Multiobjective Genetic Fuzzy Rule Selection with Fuzzy Relational Rules

Yusuke Nojima and Hisao Ishibuchi
Department of Computer Science and Intelligent Systems
Osaka Prefecture University
Sakai, Osaka, Japan
{nojima, hisaoi}@cs.osakafu-u.ac.jp

Abstract—Genetic fuzzy rule selection has been frequently used for fuzzy rule-based classifier design. A number of its variants have also been proposed in the literature. In many studies on genetic fuzzy rule selection, each antecedent condition in fuzzy rules is given for a single input variable such as “x₁ is small” and “x₂ is large”. As a result, each antecedent fuzzy set is defined on a single input variable. In this paper, we examine the use of fuzzy relational conditions with respect to the relation between two input variables such as “x₁ is approximately equal to x₂” and “x₂ is approximately larger than x₁”. Such a fuzzy relational condition is defined by a fuzzy set on a pair of input variables. We examine the effect of using fuzzy rules with relational conditions on the performance of fuzzy rule-based classifiers designed by multiobjective genetic fuzzy rule selection.

Keywords—Fuzzy relational rules, genetic fuzzy rule selection, multiobjective optimization, pattern classification.

I. INTRODUCTION

Fuzzy rule-based systems have been successfully used in various application fields due to their high interpretability and approximation ability [1]-[3]. Evolutionary computation has often been used for the design of fuzzy rule-based systems. This research area is referred to as genetic fuzzy systems or evolutionary fuzzy systems [4]-[10]. One advantage of using evolutionary computation is its ability to simultaneously handle different design criteria such as interpretability maximization and accuracy maximization in a framework of multiobjective optimization. Currently, interpretability-related issues have been studied very actively [11]-[13].

In modeling and classification problems, there often exist some fuzzy relations between input variables such as “x₁ is approximately larger than x₂” and “x₃ is roughly equal to x₄”. Fuzzy rules with fuzzy relations in their antecedent part are referred to as fuzzy relational rules. In the 1990s, the use of fuzzy relational rules was discussed by Yager [14], [15].

With a few exceptions [16], [17], the use of fuzzy relational rules has not been examined in the field of genetic fuzzy systems. In this paper, we use fuzzy relational rules as candidate rules in multiobjective genetic fuzzy rule selection for fuzzy rule-based classifier design. Through computational experiments, we show that the use of fuzzy relational rules not only decreases the number of fuzzy rules but also improves the classification accuracy of designed fuzzy rule-based classifiers by multiobjective genetic fuzzy rule selection.

This paper is organized as follows. In Section II, we show standard fuzzy if-then rules and fuzzy relational rules for pattern classification problems. In Section III, we briefly explain multiobjective genetic fuzzy rule selection. The effectiveness of using fuzzy relational rules is discussed through computational experiments in Section IV. Finally we conclude this paper in Section V.

II. FUZZY IF-THEN RULES

A. Standard Fuzzy If-then Rules

We assume an M-class n-dimensional pattern classification problem with m training patterns \( x_p = (x_{p1}, ..., x_{pn}), p = 1, 2, ..., m \). We use fuzzy rules of the following type [18]:

\[
\text{Rule } R_q : \text{If } x_{1} \text{ is } A_{q1} \text{ and } ... \text{ and } x_n \text{ is } A_{qn} \text{ then Class } C_q \text{ with } CF_q, \tag{1}
\]

where \( R_q \) is the label of the qth fuzzy rule, \( x = (x_1, ..., x_n) \) is an n-dimensional pattern vector, \( A_{qi} \) is an antecedent fuzzy set \( i = 1, 2, ..., n \), \( C_q \) is a class label, and \( CF_q \) is a real number in the unit interval \([0, 1]\) called a rule weight.

