USING DISTRIBUTION OF DATA TO ENHANCE PERFORMANCE OF FUZZY CLASSIFICATION SYSTEMS

E. G. MANSOORI, M. J. ZOLGHADRI AND S. D. KATEBI

ABSTRACT. This paper considers the automatic design of fuzzy rule-based classification systems based on labeled data. The classification performance and interpretability are of major importance in these systems. In this paper, we utilize the distribution of training patterns in decision subspace of each fuzzy rule to improve its initially assigned certainty grade (i.e. rule weight). Our approach uses a punishment algorithm to reduce the decision subspace of a rule by reducing its weight, such that its performance is enhanced. Obviously, this reduction will cause the decision subspace of adjacent overlapping rules to be increased and consequently rewarding these rules. The results of computer simulations on some well-known data sets show the effectiveness of our approach.

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 [23]. This paper investigates the design of fuzzy rule-based classifiers from labeled data. Many approaches have been proposed for generating and learning of fuzzy if-then rules from numerical data. These include simple heuristic procedures [8, 1], neuro-fuzzy techniques [16, 14], clustering methods [2], fuzzy clustering in combination with other methods such as fuzzy relations [21], fuzzy nearest neighbor [13], and genetic algorithms [22].

Conventional algorithms have focused either on accuracy or interpretability. Recently some approaches to combine these properties have been reported [20, 19, 18]. Compact fuzzy rule-based classifiers can be designed without adjusting membership functions by assigning a weight to each rule [15]. Modifying the membership functions of antecedent linguistic values will change the associated semantics and degrade the interpretability of fuzzy rules. Learning weights can equivalently be replaced by modification of antecedent linguistic values [6].

This paper uses rule weighting to enhance the performance of resulting classifiers. Adjusting fuzzy rule weights is easier than learning antecedent fuzzy sets. Also, classification performance can be improved without modifying the membership function of each linguistic label. In the proposed approach, the distribution of training patterns in the decision subspace of each rule is used to modify its weight. This punishment approach reduces the weight of a rule to shrink its decision subspace, thus discarding the misclassified patterns from the decision subspace and hence increasing the precision of the rule. The decision subspace of the adjacent overlapping rules will grow to include the discarded patterns.

The rest of this paper is organized as follows. Section 2 explains the general method for generating fuzzy classification rules. Section 3 discusses the effect of rule weight on classification results in fuzzy rules. Section 4 describes our method for adjusting rule weights and section 5 examines the proposed approach by means of computer simulations on well-known data sets. Section 6 concludes the paper.

2. **Generating Fuzzy Classification Rules**

Fuzzy if-then rules for an *M*-class pattern classification problem with *n* attributes can be written as:

Rule R_j: If x_1 is A_{j1} and … and x_n is A_{jn} then class C_j with CF_j, for $j=1,2,…,N$ (1)

where $X=[x_1,x_2,…,x_n]$ is the *n*-dimensional pattern vector, A_{ji} ($i=1,2,…,n$) is an antecedent linguistic value such as *Small* or *Large*, C_j is the consequent class, CF_j is the certainty grade (i.e. rule weight) of the rule, and *N* is the number of fuzzy rules.

In general, *m* labeled patterns X_p , $p=1,2,…,m$ are given for an *M*-class problem. Usually, each attribute is first normalized to the unit interval [0,1]. The task of classifier design is to generate a set of fuzzy rules in the form of (1) using the information provided by labeled patterns. Assuming that linguistic values assigned to each input feature is given and fixed, the task is then to assign a consequent class to each fuzzy subspace.

Two examples of such pattern classification problems are provided in Figure 1 (see section 5 for Iris2f and Wine2f). Each example is treated as a classification problem with two features (*n*=2) and three classes (*M*=3). Figure 1 shows Iris2f and Wine2f with 75 and 89 labeled patterns, respectively.
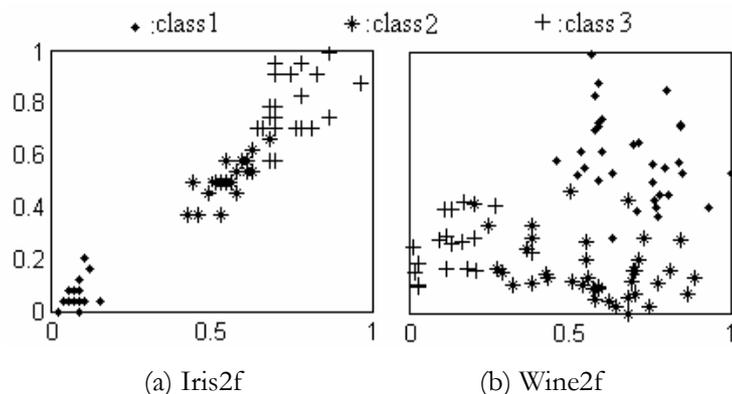
FIGURE 1. Two examples of pattern classification problems

