 4. Effect of certainty grade on weighting function.

By transforming point (t_j, t_j) to $(n_j, n_j.w_j)$ in Fig. 4, using Eq. (9) for $\theta = \alpha_j$, the unknown parameter n_j is obtained as

$$\begin{bmatrix} n_j \\ n_j.w_j \end{bmatrix} = \begin{bmatrix} \cos\alpha_j & -\sin\alpha_j \\ \sin\alpha_j & \cos\alpha_j \end{bmatrix} \times \begin{bmatrix} t_j \\ t_j \end{bmatrix} \Rightarrow n_j = t_j\sqrt{\frac{2}{1+w_j^2}}. \tag{11}$$

Similarly, transformation of point $(t_j, 1)$ to (m_j, p_j) yields

$$\begin{bmatrix} m_j \\ p_j \end{bmatrix} = \begin{bmatrix} \cos\alpha_j & -\sin\alpha_j \\ \sin\alpha_j & \cos\alpha_j \end{bmatrix} \times \begin{bmatrix} t_j \\ 1 \end{bmatrix} \Rightarrow \begin{cases} m_j = \{t_j.(w_j+1) - (w_j-1)\}/\sqrt{2w_j^2+2}, \\ p_j = \{t_j.(w_j-1) + (w_j+1)\}/\sqrt{2w_j^2+2}. \end{cases} \tag{12}$$

So, using the certainty grade w_j, the splitting threshold for rule R_j moves from t_j to $\sqrt{2/(1+w_j^2)}t_j$. The weighting function for R_j that modifies the compatibility grade of X_p before determining the single winner rule, changes to

$$\mu_j^{\text{out}}(X_p) = \begin{cases} w_j.\mu_j^{\text{in}}(X_p) & \text{if } \mu_j^{\text{in}}(X_p) < n_j, \\ \left(\dfrac{p_j - n_j.w_j}{m_j - n_j}\right).\mu_j^{\text{in}}(X_p) - \left(\dfrac{p_j - m_j.w_j}{m_j - n_j}\right).n_j & \text{if } n_j \leqslant \mu_j^{in}(X_p) < m_j, \\ w_j.\mu_j^{\text{in}}(X_p) - w_j.m_j + p_j & \text{if } \mu_j^{\text{in}}(X_p) \geqslant m_j. \end{cases} \tag{13}$$

5. Experimental results

In this section, the performance of fuzzy rule-based classification systems using weighting functions is examined. We used five data sets available from the UCI ML repository [4]. Table 1 shows their specifications.

This paper uses three evaluation methods to examine the classification accuracy of systems at various levels of training: 2FT, LV1 and 2CV [27]. In the 2FT (i.e. full train and full test) method, the entire data used for training the system, is also used for testing. The 2CV (i.e. two-fold cross-validation) method divides the data into two subsets of the same size. One subset is used as training data for generating fuzzy rules and their parameters. The other subset is used as test data for evaluating the system. The same training and testing procedure is also performed after exchanging the role of each subset. Since the error rate on test data in the 2CV depends on the initial division of the data, the 2CV is iterated 10 times using different divisions of the data set. The average classification rate in the 2CV can be viewed as indicating the generalization ability when the size of training data is small [10]. In LV1 (i.e. leaving one out) approach, one sample is used in test phase and the rest of samples are used in training phase. The procedure is repeated until all the samples are used in the test phase. The average classification rate on test data is reported as the performance of classifier.

Since the feature space of data sets is continuous, each attribute value is normalized to a real number in the unit interval [0, 1]. Then, the attribute axes are homogeneously divided by K triangular fuzzy sets. These fuzzy partitions are used for generating fuzzy rules, specifying their splitting thresholds and calculating their certainty grades and other parameters used in (13).

Tables 2–4 summarize the simulation results for five data sets. Each table compares the classification accuracy of fuzzy systems for two cases: (a) fuzzy rules do not use weighting functions (i.e. columns titled *old* in tables)

Table 1
Statistics of data sets used for system evaluation

Data set	Number of attributes	Number of classes	Number of samples
Iris	4	3	150
Wine	13	3	178
Glass	9	6	214
Image segmentation	18	7	210
Breast cancer	9	2	684

Table 2
Classification accuracy when using weighting functions (2FT method)

Data set	K	Rules without certainty grades		Rules having certainty grades		Number of rules
		Old	New	Old	New	
Iris	2	71.33	76.00	68.00	69.33	9
	3	92.00	96.00	94.00	95.33	15
	4	84.00	93.33	92.00	93.33	28
	5	94.67	98.00	96.67	98.00	48
Wine	2	92.13	98.31	93.82	97.75	124
	3	96.07	100	98.31	100	138
	4	100	100	100	100	171
	5	100	100	100	100	177
Glass	2	53.27	62.15	63.55	64.02	33
	3	67.76	69.63	69.16	69.63	44
	4	64.02	73.83	71.96	75.23	65
	5	76.64	80.37	76.64	78.5	95
Image	2	60.95	71.90	66.19	70.95	47
	3	86.67	92.86	90.00	91.43	101
	4	94.76	98.57	91.90	96.19	150
	5	95.24	98.10	95.71	97.62	171
Cancer	2	95.32	98.83	95.32	98.10	135
	3	98.25	98.68	97.95	98.39	254
	4	99.42	99.85	99.56	99.71	313
	5	100	100	100	100	326

