Performance Comparison between Fuzzy Rules and Interval Rules in Rule-Based Classification Systems

Satoshi Namba, Yusuke Nojima, and Hisao Ishibuchi
Department of Industrial Engineering
Osaka Prefecture University
Gakuen-cho 1-1, Sakai, Osaka 599-8531, Japan
{snamba, nojima, hisaoi}@ie.osakafu-u.ac.jp

Abstract

In this paper, we compare the performance of fuzzy rule-based classification systems with that of interval rule-based classification systems on real-world data sets. We generate multiple rule-based classification systems using an evolutionary multiobjective optimization (EMO) algorithm. In our EMO approach, the classification accuracy of rule sets is maximized, while their complexity is minimized. A number of non-dominated rule sets with different accuracy and different complexity are obtained by its single run. Through computational experiments on some data sets, we compare the generalization ability of fuzzy rule-based systems with that of interval rule-based systems using the ten-fold cross-validation technique. Furthermore, we perform the same computational experiments with fuzzified interval rule-based classification systems. From experimental results, we clearly demonstrate characteristics of each classification system: fuzzy rule-based classification systems have high generalization ability on test data, interval classification systems can achieve high accuracy on training data, and fuzzified interval classification systems have these advantages of two other classification systems.

1 Introduction

Fuzzy rule-based systems have been applied to various problems such as control, function approximation and pattern classification. The tradeoff between the accuracy and the complexity of fuzzy rule-based systems was often discussed in recent studies [1, 2]. When we design fuzzy rule-based classification systems, it should be noted that the maximization of the accuracy on training data often leads to the overfitting, which degrades the actual performance of fuzzy rule-based classification systems on test data.

Fuzzy rule-based systems are interpretable for human users. In our former study on fuzzy rule extraction [2], antecedent fuzzy sets in fuzzy rule-based classification systems are homogeneously triangular. Homogeneous fuzzy discretization works well because there are significant overlaps between adjacent fuzzy sets. Decision boundaries are adjustable over those overlapping regions using the certainty grade of each fuzzy rule. On the other hand, antecedent intervals in interval rule-based classification systems are generated from given numerical data using an entropy measure in some approaches to the design of decision trees [3]. We combine some advantages of fuzzy rule-based systems and interval ones into fuzzified interval rule-based classification systems. That is, the antecedent fuzzy sets in fuzzified interval rule-based classification systems are derived from intervals in the interval rule-based systems by fuzzification.

In this paper, we compare the performance of fuzzy rule-based classification systems with that of interval rule-based classification systems. All the rule-based classification systems are constructed by using our EMO approach [2]. Furthermore, we perform the same computational experiments with fuzzified interval rule-based classification systems. Experimental results show that interval rule-based classification systems can obtain high accuracy on training data, while high generalization ability on test data can be obtained by fuzzy rule-based classification systems. It is also shown that the generalization ability of the interval rule-based classification systems can be improved by their fuzzification.

2 Rule-Based Classification Systems

We use the following type of rules for an n-dimensional pattern classification problem:

$$\text{Rule } R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and ... and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q , \quad (1)$$

where $\mathbf{x} = (x_1, ..., x_n)$ is an n-dimensional pattern vector, $\mathbf{A}_q = (A_{q1}, ..., A_{qn})$ is an antecedent part (i.e., fuzzy sets in fuzzy rule-based systems, and intervals in interval ones), C_q is a consequent class, and CF_q is a rule weight (i.e., certainty grade). First we explain how the consequent class C_q of the classification rule R_q in (1) is specified from numerical data. Let us assume that we have m labeled patterns $\mathbf{x}_p = (x_{p1},...,x_{pn})$, $p = 1,2,...,m$ from M classes. We define the compatibility grade $\mu_{\mathbf{A}_q}$ of each training pattern $\mathbf{x}_p$ with the antecedent part $\mathbf{A}_q$ of the rule R_q using the product operator as

$$\mu_{\mathbf{A}_q}(\mathbf{x}_p) = \mu_{A_{q1}}(x_{p1}) \cdot \mu_{A_{q2}}(x_{p2}) \cdot ... \cdot \mu_{A_{qn}}(x_{pn}), \quad (2)$$

where $\mu_{A_{qi}}(\cdot)$ is the membership function of an antecedent set A_{qi}. We first calculate the confidence of the classification rule "$\mathbf{A}_q \Rightarrow$ Class h" in the field of data mining:

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p) \Big/ \sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p). \quad (3)$$

The consequent class C_q of the rule R_q is specified by identifying the class with the maximum confidence as

$$c(\mathbf{A}_q \Rightarrow \text{Class } C_q) = \max_{h=1,2,...,M} \{c(\mathbf{A}_q \Rightarrow \text{Class } h)\}. \quad (4)$$

Using the confidence measure, we specify the rule weight CF_q as

$$CF_q = c(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} c(\mathbf{A}_q \Rightarrow \text{Class } h). \quad (5)$$

If the rule weight CF_q is negative, we do not generate the rule "$\mathbf{A}_q \Rightarrow$ Class h".

Let S be a set of classification rules. A new pattern $\mathbf{x}_p$ is classified by a single winner rule R_w, which is chosen from the rule set S as

$$\mu_{\mathbf{A}_w}(\mathbf{x}_p) \cdot CF_w = \max\{\mu_{\mathbf{A}_q}(\mathbf{x}_p) \cdot CF_q \mid R_q \in S\}. \quad (6)$$

The winner rule R_w has the maximum product of the compatibility grade and the rule weight in S.

2.1 Fuzzy Rules

All attribute values are normalized into real numbers in the unit interval [0, 1]. As antecedent fuzzy sets, we use *"don't care"* and 14 homogeneous triangular fuzzy sets in Fig. 1 (see [2]).

2.2 Interval Rules

The whole domain interval is divided into K intervals. To specify $(K-1)$ cutting points for each attribute, we use an optimal splitting method [3] based on the class entropy measure:

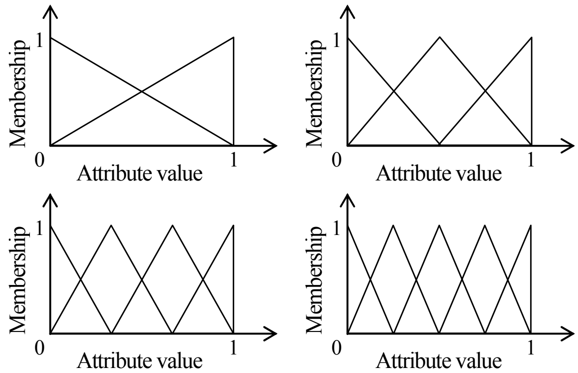


Figure 1: Four fuzzy partitions used in our computer simulations.

$$H(A_1, ..., A_K) = -\sum_{j=1}^{K} \frac{|D_j|}{|D|} \sum_{h=1}^{M} \left(\frac{|D_{jh}|}{|D_j|} \cdot \log_2 \frac{|D_{jh}|}{|D_j|} \right), \quad (7)$$

where $(A_1, ..., A_K)$ are K intervals generated by the discretization of an attribute, D_j is the set of training patterns in the interval A_j, and D_{jh} is the set of training patterns from Class h in D_j. We can efficiently find the optimal $(K-1)$ cutting points that minimize the class entropy measure in (7). In computational experiments, we use five partitions with K intervals where $K = 1,2,3,4,5$ as in Fig. 2 (see [4]).

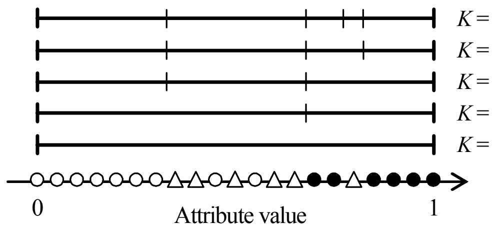


Figure 2: Five partitions with different granularities.

2.3 Fuzzified Interval Rules

We also examine the performance of fuzzy rule-based systems of the antecedent fuzzy sets in Fig. 3. The antecedent fuzzy sets of those rules are generated by fuzzifying intervals in Subsection 2.2. For more details on our fuzzification, see [5].

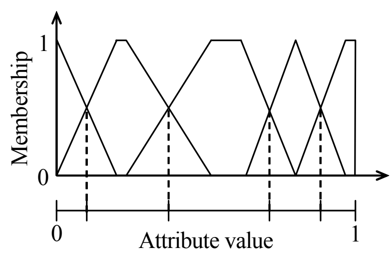


Figure 3: Fuzzy sets derived from intervals.

