Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 177 (2007) 2296–2307
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Weighting fuzzy classification rules using receiver operating characteristics (ROC) analysis

Mansoor J. Zolghadri *, Eghbal G. Mansoori

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

Received 14 June 2005; received in revised form 2 December 2006; accepted 17 December 2006

Abstract

In fuzzy rule-based classification systems, rule weight has often been used to improve the classification accuracy. In past research, a number of heuristic methods for rule weight specification have been proposed. In this paper, a method of fuzzy rule weight specification using Receiver Operating Characteristic (ROC) analysis is proposed. In order to specify the weight of a fuzzy rule, using 2-class ROC analysis, the threshold that the rule achieves its maximum accuracy is found. This threshold is used as the weight of the rule. The proposed method is compared with existing ones through computer simulations on some well-known classification problems with continuous attributes. Simulation results show that the proposed method performs better than existing methods of fuzzy rule weight specification.

Keywords: Rule weight; ROC analysis; Pattern classification; Fuzzy systems; Data mining

1. Introduction

A Fuzzy Rule-Based Classification System (FRBCS) is a special case of fuzzy modeling where the output of the system is crisp and discrete. Basically, the design of a FRBCS consists of finding a compact set of fuzzy if-then classification rules to be able to model the input–output behavior of the system. The information available about the behavior of the system is assumed to be a set of input–output example pairs (i.e., a number of pre-labeled classification examples). Many approaches have been proposed for generating fuzzy classification rules from numerical data. These include heuristic approaches [1,17,25], neuro-fuzzy techniques [5,26,27,33], clustering methods [2,24,31], support vector machines [6,22] and genetic algorithms [4,10–13,15,16,21,29,32].

In this work, we assume that for each attribute of our classification problem, a number of pre-defined fuzzy sets, each having a linguistic meaning, is given by domain experts. Each fuzzy classification rule should use one of these fuzzy sets to specify the value of each attribute. The task of constructing a FRBCS can be handled by finding a compact set of fuzzy rules for the problem in hand. Rule weight can then be used as a simple mechanism to improve the classification performance of the constructed rule-base.

* Corresponding author. Tel./fax: +98 711 6271747.
E-mail addresses: zjahromi@shirazu.ac.ir (M.J. Zolghadri), mansoori@shirazu.ac.ir (E.G. Mansoori).

0020-0255/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2006.12.009

Rule weights are not used in many research works on FRBCSs (e.g. [2,4]). In many of these works, antecedent fuzzy sets are generated and adjusted using numerical data. As shown in [28], the learning of rule weights can be replaced by modification of membership functions of antecedent fuzzy sets. On the other hand, learning of rule weights can only partially replace the learning of membership functions of antecedent fuzzy sets. Since rule weight is a single parameter per rule, its adjustment is much easier than the learning of antecedent fuzzy sets (i.e., the learning of a number of parameter values of each membership function). Apart from simplicity of adjustment, another advantage of using rule weight is that the classification performance can be improved without changing the position of fuzzy sets given by domain experts. A number of heuristic measures have been proposed to specify the weight of a fuzzy classification rule (see Section 3). In this paper, we propose a new method of rule weight specification using ROC analysis.

ROC curve [9] is a technique for visualizing, organizing and selecting classifiers based on their performance. It can be used to choose the best operating point (i.e., threshold) of a classifier resulting in minimum cost or maximum accuracy. ROC curves are especially useful in case of having unequal classification error costs and unbalanced classes (i.e., skewed class distribution).

In this work, given a rule-base for a problem, we use 2-class ROC analysis to specify the weight of each rule in the system. ROC analysis is used to find the threshold that a rule achieves its best performance (i.e., maximum accuracy). The value of the best threshold which is a real number in the interval $[0,1]$ is used as the weight of the rule.

The rest of this paper is organized as follows. In Section 2, the structure of a FRBCS and the method of rule generation used in this paper are discussed. In Section 3, existing methods of rule weight specification are given. In Section 4, after introducing ROC curves, the mechanism for selecting the best operating point for 2-class problems is discussed. In Section 5, the proposed method of rule weight specification is presented. In Section 6, the simulation results are presented. Section 7 concludes the paper.

2. Fuzzy rule-based classification systems

A 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 [7]. We use fuzzy rules of the following type for an n-dimensional problem.

$$\text{Rule } R_j: \quad \text{If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then class } C_j \text{ with } \mathrm{CF}_j, \quad j = 1,2,\ldots,N. \tag{1}$$

where, $X = [x_1, x_2, \ldots, x_n]$ is the input feature vector, $C_j \in [C_1, C_2 \ldots, C_M]$ is the consequent class of the rule, 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

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

where $\mu_{ji}(\cdot)$ is the membership function of the antecedent fuzzy set A_{ji}.