After normalizing each attribute value \( x_{pi} \) into a real number in the unit interval \([0, 1]\) for \( i = 1, 2, ..., n \) and \( p = 1, 2, ..., m \), we use 14 fuzzy sets in Fig. 1 and “don’t care” for all input variables (i.e., for all attributes). The “don’t care” condition is represented by the unit interval \([0, 1]\), which is always fully compatible with any normalized attribute value \( x_{pi} \). The use of “don’t care” conditions enables us to simplify a fuzzy rule. This also plays an important role in selecting attributes. The number of antecedent conditions (except for “don’t care” conditions) is often referred to as the rule length.

![Figure 1. Fourteen antecedent fuzzy sets for standard fuzzy if-then rules.](image-url)
B. Fuzzy Relational Rules

It is often the case that a number of standard fuzzy if-then rules are needed to represent a simple relation between input variables. For example, let us consider a pattern classification problem in Fig. 2. From Class 1 patterns in Fig. 2, we can generate the following five standard fuzzy if-then rules:

\[
\begin{align*}
\text{If } x_a \text{ is } S \text{ and } x_b \text{ is } S \text{ then Class } 1, \\
\text{If } x_a \text{ is } MS \text{ and } x_b \text{ is } MS \text{ then Class } 1, \\
\text{If } x_a \text{ is } M \text{ and } x_b \text{ is } M \text{ then Class } 1, \\
\text{If } x_a \text{ is } ML \text{ and } x_b \text{ is } ML \text{ then Class } 1, \\
\text{If } x_a \text{ is } L \text{ and } x_b \text{ is } L \text{ then Class } 1.
\end{align*}
\]

These five fuzzy if-then rules can be summarized by the following fuzzy relational rule:

\[
\text{If } x_a \text{ is } \text{approximately equal to } x_b \text{ then Class } 1.
\]

In Fig. 2, we can also generate a number of standard fuzzy if-then rules for Class 2 and Class 3. However, those standard fuzzy if-then rules can be summarized by the following two fuzzy relational rules:

\[
\begin{align*}
\text{If } x_a \text{ is } \text{approximately smaller than } x_b \text{ then Class } 2, \\
\text{If } x_a \text{ is } \text{approximately larger than } x_b \text{ then Class } 3.
\end{align*}
\]

Of course, we can use “If } x_a \text{ is } \text{approximately larger than } x_b\text{” instead of “If } x_a \text{ is } \text{approximately smaller than } x_b\text{” for Class 2. From these discussions, we can see that the pattern classification problem in Fig. 2 can be handled by the three fuzzy relational rules without using the 5 x 5 fuzzy grid. This example suggests the potential ability of fuzzy relational rules to decrease the complexity of fuzzy rule-based classifiers.

III. MULTIOBJECTIVE GENETIC FUZZY RULE SELECTION

Genetic fuzzy rule selection is a two-stage fuzzy classifier design method [19]-[21]. In the first stage, a prespecified number of candidate rules are generated from numerical data in a heuristic manner. In the second stage, subsets of candidate rules are optimized by evolutionary computation.

A. First Stage: Candidate Rule Generation

With respect to standard fuzzy if-then rules, the total number of possible rules is 15^n for our n-dimensional problem since we have 14 fuzzy sets in Fig. 1 and don’t care for each of the n attributes. If n is large, it needs huge computational time to generate all of those fuzzy rules. Moreover, long fuzzy rules with many antecedent conditions are not understandable. Thus the use of long fuzzy rules deteriorates the interpretability of fuzzy rule-based classifiers. To avoid such an undesirable
In this section, we examine only short fuzzy rules of length $L_{\text{max}}$ or less (e.g., $L_{\text{max}} = 3$). This is to find a small number of short fuzzy rules with high interpretability. With respect to fuzzy relational rules, we examine all combinations of two attributes and six relations (i.e., $6n(n-1)$).