3 Heuristic Rule Extraction and Genetic Rule Selection

As in our former studies on the design of fuzzy rule-based classifiers [2], we handle knowledge extraction as the following three-objective rule selection problem:

$$\text{Maximize } f_1(S) \text{ and minimize } f_2(S) \text{ and } f_3(S), \quad (8)$$

where *S* is a subset of candidate rules, $f_1(S)$ is the number of correctly classified training patterns by *S*, $f_2(S)$ is the number of fuzzy rules in *S*, and $f_3(S)$ is the total rule length in *S*. The number of antecedent conditions of each fuzzy rule is referred to as the rule length in this paper. To solve (8), we apply a two-step method (i.e., heuristic rule extraction and genetic rule selection).

First, in heuristic rule extraction, a pre-specified number of candidate rules with the largest values of the following criterion are found for each class.

$$f_{\text{SLAVE}}(R_q) = s(\mathbf{A}_q \Rightarrow \text{Class } C_q) - \sum_{\substack{h=1 \\ h \neq C_q}}^{M} s(\mathbf{A}_q \Rightarrow \text{Class } h), \quad (9)$$

where $f_{\text{SLAVE}}(\cdot)$ is a modified version of a rule evaluation criterion in an iterative genetic learning algorithm called SLAVE [6], and $s(\cdot)$ is the support defined as follows:

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{1}{m} \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p). \quad (10)$$

For designing rule-based systems with high comprehensibility, only short rules are examined as candidate rules. This restriction on the rule length is consistent with the third objective (i.e., to minimize the total rule length) of our three-objective formulation in (8).

Second, in genetic rule selection, let us assume that *N* rules have been extracted as candidate rules using the SLAVE criterion (i.e., N/M rules for each class). A subset *S* of the *N* candidate rules is represented by a binary string of the length *N* as

$$S = s_1 s_2 \cdots s_N, \quad (11)$$

where $s_j = 1$ and $s_j = 0$ mean that the *j*-th candidate rule is included in *S* and excluded from *S*, respectively. Since rule sets are represented by binary strings, almost all multiobjective genetic algorithms are applicable. In this paper, we use the NSGA-II [7] because its search ability is high and its implementation is relatively easy. We incorporate two problem-specific heuristic tricks into the NSGA-II. One is biased mutation where a larger probability is assigned to the mutation from 1 to 0 than that from 0 to 1. This is for efficiently decreasing the number of rules. The other is the removal of unnecessary rules. Since we use the single winner-based method for classifying each pattern, some rules in *S* may be chosen as winner rules for no patterns. We can remove those rules without degrading the first objective with respect to the classification accuracy. At the same time, the second and third objectives with respect to the complexity are improved by removing unnecessary rules. Thus, we remove all rules that are not selected as winner rules for any training patterns from the rule set *S*. The removal of unnecessary rules is performed after the first objective is calculated for each rule set and before the second and third objectives are calculated.

4 Computational Experiments

4.1 Settings of computational experiments

We use six data sets in Table 1. These data sets are available from the UC Irvine machine learning repository. We do not use incomplete patterns with missing values in our computational experiments. All attributes are handled as continuous attributes in this paper.

We evaluate the performance of our EMO approach by ten independent executions (with different data partitions) of the whole ten-fold cross-validation (10CV) procedure (i.e., 10×10CV) in [3]. We extract 300 candidate rules for each class in the heuristic greedy manner using the SLAVE criterion (i.e., 300*M* candidate rules in total where *M* is the number of classes).

The NSGA-II [7] is applied to the extracted 300*M* candidate rules to find non-dominated rule sets with respect to the three objectives. Each of the obtained rule sets is evaluated by test data. We use the following parameter values in the NSGA-II:

Population size: 200 strings,
Crossover probability: 0.8 (uniform crossover),
Biased mutation probabilities:
$p_{\text{m}}(0 \rightarrow 1) = 1/300M$ and $p_{\text{m}}(1 \rightarrow 0) = 0.1$,
Stopping condition: 5000 generations.

During ten executions of the whole 10CV procedure, the NSGA-II is employed 100 times. We examine the average error rates only for combinations of the number of rules and the average rule length obtained from more than 30 (out of 100) runs. We also calculate the average error rates on training data and test data using the majority vote where all the obtained rule sets in each run except for too small ones are used for the classification of each pattern. Due to their poor performance, we do not use small rule sets with less rules than the number of classes.

