Use of Pareto-Optimal and Near Pareto-Optimal Candidate Rules in Genetic Fuzzy Rule Selection

Hisao Ishibuchi, Isao Kuwajima, and Yusuke Nojima

Department of Computer Science and Intelligent Systems, Osaka Prefecture University,
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
{hisaoi@,kuwajima@ci.,nojima@}cs.osakafu-u.ac.jp
http://www.ie.osakafu-u.ac.jp/~hisaoi/ci_lab_e

Abstract. Genetic fuzzy rule selection is an effective approach to the design of accurate and interpretable fuzzy rule-based classifiers. It tries to minimize the complexity of fuzzy rule-based classifiers while maximizing their accuracy by selecting only a small number of fuzzy rules from a large number of candidate rules. One important issue in genetic fuzzy rule selection is the prescreening of candidate rules. If we use too many candidate rules, a large computation load is required to search for good rule sets. On the other hand, good rule sets will not be found when promising fuzzy rules are not included in the candidate rule set. It is essential for the success of genetic fuzzy rule selection to use a tractable number of promising fuzzy rules as candidate rules. In this paper, we propose an idea of using Pareto-optimal and near Pareto-optimal fuzzy rules as candidate rules in genetic fuzzy rule selection. Pareto-optimality is defined by two well-known data mining criteria: support and confidence. To extract not only Pareto-optimal but also near Pareto-optimal fuzzy rules, we modify Pareto dominance using a dominance margin ε. Through computational experiments, we examine the effect of the proposed idea on multiobjective genetic fuzzy rule selection.

1 Introduction

The main advantage of fuzzy rule-based systems over other nonlinear systems such as neural networks is their linguistic interpretability [2], [3], [12]. Human users can understand fuzzy rule-based systems through linguistic interpretation of fuzzy rules. In this sense, fuzzy rule-based systems are viewed as transparent models (i.e., white-box models) while other nonlinear systems are usually black-box models. In addition to the linguistic interpretability of each fuzzy rule, various aspects are related to the interpretability of fuzzy rule-based systems (e.g., the number of fuzzy rules and the number of antecedent conditions of each fuzzy rule). Genetic fuzzy rule selection was proposed in [13] for the design of accurate and interpretable fuzzy rule-based classifiers by minimizing the number of fuzzy rules while maximizing their accuracy. A small number of fuzzy rules were selected from a large number of candidate rules to construct an accurate and interpretable fuzzy rule-based classifier. A standard single-objective genetic algorithm (SOGA) was used to optimize a weighted sum fitness function defined by an accuracy measure and a complexity measure. Genetic fuzzy rule selection was generalized to two-objective rule selection in [10] where a multiobjective genetic algorithm (MOGA) was used to search for non-dominated fuzzy rule

P. Melin et al. (Eds.): Anal. and Des. of Intel. Sys. using SC Tech., ASC 41, pp. 387–396, 2007.
springerlink.com

sets (i.e., non-dominated fuzzy rule-based classifiers) with respect to the accuracy and complexity measures. It was further generalized to three-objective rule selection in [11] by introducing the total number of antecedent conditions as an additional complexity measure. Currently multiobjective design of fuzzy rule-based systems is an active research area in the field of fuzzy systems [9], [14], [19], [20].

In the field of data mining, MOGAs were used to search for non-dominated rules with respect to well-known rule evaluation criteria: support and confidence. Such an MOGA-based data mining approach was first proposed in [5]. Recently a similar approach was used for multiobjective genetic fuzzy data mining [15].

One important issue in genetic fuzzy rule selection is the prescreening of candidate rules. Let N be the number of candidate rules. Any subset of the N candidate rules is represented by a binary string of length N. Thus the size of the search space is 2^N. When we have too many candidate rules (i.e., when N is too large), it is very difficult to efficiently search for good rule sets. A large computation load is required to find good rule sets in the search space of size 2^N. On the other hand, genetic fuzzy rule selection is not likely to find good rule sets when N is too small. In the application of genetic fuzzy rule selection to low-dimensional pattern classification problems with only a few attributes, we can examine all combinations of antecedent fuzzy sets to generate fuzzy rules. All the generated fuzzy rules can be used as candidate rules in genetic fuzzy rule selection. It is, however, impractical to use all the generated fuzzy rules as candidate rules for high-dimensional pattern classification problems with many attributes because the total number of possible fuzzy rules exponentially increases with the number of attributes. Thus we need a heuristic rule evaluation criterion for the prescreening of candidate rules in genetic fuzzy rule selection in its application to high-dimensional pattern classification problems [14]. Whereas various rule evaluation criteria such as confidence, support and their combinations are applicable, it is not easy to choose a single criterion because their effectiveness is problem-dependent. It is not easy to appropriately specify parameter values (e.g., the minimum support and the minimum confidence) involved in each criterion, either.