Table 3
Classification accuracy when using weighting functions (2CV method)

Data set	K	Rules without certainty grades		Rules having certainty grades		Rejection rate	Average number of rules
		Old	New	Old	New		
Iris	2	71.27	77.87	70.00	72.13	8.40	9.0
	3	91.67	93.87	93.27	91.33	2.70	15.0
	4	80.40	91.53	89.33	92.00	0.00	27.5
	5	93.93	93.80	95.47	95.47	2.00	47.1
Wine	2	89.27	91.46	92.30	93.93	0.23	124.0
	3	93.26	95.22	95.45	95.56	0.45	137.0
	4	89.94	90.00	90.06	90.84	4.49	161.6
	5	79.72	79.44	79.72	79.66	17.02	143.2
Glass	2	48.83	55.47	54.07	53.32	0.14	33.0
	3	58.74	59.25	60.93	61.22	1.40	42.3
	4	53.64	62.38	59.77	61.07	2.85	58.1
	5	59.11	60.65	60.05	58.97	5.70	84.7
Image	2	55.52	66.38	57.10	60.10	1.05	46.8
	3	73.43	78.00	77.57	80.14	3.67	95.1
	4	79.10	79.38	78.29	78.05	6.76	139.6
	5	76.05	76.52	76.48	76.38	10.52	153.8
Cancer	2	95.06	95.96	95.15	96.64	0.00	135.0
	3	93.58	93.63	93.73	94.01	2.60	234.6
	4	75.98	75.99	76.07	76.20	21.61	216.0
	5	68.83	68.83	68.85	68.86	28.60	210.0

Table 4
Classification accuracy when using weighting functions (LV1 method)

Data set	K	Rules without certainty grades		Rules having certainty grades		Rejection rate	Average number of rules
		Old	New	Old	New		
Iris	2	71.33	76.00	68.00	68.00	8.67	9.0
	3	92.00	92.00	93.33	92.67	2.67	15.0
	4	82.67	90.67	88.00	90.67	0.00	28.0
	5	94.67	94.00	95.33	96.67	2.00	48.0
Wine	2	88.76	91.57	91.57	94.94	0.00	124.0
	3	94.94	98.31	97.19	96.63	0.00	138.0
	4	91.57	91.57	91.57	93.26	2.81	171.0
	5	84.27	83.71	84.27	84.27	11.24	176.8
Glass	2	51.87	58.88	56.54	56.54	0.00	33.0
	3	63.08	62.62	62.62	63.08	0.00	44.0
	4	53.27	62.15	59.35	61.22	2.34	65.0
	5	62.62	64.02	61.68	62.62	4.67	95.0
Image	2	56.67	66.19	60.00	61.90	0.95	47.0
	3	78.10	81.43	81.90	83.81	2.38	101.0
	4	81.43	81.90	80.48	81.43	4.76	150.0
	5	79.52	80.48	78.57	79.52	8.57	170.9
Cancer	2	95.03	96.35	95.18	96.78	0.00	135.0
	3	95.61	95.61	95.61	96.20	1.02	254.0
	4	78.95	79.09	78.94	79.37	18.13	312.7
	5	70.76	70.76	70.76	70.91	25.73	325.7

and (b) fuzzy rules using weighting functions (i.e. columns titled *new* in tables). When fuzzy rules have certainty grades, their performance is also included. As seen, classifier performance has improved in majority of cases for different evaluation methods considered, especially for coarse fuzzy partitioning (i.e. $K = 2, 3$). This is true, because for coarse partitioning, the covering subspace of each fuzzy rule is almost large such that there are some patterns in each subdivision specified by splitting threshold. However, when fuzzy rules utilize certainty grades, the amount of improvement is moderate because the classification accuracy is almost high and there is a little room for improvement.

Generally, the classification of a test pattern may be rejected due to lack of fuzzy rule compatible with that pattern, especially when generating only a small subset of rules. As mentioned before, we generate fuzzy rule only if there are some training patterns in the decision subspace of the rule. Intuitively, for fine fuzzy partitioning (i.e. $K = 4, 5$), the covering subspace of generated rules decrease considerably, so the probability of rejection of test patterns increases as shown in Tables 3 and 4.

6. Conclusion

In this paper, we examined the performance of fuzzy if-then rules extracted from numerical data for pattern classification problems using weighting function. This function divides the covering subspace of each fuzzy rule according to a threshold and then enhances the compatibility grade of some patterns to classify them truly. Experimental results showed that the proposed scheme can improve the classification accuracy considerably, especially when the size of covering subspace is large. Additionally, when fuzzy rules have certainty grades, the performance of new approach showed noticeable improvements.

The rule generation method used in this paper, generates rules having at least one training pattern in their decision subspace. It makes sense when generating complete fuzzy rules or selecting only good rules based on an evaluation metric, using weighting function will hopefully improve the performance of fuzzy classification systems.