Two main heuristics to generate fuzzy classification rules from labeled data are the methods proposed by Wang and Mendel [25] and Ishibuchi [8]. In the method proposed by Ishibuchi, 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. In fact, if there are K antecedent fuzzy sets on each axis of an n-dimensional pattern space, it will be partitioned into K^n fuzzy subspaces. Now, if a fuzzy rule is generated for each subspace, the number of rules will at most equal the number of subspaces. Figure 2 illustrates the grid-type fuzzy partitioning of the two-dimensional pattern space $[0,1]^2$ for the 3-class problems in Figure 1, where the pattern space are partitioned into nine fuzzy subspaces by three triangular fuzzy sets (*Small*, *Medium*, and *Large*) on each axis. Figure 2(a) only shows the fuzzy partition for the rule "If x_1 is *Small* and x_2 is *Large*", while the fuzzy partition of Figure 2(b) belongs to the rule "If x_1 is *Medium* and x_2 is *Large*".
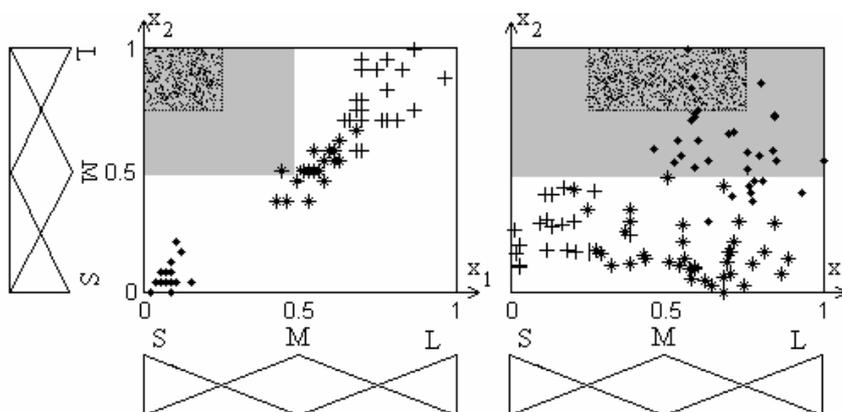

FIGURE 2. Covering and decision subspaces for fuzzy rules

Two main subspaces can be identified by each fuzzy rule: the covering subspace of the rule (gray regions in Figure 2) and its decision subspace (spotted gray regions in Figure 2). While any pattern in covering subspace will cause the rule to be fired, all patterns in decision subspace will be classified by this rule. It is obvious that the covering subspace of a rule has some overlap with the adjacent rules, so any pattern in this subspace will also fire adjacent rules.

The approach we take to handling higher 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 this rule). Using this method, at most *m* rules can be generated for a problem with *m* training patterns. This is the case if each training pattern is located in the decision subspace of a different rule.

The consequent class C_j of each rule in Eq. (1) is determined by training patterns in the corresponding covering subspace. The compatibility grade of each training pattern X_p is defined with the antecedent part of the fuzzy rule R_j as:

$$\mu_j(X_p) = \prod_{i=1}^{n} \mu_{ji}(x_{pi}) \tag{2}$$

where $\mu_{ji}(.)$ is the membership function of the antecedent fuzzy set A_{ji}. The confidence of the fuzzy association rule "$A_j \Rightarrow class\ T$" is defined as [11]:

$$Conf\ (A_j \Rightarrow class\ T) = \frac{\sum\limits_{X_p \in class\ T} \mu_j(X_p)}{\sum\limits_{p=1}^{m} \mu_j(X_p)} \tag{3}$$

The consequent class C_j of fuzzy rule R_j is specified by identifying the class with the maximum confidence. That is, the consequent class C_j is chosen so that the following relation holds:

$$Conf\ (A_j \Rightarrow class\ C_j) = \max\{Conf\ (A_j \Rightarrow class\ T):T = 1,2,...,M\} \tag{4}$$

The most popular fuzzy reasoning method in fuzzy rule-based classification systems is the reasoning based on a single winner rule [7]. This method is simple and intuitive for human users. Other fuzzy reasoning methods are studied in [7,3]. When fuzzy rules have no certainty grades, a new pattern $X_p = [x_{p1}, x_{p2}, ..., x_{pn}]$ is classified by the single winner rule R_w defined by:

$$\mu_w(X_p) = \max\{\mu_j(X_p):j = 1,2,...,N\} \tag{5}$$

where $\mu_j(X_p)$ is the compatibility grade of rule R_j with X_p, and can be calculated using (2). Figure 3 shows the fuzzy rules for the pattern classification problem of Figure 2 where a fuzzy rule is generated for each fuzzy partition and there is no missing rule.