In the case of using single winner reasoning method, the pattern is classified according to consequent class of winner rule R_w. With the rules of form (1), the certainty grade of each rule is also used in finding the winner rule:

$$\mu_w(X_t) \cdot \mathrm{CF}_w = \max\{\mu_j(X_t) \cdot \mathrm{CF}_j: \ j = 1,\ldots,N\} \tag{3}$$

$$w = \arg\max\{\mu_j(X_t) \cdot \mathrm{CF}_j: \ j = 1,\ldots,N\} \tag{4}$$

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 $\mu(X_t) \cdot \mathrm{CF}$ in Eq. (3).

2.1. Generating fuzzy rules

For an M-class problem in an n-dimensional feature space, assume that m labeled patterns $X_t = [x_{t1}, x_{t2}, \ldots, x_{tn}]$, $t = 1, 2, \ldots, m$ 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). 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 method to select the consequent class for an antecedent combination can be expressed in terms of confidence and support from the field of data mining [3]. 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) and support (denoted by S) of a fuzzy association rule [18] are defined as

$$C(A_j \Rightarrow \text{class } C_j) = \frac{\sum_{X_t \in \text{class } C_j} \mu_j(X_t)}{\sum_{t=1}^{m} \mu_j(X_t)} \tag{5}$$

$$S(A_j \Rightarrow \text{class } C_j) = \frac{1}{m} \sum_{X_t \in \text{class } C_j} \mu_j(X_t) \tag{6}$$

where, $\mu_j(X_t)$ is the compatibility grade of pattern X_t 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 is specified by finding the class with maximum confidence. This can be expressed as

$$q = \arg\max\{C(A_j \Rightarrow \text{class } h):\ h = 1, \ldots, M\} \tag{7}$$

When the consequent class C_j cannot be uniquely determined, the fuzzy rule is not generated.

The problem with grid partitioning is that an appropriate partitioning of each attribute is not usually known. One solution is to simultaneously consider different partitions shown in Fig. 1. That is, for each attribute, one of the 14 fuzzy sets shown in Fig. 1 can be used when generating a fuzzy rule. The problem is that for an n-dimensional problem, 14^n antecedent combinations should be considered. It is impractical to consider such a huge number of antecedent combinations when dealing with high dimensional problems.

One solution for the above problem is presented in [18] by adding the fuzzy set "don't care" to each attribute. The membership function of this fuzzy set is defined as $\mu_{\text{don't care}}(x) = 1$ for all values of x. The trick is not to consider all antecedent combinations (which is now 15^n and only short fuzzy rules having a limited number of antecedent conditions (excluding don't care) are generated as candidate rules.

The number of candidate rules generated with the above scheme can be quite large for many problems. A compact set of rules can be constructed in the following manner. The generated candidate rules are divided

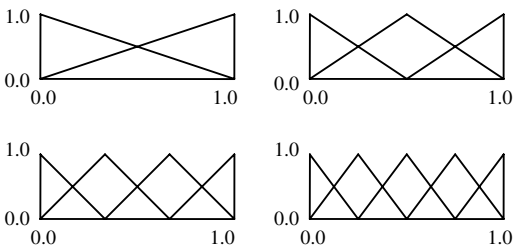

Fig. 1. Different partitioning of each feature axis.

into M groups according to their consequent classes. The candidate rules in each group are sorted in descending order of an evaluation criterion. A rule-base is constructed by choosing Q fuzzy rules from each class (i.e., $M{\cdot}Q$ fuzzy rules in total). Among many heuristic rule evaluation measures presented in the literature [19], we use the measure presented in [11]. The evaluation of rule R_j (i.e., $A_j \Rightarrow$ class C_j) with this measure can be expressed as

$$e(R_j) = \sum_{X_t \in \text{class } C_j} \mu_j(X_t) - \sum_{X_t \notin \text{class } C_j} \mu_j(X_t) \tag{8}$$

In practice, it can happen that multiple rules have the same value of rule evaluation criterion. To select a rule, we apply the following criteria in a lexicographical order.

1. Rule evaluation measure (Eq. (8)).
2. Number of antecedent conditions.
3. The size of covering subspace of the rule that is calculated by multiplying the areas of fuzzy sets in the antecedent.

Therefore, measure (8) is used as the main criterion. In a tie situation, the rule having fewer antecedent conditions is selected. In case of a new tie situation, the rule having larger covering space is selected. In case multiple rules having the same value of the three criteria, a rule is selected in random among those best fuzzy rules.

The above simple method of rule-base construction should not be expected to achieve high classification performance. The reason for this is that:

1. An equal number of rules are selected from each class to construct a rule-base. In an optimal rule-base (i.e., with high classification accuracy), the number of rules from each class may be different.
2. The combinatorial effect of fuzzy rules (i.e., the interaction among them) is not taken into account when a rule-base is constructed.

We use this simple method of rule-base construction since our intention is to compare various rule weighting schemes with each other (i.e., not to construct optimal rule-bases). For a specific data set, we can easily construct many rule-bases by varying the parameter Q and asses the performance of each method of rule weighting across these rule-bases.