Let us denote the antecedent part of each (standard and relational) fuzzy rule $R_q$ by $A_q$. The consequent class $C_q$ and the rule weight $CF_q$ of $R_q$ are specified using the given training patterns in the following heuristic manner. First we calculate the compatibility grade of each pattern $x_p$ with $A_q$. If $A_q$ includes two or more conditions, the product operation is used. Next we calculate the confidence of each class $h$ ($h = 1, 2, \ldots, M$) for the antecedent part $A_q$ as follows [22]:

$$c(A_q \Rightarrow \text{Class } h) = \frac{\sum_{p \in \text{Class } h} \mu_{A_q}(x_p)}{\sum_{p=1}^{m} \mu_{A_q}(x_p)} . \quad (8)$$

The consequent class $C_q$ is specified by identifying the class with the maximum confidence:

$$c(A_q \Rightarrow \text{Class } C_q) = \max_{h=1,2,\ldots,M} \{c(A_q \Rightarrow \text{Class } h)\} . \quad (9)$$

Different specifications of the rule weight $CF_q$ have been proposed and examined in the literature. We use the following specification because good results were reported by this specification in the literature [23], [24]:

$$CF_q = c(A_q \Rightarrow \text{Class } C_q) - \sum_{h=1}^{M} c(A_q \Rightarrow \text{Class } h) . \quad (10)$$

In this manner, we can generate a number of fuzzy rules. Among the generated fuzzy rules, we choose only promising rules by a heuristic rule evaluation criterion as candidate rules. In the field of data mining, two rule evaluation criteria (i.e., confidence and support) have been frequently used. We have already shown the fuzzy version of the confidence criterion in (3). The support of the fuzzy rule is calculated as follows [24]:

$$s(A_q \Rightarrow \text{Class } h) = \frac{\sum_{x_p \in \text{Class } h} \mu_{A_q}(x_p)}{m} . \quad (11)$$

We use the product of the confidence and the support to evaluate each rule. In our computational experiments, the upper limit of candidate rules was specified as 300 for each class (i.e., 300 rules per class).

### B. Second Stage: Combinatorial Optimization by NSGA-II

In the second stage of our multiobjective genetic fuzzy rule selection, NSGA-II [25] is used to search for Pareto-optimal subsets of the candidate rules. A subset of the candidate rules is coded as a binary string in NSGA-II.

Let $N$ be the number of the candidate rules. An arbitrary subset $S$ of the candidate rules is represented by a binary string $S = s_1 s_2 s_3 \cdots s_N$, where $s_i = 1$ and $s_i = 0$ mean that the $i$th candidate rule is included in and excluded from the rule set $S$, respectively. The framework of the second stage is written as follows:

Step 1: Randomly generate a prespecified number of binary strings as an initial population.

Step 2: Generate a prespecified number of offspring strings from the current population using binary tournament selection, uniform crossover and biased mutation.

Step 3: Combine the current population and the offspring population into a single merged population.

Step 4: If a termination condition is satisfied, output the combined population. Otherwise, construct the next population from the combined population and go back to Step 2.

In Step 2, the biased mutation changes 0 to 1 with a small probability and 1 to 0 with a large probability to decrease the number of 1’s (i.e., selected rules) in the offspring.

We use the following two objectives to search for Pareto optimal rule sets with respect to their accuracy and simplicity:

$\text{f}_1(S)$: The number of correctly classified training patterns by $S$, $S_{\text{RCFR}}$.

$\text{f}_2(S)$: The number of selected fuzzy rules in $S$, where $\text{f}_1(S)$ is to be maximized while $\text{f}_2(S)$ is to be minimized.

To calculate the first objective $\text{f}_1(S)$, we classify training patterns by $S$. A single winner rule $R_w$ is chosen from the rule set $S$ as follows:

$$R_w = \arg \max \{ \mu_{A_q}(x_p) \cdot CF_q \mid R_q \in S \} . \quad (12)$$

As shown in (12), the winner rule $R_w$ has the maximum product of the compatibility grade and the rule weight in $S$. The classification of $x_p$ is rejected when no rules are compatible with $x_p$ (which is counted as an error in our computational experiments). The classification of $x_p$ is also rejected when multiple rules with different consequent classes have the same maximum product in (12).

The second objective $\text{f}_2(S)$ is calculated by just counting the number of 1’s (i.e., the number of the selected rules) in $S$. Since we use the single winner-based method, only a single rule is responsible for the classification of each training pattern. As a result, some rules in $S$ may be used for the classification of no training patterns. Those rules in $S$ are not necessary for the classification of any training patterns. From each string, we remove all the unnecessary rules that are responsible for the classification of no training patterns before calculating the $\text{f}_2(S)$.