Class 1: R1, R2, R4
Class 2: R3, R5, R7
Class 3: R6, R8, R9

Class 1: R3, R5, R6, R8, R9
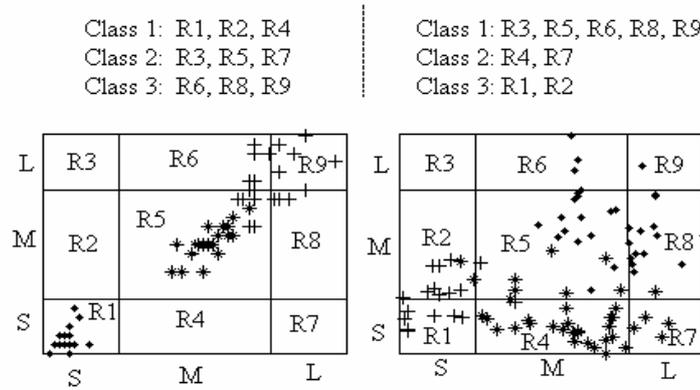Class 2: R4, R7
Class 3: R1, R2

FIGURE 3. Complete fuzzy rules and their decision area

Consider the case where the training patterns in the covering area of a fuzzy rule are distributed such that there is no pattern in its decision area (e.g. rules *R2*, *R3*, *R4*, and *R7* in Figure 3). As shown in Figure 4, such fuzzy rules are discarded in this paper.

Class 1: R1
Class 2: R5
Class 3: R6, R8, R9

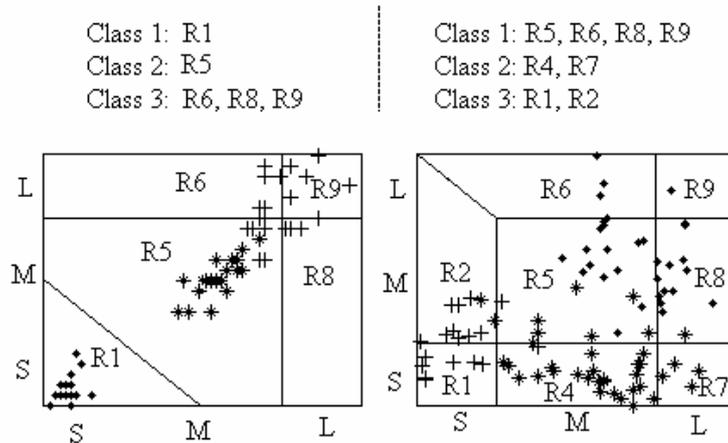Class 1: R5, R6, R8, R9
Class 2: R4, R7
Class 3: R1, R2

FIGURE 4. Reduced fuzzy rules and their decision area

In general, the classification of a test pattern may be rejected due to lack of a fuzzy rule compatible with the pattern, even when the rules are complete. In the reduced case, only a small subset of rules is selected, and hence the probability of rejection of test patterns increases.

As shown in Figure 3, fuzzy rules have rectangular or hyper-rectangular decision areas when no rule is missing [12]. The classification boundaries in this case are always parallel to the axes of the pattern space. On the other hand, when some rules

are missing, the decision area of each rule is not always rectangular (see Figure 4) and classification boundaries are not always parallel to the axes of the pattern space. Figure 5 shows the classification boundaries.
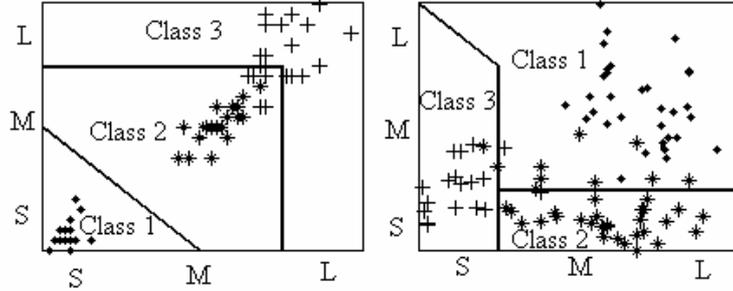

FIGURE 5. Classification boundaries of fuzzy rules

3. **Rule Weighting Measures**

As shown in [6], fuzzy rule-based systems can generate various classification boundaries by adjusting the weight of each rule even when fixed membership functions are used. In this case, the classification boundaries are not always parallel to the axes of the pattern space. Moreover, the decision subspace of each fuzzy rule is not always rectangular or hyper-rectangular. Using weight CF_j for rule R_j, a new pattern X_p will be classified by the winner rule R_w when:

$$\mu_w(X_p).CF_w = \max\{\mu_j(X_p).CF_j : j = 1,2,...,N\} \tag{6}$$

In this case, the size of decision subspace of each rule is determined by its certainty grade and the membership functions of its antecedent linguistic values. In other words, the decision subspace can be adjusted by modifying the certainty grade even if the membership functions are not changed.

There exist several methods for specifying the rule weight [10]. The choice of an appropriate specification depends on a fuzzy reasoning method used for pattern classification [24]. Using the single winner-based method, we select the definitions of rule weight as follows. The first definition has been used in some fuzzy classification systems [5,8].

$$CF_j^1 = Conf(A_j \Rightarrow class\, C_j) - \tfrac{1}{M-1} Conf(A_j \Rightarrow class\, \overline{C}_j) \tag{7}$$

where class C_j is the consequent class, and

$$Conf\,(A_j \Rightarrow class\,\overline{C}_j) = \sum_{\substack{T=1\\T\neq C_j}}^{M} Conf\,(A_j \Rightarrow class\,T) \tag{8}$$

In this definition, the *M*-class pattern classification problem is virtually handled as a two-class problem where the classification is between class C_j and a merged class including all the other classes (i.e., $\overline{C}_j = \{1,2,...,M\} - \{C_j\}$) [9]. The second definition of rule weight is appropriate for multi-class problems [10] and is defined as:

$$CF_j^2 = Conf(A_j \Rightarrow class\, C_j) - Conf(A_j \Rightarrow class\, \overline{C}_j) \qquad (9)$$

Obviously, for 2-class problems we obtain $CF_j^1 = CF_j^2$.