In this paper, we propose an idea of using Pareto-optimal and near Pareto-optimal rules with respect to support and confidence as candidate rules in genetic fuzzy rule selection. We modify Pareto dominance by introducing a dominance margin ε in the same manner as [18] to extract not only Pareto-optimal rules but also near Pareto-optimal rules. A similar modification method was also used to improve the performance of MOGAs under the name of ε-dominance [7], [16].

This paper is organized as follows. First we explain fuzzy rule-based classifiers in Section 2. Next we explain genetic fuzzy rule selection in Section 3. Then we examine the effect of using Pareto-optimal and near Pareto-optimal fuzzy rules as candidate rules on multiobjective genetic fuzzy rule selection in Section 4. Finally we conclude this paper in Section 5.

2 Fuzzy Rule-Based Classifiers

Let us assume that we have m training patterns $\mathbf{x}_p = (x_{p1}, x_{p2}, ..., x_{pn})$, $p = 1, 2, ..., m$ from M classes in an n-dimensional continuous pattern space $[0, 1]^n$ where x_{pi} is the attribute value of the p-th training pattern for the i-th attribute. For our pattern classification problem, we use fuzzy rules of the following form:

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

where R_q is the label of the q-th fuzzy rule, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is an n-dimensional pattern vector, A_{qi} is an antecedent fuzzy set, C_q is a class label, and CF_q is a rule weight. We denote R_q in (1) as "$\mathbf{A}_q \Rightarrow$ Class C_q" where $\mathbf{A}_q = (A_{q1}, A_{q2}, \ldots, A_{qn})$.

Since we usually have no *a priori* information about an appropriate granularity of the fuzzy discretization for each attribute, we simultaneously use multiple fuzzy partitions with different granularities to extract candidate fuzzy rules. In computational experiments, we use four fuzzy partitions with triangular fuzzy sets in Fig. 1. In addition to the 14 fuzzy sets in Fig. 1, we also use the domain interval [0, 1] itself as an antecedent fuzzy set in order to represent a *don't care* condition.

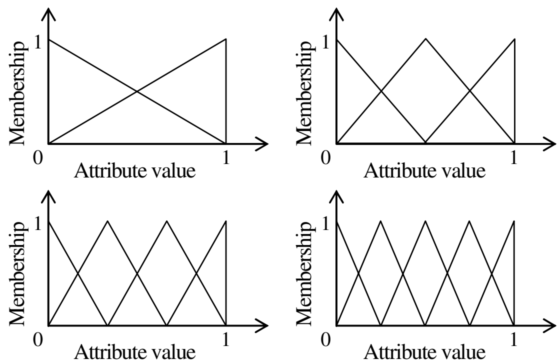


Fig. 1. Antecedent fuzzy sets used in computational experiments

Since we use the 15 antecedent fuzzy sets for each of the n attributes, the total number of combinations of the antecedent fuzzy sets is 15^n. Each combination is used in the antecedent part of the fuzzy rule in (1). The consequent class and the rule weight of each fuzzy rule can be easily specified from compatible training patterns in a heuristic manner [12]. This means that we can easily generate a large number of fuzzy rules by specifying the consequent class and the rule weight for each of the 15^n combinations of the antecedent fuzzy sets. It is, however, very difficult for human users to manually examine such a large number of fuzzy rules. It is also very difficult for human users to intuitively understand long fuzzy rules with many antecedent conditions. Thus we examine only short fuzzy rules of length $L_{\max}$ or less (e.g., $L_{\max} = 3$). This restriction on the rule length is to find fuzzy rule-based classifiers with high interpretability.