In order to construct rule-bases with high classification performance, above problems can be solved by directly searching for good rule sets (i.e., combination of fuzzy rules). One way of doing this already used by other researchers is to use Genetic Algorithms (GA) to directly search for good subsets of candidate rules generated using the heuristic criterion [18,19]. In this scheme, each individual in GA population represents a rule-base. Error rate on train data is usually used to evaluate an individual. A rule-weighting scheme is used to tune the rule-base before calculating its fitness. The choice of rule weighting scheme obviously can have a significant effect on the classification performance of the final rule-base.

3. Rule weight specification methods

In order to assign a weight to each fuzzy classification rule, four heuristic measures proposed in past research will be investigated here. In the first method [7], the confidence of the fuzzy association rule is used as its weight

$$\mathrm{CF}_j^1 = C(A_j \Rightarrow \text{class } C_j) \tag{9}$$

The second definition of rule weight proposed in [14] can be stated as

$$\mathrm{CF}_j^2 = C(A_j \Rightarrow \text{class } C_j) - C_{\text{Average}} \tag{10}$$

where, C_{Average} is the average confidence of the fuzzy association rules having the same antecedent but consequent classes other than C_j. For an M-class problem, it can be expressed as

$$C_{\text{Average}} = \frac{1}{M-1} \sum_{\substack{h=1 \\ h \neq C_j}}^{M} C(A_j \Rightarrow \text{class } h) \tag{11}$$

The third definition of rule weight used in [18,20] can be stated as

$$\text{CF}_j^3 = C(A_j \Rightarrow \text{class } C_j) - C_{\text{Second}} \tag{12}$$

where, C_{Second} is the largest confidence of fuzzy association rules having the same antecedent but consequent classes other than C_j.

$$C_{\text{Second}} = \max\{C(A_j \Rightarrow \text{class } h):\ h = 1, \ldots, M \text{ and } h \neq C_j\} \tag{13}$$

The fourth definition of rule weight proposed in [20] can be stated as

$$\text{CF}_j^4 = C(A_j \Rightarrow \text{class } C_j) - C_{\text{Sum}} \tag{14}$$

where, C_{Sum} is the sum of confidence over fuzzy association rules having the same antecedent but consequent classes other than C_j.

$$C_{\text{Sum}} = \sum_{\substack{h=1 \\ h \neq C_j}}^{M} C(A_j \Rightarrow \text{class } h) \tag{15}$$

Note that with fourth definition of rule weight (14), the weight of a rule can be negative while with other definitions it is always positive. The weight of the rule is set to zero in case the calculated value of rule weight using Eq. (14) is negative.

4. ROC curves

ROC curve is a technique for visualizing the performance of a classifier as a trade-off between sensitivity and selectivity. It was first introduced in signal detection theory to demonstrate how well a receiver distinguishes a signal from noise [8]. ROC analysis has been extensively used in medical diagnostic tests to specify sensitivity/specificity trades-off. Recently, a lot of interest is shown to use ROC as an analysis tool in the field of machine learning [23,30].

A discrete classifier such as a classification tree only produces a class label for an input pattern. For a 2-class problem (with positive (p) and negative (n) class labels), given a test set of P positive and N negative labeled patterns, a classifier of this type generates a 2×2 confusion matrix (shown in Fig. 2) representing the performance of the classifier.

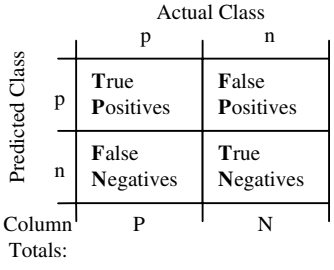

Fig. 2. Confusion matrix for a discrete classifier.

Some common metrics, which can be calculated from the confusion matrix, are:

$$\text{True positive (TP) rate} = \frac{\text{TP}}{P} \tag{16}$$

$$\text{False positive (FP) rate} = \frac{\text{FP}}{N} \tag{17}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{18}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{P + N} \tag{19}$$

The ROC curve for a two-class problem is a two dimensional curve in which TP rate is plotted on y axis and FP rate is plotted on x axis. Discrete classifiers produce a single (TP rate, FP rate) pair, thus produce a single point in ROC space. Many classifiers, such as Bayesian classifiers or FRBCSs, naturally assign a score $f(X_t)$ to each input pattern X_t (i.e., scoring classifiers). The score is a numeric value expressing the degree that X_t is thought to be positive. For example, naive Bayes classifiers output posterior probability distribution over classes. Let $\text{pr}(p, X_t)$ and $\text{pr}(n, X_t)$ denote the estimated probabilities that the pattern X_t is of positive and negative class, respectively. The score of a pattern can be defined as

$$f(X_t) = \frac{\text{pr}(p, X_t)}{\text{pr}(n, X_t)} = \frac{\text{pr}(p, X_t)}{1 - \text{pr}(p, X_t)} \tag{20}$$

A scoring classifier can be converted to a discrete classifier by specifying a threshold on score. A pattern is classified as positive if its score of the pattern is greater than the specified threshold, and negative otherwise. In this way, each threshold produces a different point in ROC space. As the threshold is varied from 0 to ∞, a ROC curve can be drawn. The ROC curve starts from the point $(0,0)$ where the classifier finds no positives (everything is classified as negative) and stops at point $(1,1)$ where the classifier finds no negative (everything is classified as positive).