By applying definition (7) to the classification problems in Figure 4, the rule weights in Figure 6 are obtained. As shown in Figure 6, only 6 and 8 patterns are misclassified which is comparable with the non-weighted fuzzy rules in Figure 5 where the misclassified patterns were 8 and 11, respectively.
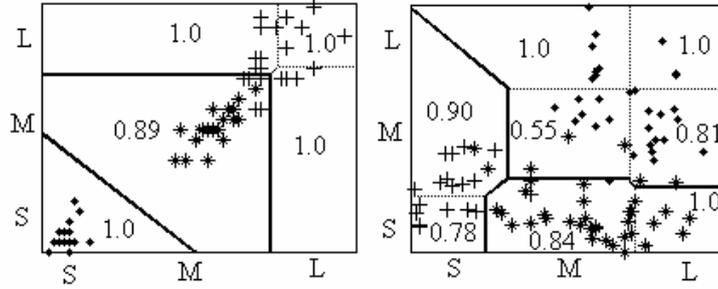

FIGURE 6. Decision area and classification boundaries of fuzzy rules having weights

4. **Improving the Weight of Fuzzy Rules**

Consider the case where the pattern space in Figure 1 is partitioned into four fuzzy subspaces by two antecedent fuzzy sets (i.e. *S* and *L*) on each axis as shown in Figures 7(a) and 7(b). In this case, four fuzzy rules are generated (i.e. there is no missing rule). Without using rule weight, the number of misclassified patterns in each case is 21. By calculating the rule weights using (7), the decision area of fuzzy rules shown in Figure 7(c) and 7(d) are obtained which illustrates that 23 and 15 are the misclassified patterns for Iris2f and Wine2f, respectively.

Comparing the number of misclassified patterns for the Iris2f example shows that either weighted rules are not very effective, or the assigned weights are not suitable. This drawback clarifies that selecting a suitable weight for each fuzzy rule depends on other parameters as well as its confidence.

In this paper, we consider the distribution of patterns in decision subspace of each rule to improve its weight. The following algorithm is proposed to improve the rule weighting mechanism.

Algorithm: Improves the initially assigned weights.

1. Specify K, the number of antecedent linguistic values on each feature.

2. Generate fuzzy rules from training data as proposed in section 2.

3. Calculate the certainty grade for each fuzzy rule as its weight using (7) or (9).

4. Apply the fuzzy rules to classify the training patterns. The decision subspace of each rule would be obtained.

5. Compute the confidence of each rule using only patterns in its decision subspace.

6. Use this confidence and the distribution of training patterns in decision subspace of each rule to modify its weight using (11).
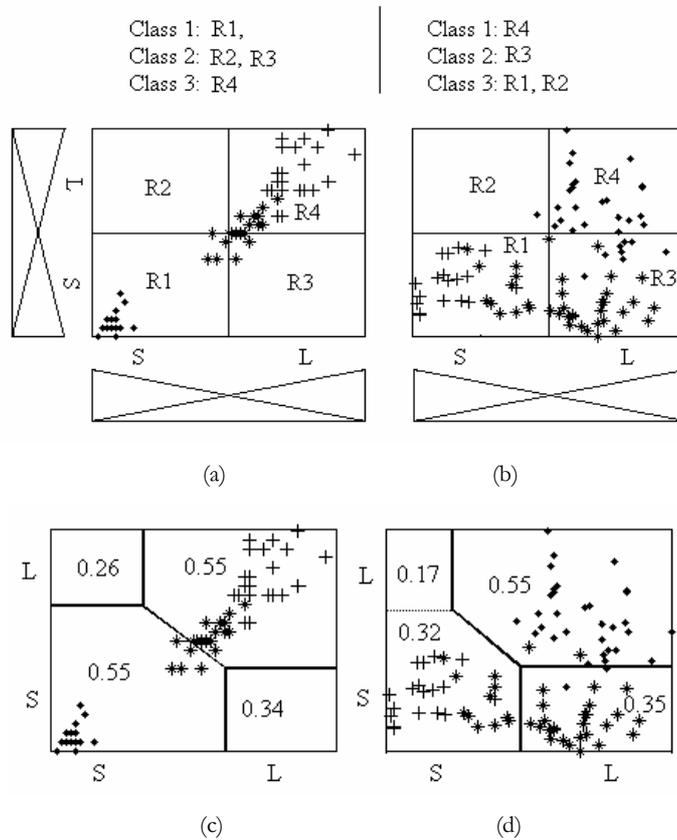

FIGURE 7. Decision area of four fuzzy rules

After applying the designed fuzzy system on training data, some useful information is obtained. For each training pattern, its classifying rule is known. Moreover, the confidence of each rule in its decision subspace can be calculated. This new confidence ($Conf_j$) is used to specify the effectiveness of rule in classification phase.