Let S be a set of fuzzy rules of the form in (1). That is, S is a fuzzy rule-based classifier. When an input pattern $\mathbf{x}_p$ is presented to S, $\mathbf{x}_p$ is classified by a single winner rule that has the maximum product of the compatibility grade and the rule weight (see [12] for various fuzzy reasoning methods for classification problems). In this paper, we use the product operator to calculate the compatibility grade.

3 Genetic Fuzzy Rule Selection

Genetic fuzzy rule selection is a two-step approach to the design of fuzzy rule-based classifiers. In the first phase, a large number of fuzzy rules are generated as candidate rules. A heuristic rule evaluation criterion is usually used for the prescreening of candidate rules [9], [11], [12], [14]. In the second phase, a genetic algorithm is used to select a small number of candidate rules.

In the field of data mining [1], two rule evaluation criteria (confidence and support) have been often used to evaluate an association rule. Fuzzy versions of these two criteria can be written as follows [12]:

$$c(\mathbf{A}_q \Rightarrow \text{Class } h) = \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p) \Bigg/ \sum_{p=1}^{m} \mu_{\mathbf{A}_q}(\mathbf{x}_p)\,, \tag{2}$$

$$s(\mathbf{A}_q \Rightarrow \text{Class } h) = \frac{1}{m} \sum_{\mathbf{x}_p \in \text{Class } h} \mu_{\mathbf{A}_q}(\mathbf{x}_p)\,, \tag{3}$$

where $c(\cdot)$ and $s(\cdot)$ are the confidence and the support of a fuzzy rule, respectively.

Association rule mining techniques extract all rules that satisfy the prespecified minimum confidence and support [1]. In computational experiments, we extract all fuzzy rules of length three or less that satisfy this condition. The extracted fuzzy rules are used as candidate rules in genetic fuzzy rule selection.

Let N be the number of candidate rules. A rule set S, which is a subset of the N candidate rules, is represented by a binary string of length N. Our fuzzy rule selection problem is formulated as follows [11]:

$$\text{Maximize } f_1(S)\text{, and minimize } f_2(S) \text{ and } f_3(S)\,, \tag{4}$$

where $f_1(S)$ 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 number of antecedent conditions (i.e., total rule length) in S. It should be noted that each fuzzy rule has a different number of antecedent conditions. This is because we use *don't care* as a special antecedent fuzzy set, which is not counted in the number of antecedent conditions.

In this section, we use a standard single-objective genetic algorithm (SOGA) to maximize the following weighted sum fitness function:

$$\mathit{fitness}(S) = w_1 \cdot f_1(S) - w_2 \cdot f_2(S) - w_3 \cdot f_3(S)\,, \tag{5}$$

where w_i is a non-negative weight for the i-th objective. We use the $(\mu+\lambda)$-ES generation update mechanism with $\mu=\lambda$ in our SOGA.

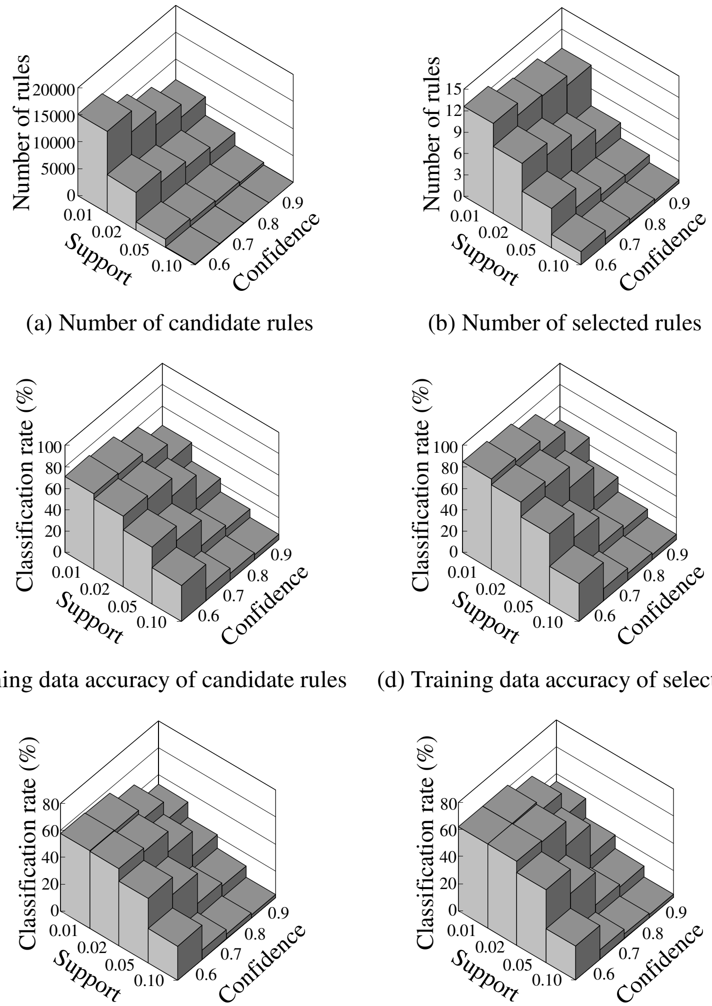
We applied genetic fuzzy rule selection to the glass data set (a nine-dimensional patterns classification problem with 214 patterns from six classes) in the UCI machine learning repository using the following parameter specifications:

Minimum confidence: 0.6, 0.7, 0.8, 0.9,
Minimum support: 0.01, 0.02, 0.05, 0.10,
Weight vector in the fitness function in (5): (10, 1, 1),
Population size: 200 (i.e., $\mu = \lambda = 200$),
Crossover probability: 0.9 (uniform crossover),
Mutation probability: 0.05 ($1 \rightarrow 0$) and $1/N$ ($0 \rightarrow 1$),
Termination condition: 1000 generations.

We examined the 4×4 combinations of the minimum confidence and support. For each combination, all the extracted fuzzy rules were used as candidate rules in the second phase where SOGA searched for the optimal rule set from the candidate rules. In our SOGA, we used biased mutation where a larger probability (i.e., 0.05) was assigned to the mutation from 1 to 0 than that from 0 to 1 for efficiently decreasing the number of fuzzy rules in each rule set. We removed unnecessary fuzzy rules from each string after mutation. That is, we removed all fuzzy rules that were chosen as winner rules for no training patterns. We performed five independent runs of the two-fold cross-validation procedure (i.e., 5×2CV).

Experimental results are summarized in Fig. 2. Fig. 2 (a) shows the number of extracted candidate rules. Their classification rates on training and test patterns are shown in Fig. 2 (c) and Fig. 2 (e), respectively. On the other hand, experimental results after genetic rule selection are shown in the right plots of Fig. 2. Only a small number of fuzzy rules were selected in Fig. 2 (b) from thousands of candidate rules in Fig. 2 (a). Training data accuracy was improved in many cases in Fig. 2 (d) by genetic rule selection from Fig. 2 (c). Test data accuracy was also improved in many cases in Fig. 2 (f) from Fig. 2 (e).

From Fig. 2 (c) and Fig. 2 (e), we can see that the accuracy of candidate rules strongly depended on the specification of the minimum confidence and support (see [4] for the learning of these parameter values). When both the minimum confidence and support were small, a larger number of candidate rules were extracted. For example, about 15000 candidate rules were generated in Fig. 2 (a) in the case of the minimum confidence 0.6 and the minimum support 0.01. Among those 15000 candidate rules, only 13 fuzzy rules were selected in Fig. 2 (b) on average. The average classification rates were improved by genetic fuzzy rule selection for both training and test data in Fig. 2. These observations demonstrate the effectiveness of genetic fuzzy rule selection. One difficulty of genetic fuzzy rule selection with a large number of candidate rules is its large computation load. Actually, our SOGA spent about one hour using a PC with Xeon 3.6 GHz CPU in the case of the minimum confidence 0.6 and the minimum support 0.01. The computation load can be significantly decreased by decreasing the number of candidate rules (i.e., by increasing the minimum confidence and support). High classification rates, however, were not obtained when the number of candidate rules was small in Fig. 2. In the next section, we discuss how we can decrease the number of candidate rules without severely degrading the accuracy of selected rules.