As shown in Step 3 and Step 4, NSGA-II uses a $(\mu + \lambda)$-ES style generation update mechanism where $\mu = \lambda$. That is, the number of offspring strings is the same as the population size.

### IV. Computational Experiments

In this session, we examine the quality of fuzzy relational rules by using the confidence-support plots. We also demonstrate the effectiveness of using fuzzy relational rules in...
our multiobjective genetic fuzzy rule selection for pattern classification problems.

A. Experimental Setting

We applied our multiobjective genetic fuzzy rule selection to three benchmark data sets in Table I which are available from the Keel data repository (http://keel.es) [26]. We used tenfold cross validation (10CV) with five different data partitions (i.e., 5 × 10CV). Since only one data partition for 10CV is provided in the Keel data repository, we randomly divided each data for other four data partitions. In this manner, we performed the 10CV five times.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Number of patterns</th>
<th>Number of attributes</th>
<th>Number of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>214</td>
<td>9</td>
<td>6</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>351</td>
<td>34</td>
<td>2</td>
</tr>
<tr>
<td>Sonar</td>
<td>208</td>
<td>60</td>
<td>2</td>
</tr>
</tbody>
</table>

Table I. Data Sets Used in Our Computational Experiments. In real-world data sets, attributes do not always have the same unit. For example, one attribute may represent the length (e.g., [cm]) while another attribute may represent the weight (e.g., [g]). In this case, fuzzy relational conditions (e.g., the weight [g] is approximately equal to the length [cm]) are meaningless. Thus we first checked the characteristics of each attribute of various data sets. Then we decided to use only the three data sets in Table I. Even in the three data sets, some attributes are not appropriate for relational conditions. For example, Glass data has nine attributes. The first attribute represents the refractive index. The other eight attributes represent weight percentages. Thus, we did not use the first attribute to generating fuzzy relational rules.

In our computational experiments, we used the following parameter specifications in our multiobjective genetic fuzzy rule selection:

- Upper limit of the number of candidate rules: 300 per class,
- Population size: 200,
- Termination condition: 2000 generations,
- Crossover probability: 0.9,
- Biased mutation probabilities: 0.05 for \( s_i = 1 \), 1/N for \( s_i = 0 \).

We examined four cases: Case (S) used only standard fuzzy if-then rules as candidate rules. This is the conventional version of our multiobjective genetic fuzzy rule selection. Case (R) used only fuzzy relational rules as candidate rules. Case (S, R) used both standard fuzzy rules and fuzzy relational rules. After combining those two rule sets, we chose 300 rules per class based on the product of the confidence and the support. Finally, Case (S+R) used both types of rules but they were simply combined after individually generating each type of rules. That is, the total number of candidate rules was more than 300 per class. The upper limit was 600 rules per class in Case (S+R).

B. Candidate Rule Analysis

We first examined how informative each fuzzy relational rule is. Figure 3 shows the confidence-support plots of extracted rules for the Glass data set. Each symbol represents a single rule. As we mentioned earlier, the confidence and the support are often used to evaluate the quality of rules. In Fig. 3, standard fuzzy rules were generated from training data in the first run of 10CV in Case (S). Fuzzy relational rules were also generated from the same training data in Case (R). The number of generated fuzzy relational rules was very small. This is because most fuzzy relational rules had negative rule weights. (We do not use any fuzzy rules with negative rule weights).

For Class 3 in Fig. 3 (c), there was no fuzzy relational rule. For some classes, fuzzy relational rules seem to be more useful than standard fuzzy rules. For example, one fuzzy relational rule had the highest support value for Class 5 in Fig. 3 (e). From Fig. 3, we can see that a small number of standard fuzzy rules were replaced with fuzzy relational rules in Case (S, R).

Table II shows the number of extracted candidate rules for the Glass data set. The first and second rows are the numbers of standard fuzzy rules and fuzzy relational rules, respectively. The other rows show the number of rules for each fuzzy relational condition.