If the confidence of a rule is high, this means that the decision subspace is approximately accurate and the weight is suitable. However a low value of confidence confirms that the decision subspace of rule should be modified such that the misclassified patterns are discarded. A straightforward approach is to shrink the decision subspace of rule by reducing its weight.

The amount of reduction depends on several factors. One factor is the new confidence. The other effective parameter is the distribution of relevant (i.e. true positive) and irrelevant (i.e. false positive) patterns in its decision subspace. For this purpose, a new parameter denoted by MC_j in (10) is defined. This parameter is the compatibility grade of a relevant pattern in the decision subspace that can approximately distinguish relevant patterns from irrelevant ones.

The consequent selection scheme in (3) emphasizes that generally, the compatibility grade of relevant patterns in decision subspace of a rule is almost always higher than of irrelevant patterns. Let us specify μ_j^{lr} to denote the compatibility grade of the last relevant pattern (i.e. with the lowest compatibility grade) in the decision subspace for rule R_j, and μ_j^{fi} be the compatibility grade of the first irrelevant pattern (i.e. with the highest compatibility grade). If $\mu_j^{lr} \geq \mu_j^{fi}$ (e.g. rule R_1 in Figure 7(a)), this means that the relevant and irrelevant patterns in decision subspace of rule R_j are well separated, so for such cases μ_j^{lr} is used for MC_j. On the other hand, if $\mu_j^{lr} < \mu_j^{fi}$ (e.g. rules R_2 and R_3 in Figure 7(b)), then the patterns in decision subspace of this rule are scattered such that some irrelevant patterns are placed between relevant ones, or in the worse case, the patterns are randomly distributed in decision subspace. For this case, a relevant pattern with compatibility grade μ_j^{ir} is selected for MC_j, such that the compatibility grade summation of relevant patterns below MC_j is almost equal to those of irrelevant patterns above MC_j. Therefore,

$$MC_j = \begin{cases} \mu_j^{lr} & \text{if } \mu_j^{lr} \geq \mu_j^{fi} \\ \mu_j^{ir} & \text{if } \mu_j^{lr} < \mu_j^{fi} \text{ and } \sum_{\substack{X_p \notin class\,C_j \\ \mu_j(X_p) > \mu_j^{lr}}} \mu_j(X_p) \cong \sum_{\substack{X_p \in class\,C_j \\ \mu_j^{ir} > \mu_j(X_p) > \mu_j^{lr}}} \mu_j(X_p) \end{cases} \qquad (10)$$

The parameter MC_j illustrates the concentration degree of relevant patterns in the decision subspace of rule R_j. If MC_j is high, this means that either the relevant patterns are well concentrated with high certainty grade (e.g. rule R_1 in Figure 7(a)), or the relevant and irrelevant patterns are mixed together. Both cases emphasize the reduction of the weight which will shrink the decision subspace of the rule. Since a higher value of MC_j needs lower rule weight, $1-MC_j$ is used in (11). So, the modified rule weight, $\overline{CF_j}$, is computed as:

$$\overline{CF_j} = CF_j.Conf_j.(1 - MC_j) \qquad (11)$$

Figure 8 shows the classification results when the modified rule weights are applied. This modification enhances the fuzzy classification system such that for Iris2f, the misclassified patterns reduce from 23 in Figure 7(c) to 4. For the Wine2f problem, this reduction is from 15 in Figure 7(d) to 11. Clearly, when some rules shrink because of reduction in their weight, the decision area of adjacent rules will grow accordingly, as shown in Figure 8.
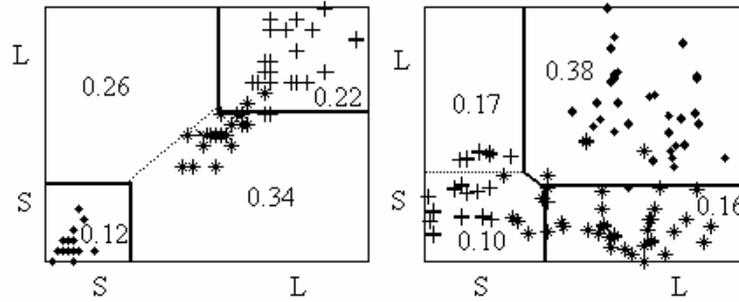

FIGURE 8. Decision area of fuzzy rules with modified weights

Figure 9 illustrates the decision subspace of fuzzy rules from Figure 6 when the modified rule weights are applied. For this case, the misclassified patterns are reduced from 6 to 2 for Iris2f and from 9 to 6 for Wine2f.