(a) Number of candidate rules (b) Number of selected rules

(c) Training data accuracy of candidate rules (d) Training data accuracy of selected rules

(e) Test data accuracy of candidate rules (f) Test data accuracy of selected rules

Fig. 2. Experimental results using SOGA for the glass data

4 Use of Pareto-Optimal and Near Pareto-Optimal Rules

As shown in Fig. 2, good rule sets are not likely to be obtained by genetic fuzzy rule selection when the number of candidate rules is too small. On the other hand, we need a long CPU time when the number of candidate rules is large. In our former study [9], we examined the use of Pareto-optimal fuzzy rules with respect to confidence and support as candidate rules. In this section, we examine the use of not only Pareto-optimal but also near Pareto-optimal fuzzy rules.

Using a dominance margin ε, we modify Pareto dominance as in [18]: A fuzzy rule R_i is said to be ε-dominated by another fuzzy rule R_j when both the following two inequalities hold,

$$c(R_i)+\varepsilon \le c(R_j)\,,\;\; s(R_i)+\varepsilon \le s(R_j)\,, \tag{6}$$

and at least one of the following two inequalities holds:

$$c(R_i)+\varepsilon < c(R_j)\,,\;\; s(R_i)+\varepsilon < s(R_j)\,. \tag{7}$$

When a fuzzy rule R_i is not dominated by any other fuzzy rules in the sense of the ε-dominance in (6) and (7), we call R_i an ε-non-dominated rule. It should be noted that the ε-dominance with $\varepsilon = 0$ is the same as Pareto dominance.

We examined the effect of using ε-non-dominated rules as candidate rules in multiobjective genetic fuzzy rule selection by computational experiments on five data sets from the UCI machine learning repository. The two-fold cross-validation procedure was iterated five times for each data set. First we extracted fuzzy rules using the minimum confidence 0.6 and the minimum support 0.01 (0.04 for the wine data set). Among the extracted fuzzy rules, only ε-non-dominated rules were used as candidate rules. Then we applied NSGA-II [6], [8] to search for non-dominated rule sets from the candidate rules. Finally the accuracy of each of the obtained non-dominated rule sets was calculated for training and test data.

In Table 1, we show the average number of candidate rules for each value of ε. In the case of $\varepsilon = 0$, the number of candidate rules (i.e., Pareto-optimal rules) was very small. On the other hand, it is large when $\varepsilon = \infty$. In this case, all the extracted fuzzy rules were used as candidate rules. By decreasing the value of ε, we can decrease the number of candidate rules as shown in Table 1.

Table 1. Average number of generated candidate fuzzy rules

Data set	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = \infty$
Breast W	74	28323	35093	46941	57931	78650
Glass	163	4496	6571	11324	14850	15140
Heart C	349	9407	11835	17154	30928	102560
Iris	21	1995	2161	2555	3264	4725
Wine	43	4728	7081	15948	37915	77805

Table 2. Average CPU time of a single run of NSGA-II (minutes)

Data set	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = \infty$
Breast W	4.9	167.9	185.8	270.0	338.3	453.2
Glass	2.0	11.2	27.9	41.5	59.5	73.3
Heart C	7.8	31.2	48.4	78.6	130.9	266.7
Iris	0.8	4.8	9.2	10.9	10.9	15.2
Wine	3.0	11.5	26.1	60.4	104.9	136.6

The average CPU time for a single run of NSGA-II is shown in Table 2. From Table 1 and Table 2, we can see that the computation load depends on the number of candidate rules. Using a small value of ε, we can decrease the CPU time.

From the viewpoint of CPU time, the use of Pareto-optimal rules as candidate rules (i.e., $\varepsilon = 0$ in Table 2) is a good strategy. The training data accuracy of obtained non-dominated rule sets in this case was not necessarily good as shown in Table 3. In Table 3, we calculated the average value of the highest classification rates on training data among obtained non-dominated rule sets by each run of NSGA-II. Good rule sets were not obtained in the case of $\varepsilon = 0$ for some data sets (e.g., Glass and Heart C). An interesting observation in Table 3 is that the best training data accuracy was not always obtained from a large value of ε. For example, the best training data accuracy

Table 3. Average value of the best classification rates on training data (%)