An efficient method of constructing ROC curves is given in [9]. In this method, the patterns are sorted in decreasing order of their scores. Starting at $(0,0)$, the ROC curve is drawn by moving $\frac{1}{P}$ up (along y-axis) if the next pattern is an actual positive and moving $\frac{1}{N}$ to the right (along x-axis) if the next instance is an actual negative until we reach the point $(1,1)$. Since it can happen that some patterns from both classes having the same score (i.e., n_1 patterns from positive class and n_2 patterns from negative class), the ROC curve is drawn by a single diagonal element moving $\frac{n_1}{P}$ up and $\frac{n_2}{N}$ to the right simultaneously.

4.1. Learning the best operating point for 2-class problems

ROC analysis provides an elegant mechanism for choosing the best operating point when the misclassification costs are unequal. For two class problems, the best operating point must be chosen such that the classifier gives the best trade-off between the costs of failing to detect positives against the cost of failing to detect negatives. Let C_N be the cost of misclassifying a negative and C_P be the cost of misclassifying a positive. The expected cost of a point (x,y) in ROC space is

$$\text{expected cost} = \frac{P \cdot C_P \cdot (1 - y) + N \cdot C_N \cdot x}{P + N} \tag{21}$$

In this paper, we assume that misclassification costs are equal to each other and unity (i.e., $C_P = C_N = 1$). For a 2-class problem with P positive and N negative labeled patterns, the expected accuracy corresponding to a specific threshold is

$$\text{expected accuracy} = \frac{\text{TP} - \text{FP}}{P + N} + \frac{N}{P + N} \tag{22}$$

Since we are only interested in finding a threshold which maximizes Eq. (22), with P and N being constant for a problem, we can equivalently search for a threshold maximizing the following measure:

$$\text{accuracy-measure} = \text{TP} - \text{FP} \tag{23}$$

Having the relation between a threshold and corresponding accuracy of the classifier, the best threshold can be easily found by varying the threshold from 0 to ∞. Actually, it is sufficient to consider those thresholds that the classification of an instance changes from positive to negative. For this purpose, the patterns are ranked in descending order of their scores (i.e., $f(X_1) > f(X_2) > \cdots > f(X_{P+N})$). Considering any threshold between $f(X_K)$ and $f(X_{K+1})$, the first K patterns will be classified as positive and the remaining $P + N - K$ patterns as negative. In this way, a maximum of $P + N+1$ thresholds should be examined to find the best threshold. The first threshold classifies everything as negative and the last threshold classifies everything as positive. The rest of the thresholds are chosen in the middle of two successive scores $f(X_K), f(X_{K+1})$ in the list where $f(X_K) \neq f(X_{K+1})$. The best threshold is the one that maximizes Eq. (23).

5. Rule weight specification using ROC

For a specific problem, assume that a rule-base is constructed using the mechanism described in Section 2.1. Our aim in this section is to propose a rule-weight specification method based on ROC analysis. Consider the following rule as a typical rule of the system.

$$\text{Rule } R_j\text{: If } x_1 \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{jn} \text{ then class } C_j \tag{24}$$

In order to use the mechanism described in Section 4.1, a 2-class situation is formed by denoting the consequent class of the rule (class C_j) as positive and all other classes as negative. That is, in rule weighting mechanism given in this section, class p denotes the consequent class of the rule under investigation (to specify its weight) and class n is formed by merging all other classes. The score of each training pattern X_t covered by rule R_j can be defined as

$$f(X_t) = \frac{\mu_p(X_t)}{\mu_n(X_t)} \tag{25}$$

where $\mu_p(X_t)$ denotes the compatibility grade of pattern X_t with rule R_j (i.e., $\mu_p(X_t) = \mu_j(X_t)$) and $\mu_n(X_t)$ is defined as

$$\mu_n(X_t) = \max\{\mu_j(X_t) | R_j \in \text{rule-base}, \text{Consequent}(R_j) \neq \text{Class } p\} \tag{26}$$

The numerator and denominator of Eq. (25) specify the degree to which X_t is thought to be of class p and class n, respectively.