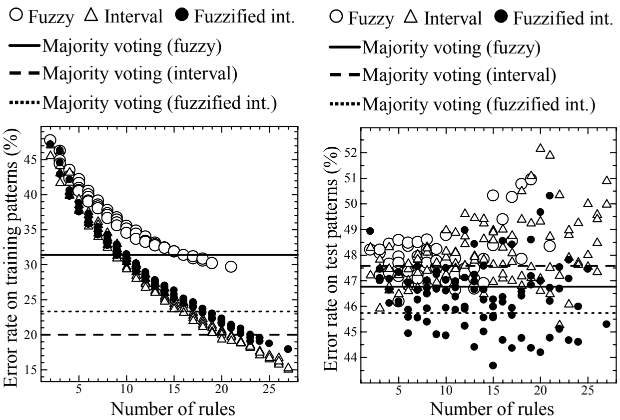
Table 1: Data sets used in our computational experiments.

Data set	Attributes	Patterns	Classes
Breast W	9	683*	2
Diabetes	8	768	2
Glass	9	214	6
Heart C	13	297*	5
Sonar	60	208	2
Wine	13	178	3

* Incomplete patterns with missing values are not included.

4.2 Experimental Results

We show the result on the Cleveland heart disease data set (i.e., Heart C in Table 1) in Fig. 4. A number of non-dominated rule sets with different accuracy and different complexity were obtained for each classification system. In the comparison between fuzzy rules and interval rules, we can see that interval rule-based classification systems can obtain high accuracy on training data, while high generalization ability on test data can be obtained by fuzzy rule-based classification systems from this figure. We can also see that fuzzified interval rule-based classification systems are comparable to interval rule-based classification systems on training data, while their generalization ability outperforms two other classification systems.



(a) Error rates on training patterns. (b) Error rates on test patterns.

Figure 4: Error rates for the Cleveland heart disease data set.

Table 2: Error rates on test data by majority voting. The best result for each data set is shown by boldface.

Data set	Fuzzy	Interval	Fuzzified int.
Breast W	3.75	4.06	**3.31**
Diabetes	25.73	25.27	**24.29**
Glass	39.31	**30.60**	32.67
Heart C	46.77	47.58	**45.74**
Sonar	**22.16**	25.58	23.59
Wine	**4.15**	5.00	4.78

Table 2 shows error rates on test data by majority voting for each data set. The results show that the generalization ability of the interval rule-based classification systems can be improved by their fuzzification for almost all data sets.

5 Summary

We generated multiple rule-based classification systems using an EMO algorithm with respect to the classification accuracy and the complexity, and compared the performance of fuzzy rule-based classification systems with that of interval rule-based classification systems on some benchmark data sets. Experimental results showed that interval rule-based classification systems can obtain high accuracy on training data, while high generalization ability on test data can be obtained by fuzzy rule-based classification systems. It was also shown that the generalization ability of the interval rule-based classification systems can be improved by their fuzzification.

References

[1] H. Roubos and M. Setnes, "Compact and transparent fuzzy models and classifiers through iterative complexity reduction," *IEEE Trans. on Fuzzy Systems*, vol. 9, no. 4, pp. 516-524, August 2001.

[2] H. Ishibuchi and T. Yamamoto, "Evolutionary multiobjective optimization for generating an ensemble of fuzzy rule-based classifiers," *Lecture Notes in Computer Science 2723: Genetic and Evolutionary Computation Conference - GECCO 2003*, pp. 1077-1088, Springer, Berlin, July 2003.

[3] T. Elomaa and J. Rousu, "General and efficient multisplitting of numerical attributes," *Machine Learning*, vol. 36, no. 3, pp. 201-244, September 1999.

[4] H. Ishibuchi and S. Namba, "Evolutionary multiobjective knowledge extraction for high-dimensional pattern classification problems," *Lecture Notes in Computer Science 3242: Parallel Problem Solving from Nature - PPSN VIII*, pp. 1123-1132, Springer, Berlin, October 2004.

[5] H. Ishibuchi and T. Yamamoto, "Performance evaluation of fuzzy partitions with different fuzzification grades," *Proc. of IEEE International Conference on Fuzzy Systems*, pp. 1198-1203, Hawaii, USA, May 2002.

[6] A. Gonzalez and R. Perez, "SLAVE: A genetic learning system based on an iterative approach," *IEEE Trans. on Fuzzy Systems*, vol. 7, no. 2, pp. 176-191, April 1999.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002.