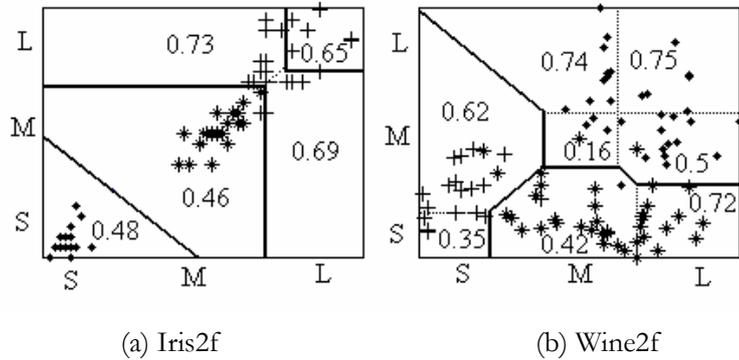(a) Iris2f   (b) Wine2f

FIGURE 9. Classification results of fuzzy rules with modified weights

5. Performance Evaluation

In this section, the performance of fuzzy rule-based classification systems is examined through computer simulations on some well-known data sets available from the UCI ML repository [4]. Table 1 shows the specifications of these data sets.

To illustrate graphically the effect of pattern distribution in generating fuzzy rules and improving the rule weights, some data sets with two attributes were needed. For this purpose, Fisher interclass separability criterion [18] is used to rank the features of Iris and Wine data sets. Two highest ranked features {4,3} for Iris and {13,12} for Wine are selected and called Iris2f and Wine2f in this paper.

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 1. Well-known data sets used for system evaluation

This paper uses the 10CV evaluation method [26] to examine the classification accuracy of designed systems. The 10CV (i.e. ten-fold cross-validation) method divides the data into ten subsets of the same size. Nine subsets are used as training data for generating fuzzy rules. The tenth subset is used as test data for evaluating the system. The same training and testing procedure is also performed nine times after exchanging the role of each subset such that all subsets are used as test data. Since the error rate on test data in the 10CV depends on the initial division of the data, the 10CV is iterated five times using different divisions of the data set and the average classification rate is reported as the performance of classifier.

In our computer simulations, the pattern space is homogeneously divided by K triangular fuzzy sets as in Figure 7. These fuzzy partitions are used for designing the fuzzy rule-based classification system. Our comparison illustrates the effect of rule weight and its improvement on fuzzy classification systems in Table 2. The accuracy of classification for five cases is compared:

- Fuzzy rules have no weight (i.e. 1).
- Fuzzy rules use certainty grade in (7) as weight (i.e. CF_j^1).
- Fuzzy rules use modified certainty grade in (11) for weight (i.e. $\overline{CF_j^1}$).
- Fuzzy rules use certainty grade in (9) as weight (i.e. CF_j^2).
- Fuzzy rules use modified certainty grade in (11) for weight (i.e. $\overline{CF_j^2}$).

As Table 2 shows, using certainty grade as rule weight enhances the performance of fuzzy rule-based classification systems considerably, especially for fine fuzzy partitioning (i.e., $K=4$ and $K=5$) [6]. For coarse fuzzy partitions (i.e., $K=2$ and $K=3$)

where decision subspace of fuzzy rules are almost large, the effect of certainty grades are not significant.

On the other hand, using modified certainty grade as rule weight can improve the classification accuracy of fuzzy rules. This effect is prominent especially when coarse fuzzy partitioning is used, or indeed when the decision subspace of fuzzy rules are large enough to contain some patterns. In this case, the distribution factors are used to decide if we should select more suitable weights for rules.

Data set	K	Weight of rule R_j					Average no. of rules
		1	CF_j^1	$\overline{CF_j^1}$	CF_j^2	$\overline{CF_j^2}$	
Iris	2	70.67	67.20	82.27	66.00	76.93	8.8
	3	91.33	92.67	94.40	92.67	94.67	14.7
	4	81.60	86.53	89.20	88.80	92.93	27.3
	5	94.67	95.33	95.33	95.60	95.46	45.6
Wine	2	86.86	91.01	92.02	92.25	92.25	114.8
	3	92.39	94.06	94.06	95.07	95.07	126.3
	4	95.09	95.09	95.09	95.09	95.09	154.6
	5	87.90	87.90	87.90	87.90	87.90	159.5
Glass	2	49.13	49.87	58.10	34.90	38.96	31.6
	3	59.73	62.81	62.45	61.10	62.43	41.8
	4	51.91	55.36	57.82	63.76	64.89	60.9
	5	59.43	61.52	61.51	57.12	58.03	87.8
Image	2	58.38	58.00	61.05	55.33	55.81	45.2
	3	75.90	77.14	82.00	82.19	83.62	95.0
	4	78.86	79.33	79.14	79.14	78.76	138.7
	5	77.90	78.57	78.00	78.00	78.19	156.1
Cancer	2	94.41	94.74	95.58	94.74	95.58	127.2
	3	91.31	91.40	91.43	91.40	91.43	231.7
	4	73.29	73.17	73.08	73.17	73.08	285.7
	5	66.80	66.80	66.83	66.80	66.83	296.6

TABLE 2. Classification accuracy of five weighting methods (10CV testing method)