Note that in calculating the score of each pattern covered by rule R_j, all other rules in the system having class C_j in the consequent are ignored. In other words, to specify the weight of this rule, we assume that this is the only rule in the system having class C_j in the consequent. Using Eq. (25), the score of each training pattern will be in the range of 0 to ∞. In this work, we use the following definition of score which normalizes the score to a value in the interval $[0, 1]$.

$$\text{Score}(X_t) = \frac{\mu_n(X_t)}{\mu_n(X_t) + \mu_p(X_t)} \tag{27}$$

Notice that the two definitions of score (Eqs. (25) and (27)) are related in the following way:

$$\text{Score}(X_t) = \frac{1}{1 + f(X_t)} \tag{28}$$

That is, ranking positions of patterns will not be affected if we use Eq. (27) as the definition of score and rank the patterns in ascending order of their scores (instead of ranking them in descending order using (25)).

In order to specify the weight of fuzzy rule R_j, in the first step, the scores of the patterns in the covering area of the rule are calculated using Eq. (27). In the next step, the best threshold on score resulting in maximum accuracy is calculated (using the mechanism described in Section 4.1). An algorithm for doing this is given in Table 1. This algorithm receives the set of patterns X_t and their scores Score(X_t) as input and returns the best threshold as output. The value of the best threshold is simply used as the weight of the rule (i.e.,

Table 1
Algorithm for finding the best threshold

Input: patterns X_t, scores $Score(X_t)$
Output: the threshold (th) resulting in maximum accuracy
  $current$ = accuracy-measure (Eq. (23)) corresponding to the threshold of $th = 0$ (i.e., classifying everything as negative)
  $optimum = current$
  $best\text{-}threshold = 0$
  rank the patterns in ascending order of their scores
  {assume that x_k and X_{k+1} are two successive patterns in the list}
  for each different threshold $th = (Score(X_k) + Score(X_{k+1}))/2$ **do**
    $current$ = accuracy-measure corresponding to the specified threshold (i.e., all patterns X_t having $Score(X_t) < th$ are classified as positive)
    if $current > optimum$ **then**
      $optimum = current$
      $best\text{-}threshold = th$
    end if
  end for
  $current$ = accuracy-measure corresponding to $th = 1$ (i.e., classifying everything as positive)
  if $current > optimum$ **then**
    $optimum = current$
    $best\text{-}threshold = 1$
  end if
return *best-threshold*

CF_j = best-threshold). Notice that with scores of patterns being in the interval $[0, 1]$, the weight assigned to each rule will be in this interval.

With the four definitions of rule weight discussed in Section 3, the weight assigned to a fuzzy rule is independent of other rules in the rule-base. On the other hand, our proposed method considers the rules of negative class in the rule-base to specify the weight of a rule. That is, the weight of a rule changes when included in different rule-bases.

6. Simulation results

In this section, we evaluate the performance of the proposed rule-weighting scheme in comparison with other methods discussed in Section 3. In computer simulations, we used four data sets in Table 2 available from UCI machine learning repository. For each data set, each attribute was normalized into a real number in the interval $[0, 1]$. Using "don't care" in addition to 14 fuzzy sets in Fig. 1; we generated all fuzzy rules having two antecedent conditions or less (excluding "don't care"). The consequent of each fuzzy rule was specified using Eq. (7). The generated fuzzy rules were divided into M groups according to their consequent classes. A rule-base was then constructed by choosing Q fuzzy rules from each group using the three criteria given in Section 2.1. We used various values of Q (i.e., $Q = 1, 2, \ldots, 10$) to examine the classification rate of fuzzy rule-bases of different sizes. For each data set, we report on the classification rate for the following cases:

- Full Train–Full Test (FT–FT): The full data set is used in the training phase to construct the rule-bases of different sizes (i.e., $Q = 1, 2, \ldots, 10$). The full data set is also used in the testing phase to evaluate various rule-bases.

Table 2
Data sets used in this paper

Data set	No. of attributes	No. of samples	No. of classes
Glass	9	214	6
Pima	8	768	2
B. cancer	9	684	2
Wine	13	178	3

- Leave one out (LV1): One sample is put aside for the testing phase and the rest of samples are used in the training phase (i.e., to construct the rule-base). The procedure is repeated until all the samples are used in the testing phase. The average classification rate on test data is reported as the performance of classifier.

Tables 3–6 give the FT–FT classification performance for the data sets used in this paper. For each constructed rule-base, we report on classification rate for the case of no rule weight, the four heuristic definitions of rule weight and our proposed method. The best result in each column (i.e., each specification of Q) is shown in boldface. As seen, the proposed method achieves the best performance in nearly all experimental cases (i.e., across different values of Q for all data sets).