Figure 3. Distributions of candidate rules for Glass data.
In Table II, the number of relational rules in Case (S, R) is almost the same as that in Case (R). This means that almost all of the extracted relational rules have large product values of the confidence and the support if compared with the extracted standard fuzzy rules.

Figure 4 shows the distributions of standard fuzzy rules and fuzzy relational rules in the confidence-support plots for the Ionosphere data set. For Class 1 in Fig. 4 (a), relational rules have higher support values while standard rules have higher confidence values. For Class 2 in Fig. 4 (b), relational rules dominate standard rules in the sense of two-objective maximization with respect to the support and the confidence. However, the majority of extracted rules are standard rules in Table III. An interesting observation is that no relational rules with "twice" were extracted in Table III. These conditions may be too specific for the Ionosphere data set.

We can obtain almost the same observations from Fig. 5 for the Sonar data set. Fuzzy relational rules have higher support values. This means that each relational rule covers a larger number of patterns. Moreover, Table IV shows that most of the extracted relational rules have approximately larger and roughly larger conditions.

From Figs. 3, 4, and 5, we can say that the quality (i.e., confidence and support) of fuzzy relational rules strongly depends on the data set. In some data sets, fuzzy relational rules cover a larger number of patterns. From Tables II-IV, we can say that the appropriate relational conditions strongly depend on the data set.

C. Tradeoff Analysis of Obtained Classifiers

Figures 6, 7, and 8 show the average classification rates over 50 runs in the 5x10CV. Each circle represents the average value of classifiers with the same number of rules. It should be noted that classifiers with many rules were not always obtained. A circle with dot represents that a classifier with the same number of rules was obtained in more than 25 runs. It means that the average value is somewhat reliable.

From Fig. 6, we can see that the classification rates in Case (R) were very bad. Let us focus on the results of the obtained classifiers with less than 15 rules. An interesting observation is that some classifiers with better test data accuracy were obtained in Case (S+R) where only a few fuzzy relational rules were added to standard fuzzy rules as candidate rules.

In Fig. 7, the classifiers obtained in Case (R) have good test data accuracy. This means that the use of fuzzy relational rules is effective for the Ionosphere data set.
Under the condition where the number of rules is less than ten, the obtained classifiers in Case (S, R) and Case (S+R) are slightly better than those in Case (S) with respect to test data accuracy. This is a positive effect of fuzzy relational rules.

From Fig. 8, we can see that accurate classifiers were not obtained in Case (R). When we used fuzzy relational rules together with standard rules in Case (S, R) and Case (S+R), classifiers with many rules were obtained in some runs. Those classifiers had high training data accuracy but poor test data accuracy (i.e., over-fitted to the training data).

A number of non-dominated fuzzy classifiers were obtained in each run in our computational experiments. In Tables V-VII, we compare the four cases with each other using the fuzzy classifier with the best training data accuracy among a number of obtained non-dominated classifiers in each run. Each table includes average results over 50 runs for the training data accuracy, the test data accuracy, the number of rules, the number of standard rules, and the number of relational rules. The best results are highlighted by bold face. From these tables, we can observe that the classifiers obtained in Case (S, R) and Case (S+R) have higher training data accuracy than those in Case (S) for all the three data sets.

In Table V for the Glass data set, higher test data accuracy was obtained from Case (S) than Case (S, R). This observation suggests that some good standard rules in Case (S) were
replaced with fuzzy relational rules in Case (S, R) in the candidate rule selection phase. The best test data accuracy was obtained in Case (S+R) using all standard candidate rules in Case (S) and all relational candidate rules in Case (R).

In Table VI for the Ionosphere data set, we can observe clear improvement by using fuzzy relational rules (i.e., from Case (S) to Case (S, R) and Case (S+R). In Case (S+R) in Table VI, about a half of the obtained rules are relational rules. Moreover, the highest test data accuracy was obtained from Case (R). These observations suggest that fuzzy relational rules are useful in the Ionosphere data set.