To show the effectiveness of our weight improving algorithm, we compare it with weight learning approach proposed by Nozaki et al. [17]. In this learning procedure,

the certainty grade CF_j of the winner rule R_j is increased (i.e. rule R_j receives some reward) when X_p is correctly classified by R_j:

$$CF_j^{new} = CF_j^{old} + \eta_1(1 - CF_j^{old})$$ (12)

where η_1 is a positive learning rate for increasing the grade of certainty. On the other hand, when X_p is misclassified by R_j, the certainty grade CF_j is decreased as follows (i.e. some punishment may be given to R_j):

$$CF_j^{new} = CF_j^{old} - \eta_2 CF_j^{old}$$ (13)

where η_2 is a positive learning rate for decreasing the grade of certainty.

We examine the performance of this learning approach by computer simulations on the Iris and Glass data sets. We specify η_1=0.001 and η_2=0.1 as in [17]. This procedure is iterated only one time, because our proposed approach is non-iterative. The results are summarized in Table 3 for 10CV testing method. As shown, the enhancement obtained in classification accuracy using our approach is comparable to Nozaki's procedure.

Data set	K	Initial weight CF_j^2	Our weight $\overline{CF_j^2}$	Nozaki's weight CF_j^{new}
Iris	2	66.00	76.93	80.13
	3	92.67	94.67	95.07
	4	88.80	92.93	90.67
	5	95.60	95.46	95.07
Glass	2	34.90	38.96	45.66
	3	61.10	62.43	58.83
	4	63.76	64.89	64.60
	5	57.12	58.03	58.523

TABLE 3. Comparing the performance of two weight improving methods

However, Nozaki's procedure is essentially iterative. In order to make our comparison fair, we iterate both procedures 0 to 20 times (zero iteration for before learning) and compare their performance. Figure 10 illustrates the classification accuracy versus number of iterations for Iris and Glass. As shown, both approaches considerably improve the initially assigned weights, but their improvements are not permanent and there are some oscillations. Therefore, finding the best number of epochs is not simple and depends on data and the number of partitions. Moreover, two user-defined learning constants are required for Nozaki's procedure.
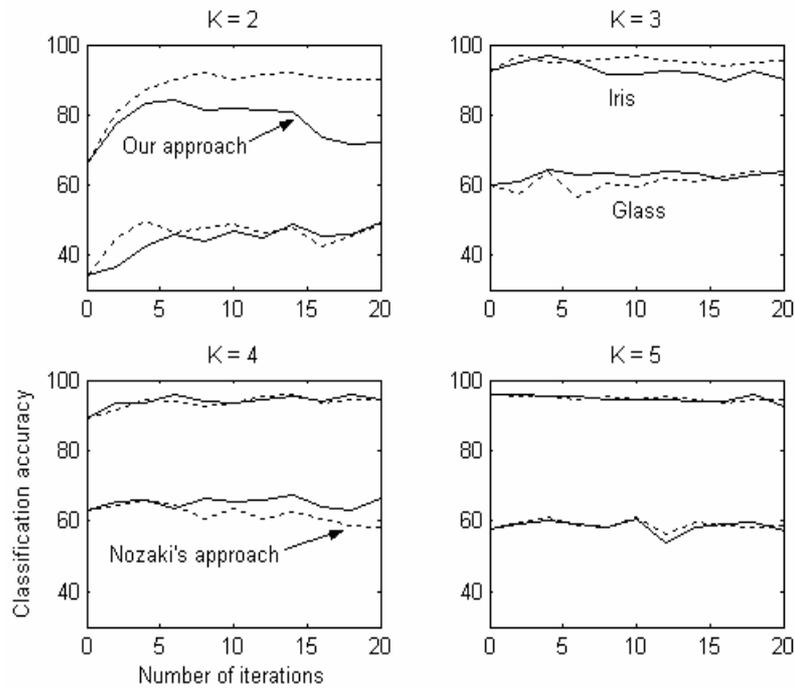
FIGURE 10. Comparing the performance of our approach with Nozaki's approach.

5. **Conclusion**

This paper examines the performance of fuzzy rules extracted from numerical data for pattern classification problems. The effect of rule weights on performance of fuzzy rules are investigated and it is shown that their effect on classification boundaries is almost the same as changing the membership functions of antecedent fuzzy sets. Experimental results show that the classification accuracy of fuzzy rules using certainty grades as rule weights is not significant for coarse fuzzy partitioning. It is possible that applying the improving algorithm based on distribution of patterns in decision subspace of rules may modify their weights. Computational experiments show that the proposed approach could improve classification accuracy, especially for coarse fuzzy partitions.

REFERENCES

[1] S. Abe and M. S. Lan, *A method for fuzzy rules extraction directly from numerical data and its application to pattern classification*, IEEE Transaction on Fuzzy Systems, **3 (1)** (1995), 18-28.
[2] S. Abe and R. Thawonmas, *A fuzzy classifier with ellipsoidal regions*, IEEE Transaction on Fuzzy Systems, **5 (3)** (1997), 358-368.