Tables 7–10 give the average classification rates on test patterns of various data sets using LV1 evaluation method. We can see that the classification performance of fuzzy rule-based systems with no rule weights are

Table 3
FT–FT evaluation of Glass data set

Rules/class	1	2	3	4	5	6	7	8	9	10
No. weight	51.86	52.33	53.27	56.54	58.41	60.28	60.28	60.28	66.82	66.82
CF^1	53.27	53.33	54.20	58.87	59.81	61.21	61.21	61.21	66.35	66.35
CF^2	53.27	53.33	54.20	57.94	59.34	60.74	60.74	60.74	65.88	65.88
CF^3	54.20	55.14	56.07	57.47	59.34	61.68	61.68	61.68	64.01	64.01
CF^4	57.00	57.47	57.00	59.34	61.68	62.14	62.14	62.14	62.14	62.14
ROC	**60.74**	**61.21**	**60.28**	**63.55**	**66.35**	**66.82**	**66.35**	**66.35**	**70.09**	**70.09**

Table 4
FT–FT evaluation of Pima data set

Rules/class	1	2	3	4	5	6	7	8	9	10
No. weight	73.30	73.30	73.30	67.44	67.57	67.18	67.18	67.18	67.18	67.18
CF^1	73.30	73.30	73.30	68.61	68.75	67.83	67.83	67.83	67.83	67.83
CF^2	73.82	73.56	73.82	70.96	71.09	70.57	70.57	70.57	70.57	69.92
CF^3	73.82	73.56	73.82	70.96	71.09	70.57	70.57	70.57	70.57	69.92
CF^4	73.82	73.56	73.82	70.96	71.09	70.57	70.57	70.57	70.57	69.92
ROC	**73.95**	**74.73**	**74.73**	**74.86**	**74.86**	**74.60**	**74.60**	**74.60**	**74.60**	**75.00**

Table 5
FT–FT evaluation of B. cancer data set

Rules/class	1	2	3	4	5	6	7	8	9	10
No. weight	94.44	93.42	93.42	95.17	94.88	94.88	95.02	95.02	95.32	95.32
CF^1	94.44	93.12	93.12	94.88	94.44	94.44	94.73	94.73	95.17	95.32
CF^2	94.29	93.12	93.12	94.73	94.29	94.29	94.59	94.59	95.17	95.32
CF^3	94.29	93.12	93.12	94.73	94.29	94.29	94.59	94.59	95.17	95.32
CF^4	94.29	93.12	93.12	94.73	94.29	94.29	94.59	94.59	95.17	95.32
ROC	**94.73**	**93.56**	**93.56**	**95.90**	**95.76**	**95.76**	**96.05**	**96.05**	**96.34**	**96.34**

Table 6
FT–FT evaluation of Wine data set

Rules/class	1	2	3	4	5	6	7	8	9	10
No. weight	89.32	93.82	92.69	92.69	92.69	93.82	93.82	93.82	93.25	93.82
CF^1	89.32	93.25	92.69	93.25	93.25	93.25	93.82	93.82	93.82	94.38
CF^2	89.88	93.25	92.69	93.82	93.82	93.25	93.82	93.82	93.82	94.94
CF^3	89.32	93.25	93.25	**95.50**	**95.50**	**95.50**	**95.50**	**95.50**	**94.94**	**96.06**
CF^4	89.88	93.25	93.25	94.94	94.94	94.94	**95.50**	**95.50**	94.38	**96.06**
ROC	**91.01**	**94.38**	**94.38**	**95.50**	**95.50**	**95.50**	**95.50**	**95.50**	94.38	**96.06**

Table 7
Leave one out evaluation of Glass data set

Rules/class	1	2	3	4	5	6	7	8	9	10
No. weight	48.13	49.53	48.59	49.06	49.06	50.46	53.73	52.33	50.46	45.32
CF^1	49.53	50.93	50.00	50.46	50.93	53.27	56.07	54.67	53.27	49.53
CF^2	49.53	50.93	50.00	50.46	50.93	53.27	56.07	54.67	53.27	50.46
CF^3	50.00	50.93	49.53	49.06	50.00	52.80	56.07	55.14	55.14	55.14
CF^4	49.53	50.93	50.00	50.46	50.93	53.27	56.07	54.67	53.27	49.53
ROC	**57.94**	**59.34**	**56.54**	**57.47**	**57.00**	**59.81**	**61.21**	**60.28**	**62.61**	**63.08**

Table 8
LV1 evaluation of Pima data set

Rules/class	1	2	3	4	5	6	7	8	9	10
No. weight	73.30	73.30	65.23	67.44	67.57	67.18	67.18	67.18	67.18	67.18
CF^1	73.30	73.30	66.40	68.61	68.61	67.83	67.83	67.83	67.83	67.70
CF^2	73.82	73.43	69.53	70.83	70.96	70.44	70.44	70.44	70.44	70.05
CF^3	**73.82**	73.43	69.53	70.83	70.96	70.44	70.44	70.44	70.44	70.05
CF^4	73.30	73.30	66.40	68.61	68.61	67.83	67.83	67.83	67.83	67.70
ROC	**73.82**	**73.82**	**74.60**	**74.73**	**74.21**	**74.60**	**74.60**	**74.60**	**74.60**	**74.60**