References

[1] S. Abe, M.S. Lan, A method for fuzzy rules extraction directly from numerical data and its application to pattern classification, IEEE Trans. Fuzzy Systems 3 (1) (1995) 18–28.
[2] S. Abe, R. Thawonmas, A fuzzy classifier with ellipsoidal regions, IEEE Trans. Fuzzy Systems 5 (3) (1997) 358–368.
[3] J.C. Bezdek, Pattern analysis, in: E.H. Ruspini, P. Bonissone, W. Pedrycz (Eds.), Handbook of Distributed Representation of Fuzzy Rules and its Application to Pattern Classification, Handbook of Fuzzy Computation, Chapter F6, Institute of Physics Publishing, London, 1998.
[4] C.L. Blake, C.J. Merz, UCI Repository of machine learning databases, University of California, Department of Information and Computer Science, Irvine, CA, 1998.
[5] O. Cordon, M.J. del Jesus, F. Herrera, Genetic learning of fuzzy rule-based classification systems cooperating with fuzzy reasoning methods, Internat. J. Intelligent Systems 13 (1998) 1025–1053.
[6] O. Cordon, M.J. del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, Internat. J. Approx. Reason. 20 (1999) 21–45.
[7] F. Herrera, F.A. Marquez, A. Peregrin, Genetic adaptation of rule connectives and conjunction operators in fuzzy rule based systems: an experimental comparative study, Proc. Third Internat. Conf. of the European Society for Fuzzy Logic and Technology, 2003, pp. 100–104.
[8] H. Ishibuchi, T. Nakashima, Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes, IEEE Trans. Industrial Electronics 46 (6) (1999) 157–168.
[9] H. Ishibuchi, T. Nakashima, T. Morisawa, Voting in fuzzy rule-based systems for pattern classification problems, Fuzzy Sets and Systems 103 (2) (1999) 223–238.
[10] H. Ishibuchi, T. Nakashima, Effect of rule weights in fuzzy rule-based classification systems, IEEE Trans. Fuzzy Systems 9 (4) (2001) 506–515.
[11] 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.
[12] H. Ishibuchi, T. Yamamoto, T. Nakashima, Fuzzy data mining: Effect of fuzzy discretization, Proc. First IEEE Internat. Conf. on Data Mining, 2001, pp. 241–248.
[13] H. Ishibuchi, T. Yamamoto, Comparison of heuristic rule weight specification method, Proc. 11th IEEE Internat. Conf. on Fuzzy Systems, 2002, pp. 908–913.
[14] H. Ishibuchi, T. Yamamoto, Comparison of heuristic criteria for fuzzy rule selection in classification problems, Fuzzy Optim. Decis. Making 3 (2) (2004) 119–139.
[15] L.I. Kuncheva, On the equivalence between fuzzy and statistical classifiers, Internat. J. Uncertainty, Fuzziness and Knowledge-Based Systems 4 (3) (1996) 245–253.
[16] L.I. Kuncheva, Fuzzy Classifier Design, Physica-Verlag, Heidelberg, 2000.
[17] L.I. Kuncheva, J.C. Bezdek, A fuzzy generalized nearest prototype classifier, Proc. Seventh IFSA World Congress, vol. 3(1), 1997, pp. 217–222.
[18] S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, IEEE Trans. Neural Networks 11 (3) (2000) 748–768.
[19] D. Nauck, R. Kruse, A neuro-fuzzy method to learn fuzzy classification rules from data, Fuzzy Sets and Systems 89 (3) (1997) 277–288.
[20] D. Nauck, R. Kruse, How the learning of rule weights affects the interpretability of fuzzy systems, Proc. Seventh IEEE Internat. Conf. Fuzzy Systems, 1998, pp. 1235–1240.
[21] J.A. Roubos, M. Setnes, J. Abonyi, Learning fuzzy classification rules from labeled data, IEEE Trans. Fuzzy Systems 8 (5) (2001) 509–522.
[22] D. Setiono, Generating concise and accurate classification rules for breast cancer diagnosis, Artificial Intelligence Med. 18 (1999) 205–219.
[23] M. Setnes, R. Babuska, Rule-based modeling: precision and transparency, IEEE Trans. Systems Man Cybernet. Part C: Appl. Rev. 28 (1) (1998) 165–169.
[24] M. Setnes, R. Babuska, Fuzzy relational classifier trained by fuzzy clustering, IEEE Trans. Systems Man and Cybernet. Part B: Cybernetics 29 (1) (1999) 619–625.
[25] M. Setnes, J.A. Roubos, GA-fuzzy modeling and classification: complexity and performance, IEEE Trans. Fuzzy Systems 8 (5) (2000) 509–522.
[26] J. Valente de Oliveira, Semantic constraints for membership function optimization, IEEE Trans. Fuzzy Systems 19 (1) (1999) 128–138.
[27] S.M. Weiss, C.A. Kulikowski, Computer Systems that Learn, Morgan Kaufmann, San Mateo, 1991.