In Table VII for the Sonar data set, the training data accuracy was improved by using fuzzy relational rules (i.e., from Case (S) to Case (S, R) and Case (S+R). However, the use of fuzzy relational rules deteriorated the test data accuracy.

<table>
<thead>
<tr>
<th>Case</th>
<th>Training data accuracy</th>
<th>Test data accuracy</th>
<th>Number of rules</th>
<th>Number of Standard rules</th>
<th>Number of relational rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>78.70</td>
<td>62.08</td>
<td>13.48</td>
<td>13.48</td>
<td>0.00</td>
</tr>
<tr>
<td>R</td>
<td>53.96</td>
<td>48.32</td>
<td><strong>10.64</strong></td>
<td>0.00</td>
<td><strong>10.64</strong></td>
</tr>
<tr>
<td>S, R</td>
<td>79.80</td>
<td>61.77</td>
<td>15.12</td>
<td>12.76</td>
<td>2.36</td>
</tr>
<tr>
<td>S+R</td>
<td><strong>79.98</strong></td>
<td><strong>62.40</strong></td>
<td>14.46</td>
<td>11.54</td>
<td><strong>2.92</strong></td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, we defined three simple fuzzy relational conditions for pattern classification. Through computational experiments, we examined the usefulness of generated fuzzy relational rules as candidate rules in our multiobjective genetic fuzzy rule selection. We also analyzed the tradeoff relationship between the accuracy and complexity of the obtained classifiers. Finally we compared four settings of candidate rule generations with each other using the fuzzy classifier with the best training data accuracy among a number of obtained non-dominated classifiers in each run. It was clearly demonstrated that the use of fuzzy relational rules potentially has a positive effect on both the training data accuracy and the test data accuracy of fuzzy rule-based classifiers.

A number of research issues still remain for future work. One issue is the parameter specifications in our computational experiments. Especially, the number of candidate rules and the rule selection criteria should be examined carefully. Pareto optimality with respect to the confidence and the support [27] [28] can be used for choosing candidate rules, which may efficiently reduce the number of candidate rules. The interpretability of fuzzy relational rules should also be discussed from the viewpoint of human users.

<table>
<thead>
<tr>
<th>Case</th>
<th>Training data accuracy</th>
<th>Test data accuracy</th>
<th>Number of rules</th>
<th>Number of Standard rules</th>
<th>Number of relational rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>95.37</td>
<td>87.37</td>
<td>9.46</td>
<td>9.46</td>
<td>0.00</td>
</tr>
<tr>
<td>R</td>
<td>94.96</td>
<td>90.60</td>
<td><strong>7.26</strong></td>
<td>0.00</td>
<td><strong>7.26</strong></td>
</tr>
<tr>
<td>(S, R)</td>
<td><strong>95.54</strong></td>
<td>88.69</td>
<td>9.64</td>
<td>8.71</td>
<td><strong>0.93</strong></td>
</tr>
<tr>
<td>(S+R)</td>
<td><strong>96.28</strong></td>
<td><strong>89.68</strong></td>
<td>9.68</td>
<td><strong>5.10</strong></td>
<td><strong>4.58</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>Training data accuracy</th>
<th>Test data accuracy</th>
<th>Number of rules</th>
<th>Number of Standard rules</th>
<th>Number of relational rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>85.90</td>
<td><strong>77.17</strong></td>
<td>6.40</td>
<td>6.40</td>
<td>0.00</td>
</tr>
<tr>
<td>R</td>
<td>74.63</td>
<td>60.47</td>
<td><strong>4.00</strong></td>
<td>0.00</td>
<td><strong>4.00</strong></td>
</tr>
<tr>
<td>(S, R)</td>
<td><strong>86.38</strong></td>
<td>75.80</td>
<td>6.82</td>
<td>6.17</td>
<td><strong>0.65</strong></td>
</tr>
<tr>
<td>(S+R)</td>
<td><strong>86.10</strong></td>
<td><strong>75.38</strong></td>
<td>5.40</td>
<td><strong>3.64</strong></td>
<td><strong>1.76</strong></td>
</tr>
</tbody>
</table>

REFERENCES