Table 9
LV1 evaluation of B. cancer data set

Rules/class	1	2	3	4	5	6	7	8	9	10
No. weight	**94.44**	**93.42**	**92.98**	**93.12**	94.88	94.88	95.02	95.02	95.02	95.32
CF^1	**94.44**	93.12	92.69	92.83	94.29	94.29	94.73	94.73	94.88	95.32
CF^2	94.29	93.12	92.69	92.83	94.15	94.15	94.59	94.73	94.88	95.32
CF^3	94.29	93.12	92.69	92.83	94.15	94.15	94.59	94.73	94.88	95.32
CF^4	**94.44**	93.12	92.69	92.83	94.29	94.29	94.73	94.73	94.88	95.32
ROC	**94.44**	**93.42**	**92.98**	92.69	**95.46**	**95.76**	**96.05**	**96.05**	**95.76**	**95.76**

Table 10
LV1 evaluation of Wine data set

Rules/class	1	2	3	4	5	6	7	8	9	10
No. weight	89.32	**93.82**	92.69	89.32	89.88	90.44	92.13	92.13	**92.13**	92.13
CF^1	89.32	93.25	92.69	90.44	90.44	91.01	92.69	91.57	91.57	91.57
CF^2	89.32	93.25	92.69	90.44	90.44	90.44	92.69	90.44	90.44	91.01
CF^3	89.32	93.25	92.69	92.13	91.57	91.57	93.82	92.13	**92.13**	**92.69**
CF^4	89.32	93.25	92.69	90.44	90.44	91.01	92.69	91.57	91.57	91.57
ROC	**90.44**	93.25	**93.25**	**92.69**	**92.69**	**92.69**	**94.38**	**93.82**	**92.13**	92.13

improved in most cases using the proposed rule-weighting scheme. While the amount of improvement is small in case of Wine and B. cancer data sets, in case of Glass and Pima data sets, a significant amount of improvement is observed only with the proposed method of rule-weighting. One possible reason for this is that Wine and B. cancer data sets do not have large overlap regions between different classes in the pattern space. For these data sets, the proposed method (and in fact all other rule-weighting schemes) does not have a large effect on the performance. In case of Glass and Pima data sets, with highly overlapped classes, a significant amount of improvement is observed only with the proposed method of rule-weighting.

As the results of Tables 7–10 suggest, none of the past methods of rule-weighting (i.e. $CF^1 - CF^4$) has a clear advantage over the others. Selecting any of these methods does not have a large effect on the classification performance. On the other hand, we can see that the proposed method performs consistently better in

case of Glass and Pima data sets. The difference in performance of the proposed method with other methods is significant for these data sets. For B. cancer and Wine data sets, the proposed method performs better in most experimental cases. For these data sets, the difference in performance of the proposed method with other methods is small.

Obviously, the computation time of the proposed method is higher in comparison with other methods. To give you an estimate, in our computer program, specification of rule weights with the proposed method for the case of Glass data set with 10 rules per class (i.e. a rule-base having 60 rules) takes about 2 s it takes around 0.4 s with other methods. For the same case, the construction of initial rule-base takes about 15 s in our simulation. As seen, the major bottleneck is the computation time needed to construct the initial rule-base not the time required for rule-weighting.

A point to mention is that, when the value of measure (8) for a fuzzy rule is negative, the calculated weight using the fourth definition of rule weight (14) is negative while with other definitions it is always positive. As we have already mentioned, we do not use fuzzy rules with negative weights. Fuzzy rules with negative weights are not used in case of fourth definition while they are used in other definitions. That is, the number of used fuzzy rules in different rule-bases of Tables 3–10 can be different between fourth and other definitions. In order to evaluate the pure effect of rule weights on the classification performance, all the fuzzy rules that are used to construct different rule-bases in Tables 3–10 have in fact a positive value of measure (8).

7. Conclusions

In this paper, we proposed a new method of rule weight specification in fuzzy rule-base classification systems. Assuming that a rule-base for a problem is available, we used 2-class ROC analysis to find the best threshold (i.e., resulting in maximum accuracy) for each rule in the rule-base. We introduced a new definition of score for this purpose that gives the score of a pattern by a real number in the interval $[0, 1]$. The weight of a fuzzy rule was then specified as its best threshold value.

Computer simulations on four well-known data sets with continuous attributes showed that the proposed method can improve the performance of initial rule-base (i.e., the case of not using rule weight). For Glass and Pima data sets, a significant amount of improvement was observed. In case of Wine and B. cancer data sets, the rule-weighting scheme did not have a large effect on the classification rate.

In comparison with other rule-weight specification methods, simulation results showed that the proposed method performs better in vast majority of simulated cases. While the difference in the performance of the proposed method with other methods is rather large in the case of Glass and Pima data sets, in case of B. cancer and Wine data sets, a significant difference was not observed.