[3] J. C. Bezdek, Pattern Analysis. In E. H. Ruspini, P. Bonissone and W. Pedrycz, *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 and C. J. Merz, *UCI Repository of machine learning databases*, University of California, Department of Information and Computer Science, Irvine, CA, 1998.

[5] H. Ishibuchi and T. Nakashima, *Improving the performance of fuzzy classifier systems for pattern classification problems with continuous attributes*, IEEE Transaction on Industrial Electronics, **46 (6)** (1999), 157-168.

[6] H. Ishibuchi and T. Nakashima, *Effect of rule weights in fuzzy rule-based classification systems*, IEEE Transaction on Fuzzy Systems, **9 (4)** (2001), 506-515.

[7] H. Ishibuchi, T. Nakashima and T. Morisawa, *Voting in fuzzy rule-based systems for pattern classification problems*, Fuzzy sets and systems, **103 (2)** (1999), 223-238.

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

[9] H. Ishibuchi and T. Yamamoto, *Comparison of heuristic criteria for fuzzy rule selection in classification problems*, Fuzzy Optimization and Decision Making, **3 (2)** (2004), 119-139.

[10] H. Ishibuchi and T. Yamamoto, *Rule Weight Specification in Fuzzy Rule-Based Classification Systems*, IEEE Trans. on Fuzzy Systems, **13 (4)** (2005), 428-435.

[11] H. Ishibuchi, T. Yamamoto and T. Nakashima, *Fuzzy data mining: Effect of fuzzy discretization*, Proceeding of 1st IEEE International Conference on Data Mining, (2001), 241-248.

[12] L. I. Kuncheva, *Fuzzy Classifier Design*, Physica-Verlag, Heidelberg, 2000.

[13] L.I.Kuncheva and J. C. Bezdek, *A fuzzy generalized nearest prototype classifier*, Proceeding of 7th IFSA World Congress, Prague, **3** (1997), 217-222.

[14] S. Mitra and Y. Hayashi, *Neuro-fuzzy rule generation: Survey in soft computing framework*, IEEE Transaction on Neural Networks, **11 (3)** (2000), 748-768.

[15] D. Nauck and R. Kruse, *How the learning of rule weights affects the interpretability of fuzzy systems*, Proceeding of 7th IEEE International Conference on Fuzzy Systems, Anchorage, (1998), 1235-1240.

[16] D. Nauk and R. Kruse, *Obtaining interpretable fuzzy classification rules from medical data*, Artificial Intelligence in Medicine, **16** (1999), 149-169.

[17] K. Nozaki, H. Ishibuchi and H. Tanaka, *Adaptive Fuzzy Rule-Based Classification Systems*, IEEE Trans. on Fuzzy Systems, **4 (3)** (1996), 238-250.

[18] J. A. Roubos, M. Setnes and J. Abonyi, *Learning fuzzy classification rules from labeled data*, IEEE Transaction on Fuzzy Systems, **8 (5)** (2001), 509-522.

[19] D. Setiono, *Generating concise and accurate classification rules for breast cancer diagnosis*, Artificial Intelligence in Medicine, **18** (1999), 205-219.

[20] M. Setnes and R. Babuska, *Rule-Based Modeling: Precision and Transparency*, IEEE Transaction on Systems, Man, and Cybernetics-Part C: Applications and reviews, **28 (1)** (1998), 165-169.

[21] M. Setnes and R. Babuska, *Fuzzy relational classifier trained by fuzzy clustering*, IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, **29** (1999), 619-625.

[22] M. Setnes and J. A. Roubos, *GA-fuzzy modeling and classification: complexity and performance*, IEEE Transaction on Fuzzy Systems, **8 (5)** (2000), 509-522.

[23] J. Valente de Oliveira, *Semantic constraints for membership function optimization*, IEEE Transaction on Fuzzy Systems, **19 (1)** (1999), 128-138.

[24] J. Van den Berg, U. Kaymak and W. M. Van den Berg, *Fuzzy classification using probability based rule weighting*, proceeding of 11th IEEE International Conference on Fuzzy Systems, (2002), 991-996.

[25] L. Wang and Jerry M. Mendel, *Generating fuzzy rules by learning from examples*, IEEE Transaction on Systems, Man, and Cybernetics, **22 (6)** (1992), 1414-1427.

[26] S. M. Weiss and C. A. Kulikowski, *Computer Systems that Learn*, Morgan Kaufmann, San Mateo, 1991.

EGHBAL G. MANSOORI*, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, COLLEGE OF ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN

E-mail address: **mansoori@shirazu.ac.ir**

MANSOOR J. ZOLGHADRI, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, COLLEGE OF ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN

E-mail address: **zjahromi@shirazu.ac.ir**

SERAJ D. KATEBI, COMPUTER SCIENCE AND ENGINEERING DEPARTMENT, COLLEGE OF ENGINEERING, SHIRAZ UNIVERSITY, SHIRAZ, IRAN

E-mail address: **katebi@shirazu.ac.ir**

* CORRESPONDING AUTHOR