References

[1] S. Abe, 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 (1995) 18–28.
[2] S. Abe, R. Thawonmas, A fuzzy classifier with ellipsoidal regions, IEEE Transaction on Fuzzy Systems 5 (3) (1997) 358–368.
[3] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proceeding of 20th International Conference on Very large Databases, 1994, pp. 487–499.
[4] J. Casillas, O. Cordón, M.J. Del Jesus, F. Herrera, Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems, Information Sciences 136 (1–4) (2001) 135–157.
[5] D. Chakraborty, N.R. Pal, A neuro-fuzzy scheme for simultaneous feature selection and fuzzy rule-based classification, IEEE Transaction on Neural Networks 15 (1) (2004) 110–123.
[6] Y. Chen, J.Z. Wang, Support vector learning for fuzzy rule-based classification systems, IEEE Transaction on Fuzzy Systems 11 (6) (2003) 716–728.
[7] 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.
[8] J. Egan, Signal Detection Theory and ROC Analysis, Academic, New York, 1975.
[9] T. Fawcett, ROC graphs: notes and practical considerations for researchers, Technical Report HPL-2003-4, HP Labs, 2004.
[10] A.F. Gómez-Skarmeta, M. Valdés, F. Jiménez, J.G. Marín-Blázquez, Approximative fuzzy rules approaches for classification with hybrid-GA techniques, Information Sciences 136 (1–4) (2001) 193–214.
[11] A. Gonzalez, R. Perez, SLAVE: A genetic learning system based on an iterative approach, IEEE Transaction on Fuzzy Systems 7 (2) (1999) 176–191.

[12] S.Y. Ho, H.M. Chen, S.J. Ho, T.K. Chen, Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space, IEEE Transaction on Systems, Man and Cybernetics, Part B 34 (2) (2004) 1031–1044.
[13] Y.C. Hu, Finding useful fuzzy concepts for pattern classification using genetic algorithm, Information Sciences 175 (1–2) (2005) 1–19.
[14] H. Ishibuchi, T. Nakashima, Effect of rule weights in fuzzy rule-based classification systems, IEEE Transaction on Fuzzy Systems 9 (4) (2001) 506–515.
[15] H. Ishibuchi, T. Nakashima, T. Murata, Three-objective genetics-based machine learning for linguistic rule extraction, Information Sciences 136 (1–4) (2001) 109–133.
[16] H. Ishibuchi, Y. Nojima, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multi-objective fuzzy genetics-based machine learning, International Journal of Approximate Reasoning 44 (1) (2007) 4–31.
[17] 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.
[18] 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.
[19] H. Ishibuchi, T. Yamamoto, Comparison of heuristic criteria for fuzzy rule selection in classification problems, Fuzzy Optimization and Decision Making 3 (2) (2004) 119–139.
[20] H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, IEEE Transaction on Fuzzy Systems 13 (4) (2005) 428–435.
[21] H. Ishibuchi, T. Yamamoto, T. Nakashima, Hybridization of fuzzy GBML approaches for pattern classification problems, IEEE Transaction on Systems, Man and Cybernetics, Part B 35 (2) (2005) 359–365.
[22] B. Jin, Y.C. Tang, Y.Q. Zhang, Support vector machines with genetic fuzzy feature transformation for biomedical data classification, Information Sciences 177 (2) (2007) 476–489.
[23] N. Lachiche, P. Flach, Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves, in: Proceeding of 20th International Conference on Machine Learning (ICML'03), 2003, pp. 416–423.
[24] C.Y. Lee, C.J. Lin, H.J. Chen, A self-constructing fuzzy CMAC model and its applications, Information Sciences 177 (1) (2007) 264–280.
[25] E.G. Mansoori, M.J. Zolghadri, S.D. Katebi, A weighting function for improving fuzzy classification systems performance, Fuzzy Sets and Systems 158 (5) (2007) 583–591.
[26] 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.
[27] D. Nauck, R. Kruse, A neuro-fuzzy method to learn fuzzy classification rules from data, Fuzzy Sets and Systems 89 (3) (1997) 277–288.
[28] D. Nauck, R. Kruse, How the learning of rule weights affects the interpretability of fuzzy systems, in: Proceeding of 7th IEEE international conference on fuzzy systems, 1998, pp. 1235–1240.
[29] T. Ozyer, R. Alhajj, K. Barker, Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening, Journal of Network and Computer Applications 30 (1) (2007) 99–113.
[30] F. Provost, T. Fawcett, Robust classification for imprecise environments, Machine Learning Journal 42 (3) (2001) 203–231.
[31] J.A. Roubos, M. Setnes, J. Abonyi, Learning fuzzy classification rules from labeled data, Information Sciences 150 (1–2) (2003) 77–93.
[32] L. Sánchez, I. Couso, J.A. Corrales, O. Cordón, M.J. Del Jesus, F. Herrera, Combining GP operators with SA search to evolve fuzzy rule based classifiers, Information Sciences 136 (1–4) (2001) 175–191.
[33] V. Uebele, S. Abe, M.S. Lan, A neural-network-based fuzzy classifier, IEEE Transaction on Systems, Man, and Cybernetics 25 (2) (1995) 353–361.