Data set	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = \infty$
Breast W	97.4	99.0	99.1	99.1	99.1	99.2
Glass	74.8	79.2	81.7	83.5	84.1	84.0
Heart C	71.8	80.8	79.4	79.7	77.9	78.5
Iris	97.2	97.5	98.8	97.7	97.9	97.9
Wine	100.0	100.0	100.0	100.0	100.0	100.0

Table 4. Average value of the best classification rates on test data (%)

Data set	$\varepsilon = 0$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = \infty$
Breast W	96.8	96.7	96.4	96.5	96.8	96.5
Glass	63.4	66.1	64.8	66.1	66.4	65.9
Heart C	54.8	55.4	56.4	56.3	55.2	56.5
Iris	96.6	97.0	97.0	96.5	96.8	96.9
Wine	93.8	92.7	93.3	90.4	94.9	94.9

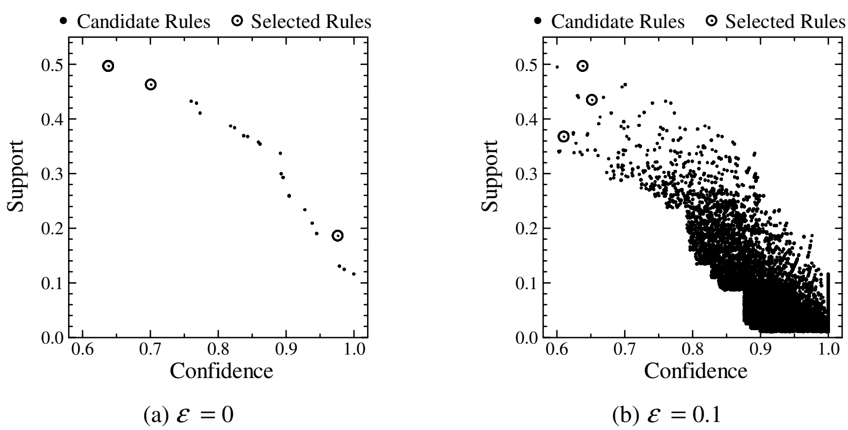


Fig. 3. Candidate rules and selected rules for Class 1 of Heart C

was obtained from $\varepsilon = 0.01$ for Heart C in Table 3. This is because NSGA-II could not find the optimal combination of candidate rules when the number of candidate rules was too large.

In Table 4, we show the average value of the highest classification rates on test data. While the training data accuracy was severely degraded for some data sets by specifying ε as $\varepsilon = 0$, the deterioration in the test data accuracy was not so severe.

For clearly demonstrating the effect of ε, we show candidate rules and selected rules for Class 1 of Heart C by a single-run of NSGA-II in Fig. 3.

5 Conclusions

We proposed an idea of using Pareto-optimal and near Pareto-optimal rules as candidate rules in genetic fuzzy rule selection. Through computational experiments, we demonstrated that the proposed idea decreased the number of candidate rules which were generated based on the minimum confidence and support. As a result, the CPU time for rule selection was decreased. We also demonstrated that rule sets with high training data accuracy were not obtained for some data sets when we used only Pareto-optimal rules. The proposed idea improved the training data accuracy of obtained rule sets by using not only Pareto-optimal but also near Pareto-optimal rules. A future research issue is to examine other definitions (e.g., multiplicative form) of ε-dominance. Different handling of multiobjective problems such as lexicographic ordering [17] is also a future research issue.

Acknowledgement

This work was supported by Grant-in-Aid for Scientific Research on Priority Areas (18049065) and Grant-in-Aid for Scientific Research (B) (17300075).

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I.: Fast Discovery of Association Rules. In: Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.): *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park (1996) 307-328
2. Casillas, J., Cordon, O., Herrera, F., Magdalena, L. (eds.): *Interpretability Issues in Fuzzy Modeling*. Springer, Berlin (2003)
3. Casillas, J., Cordon, O., Herrera, F., Magdalena, L. (eds.): *Accuracy Improvements in Linguistic Fuzzy Modeling*. Springer, Berlin (2003)
4. Coenen, F., Leng, P.: Obtaining Best Parameter Values for Accurate Classification. *Proc. of 5th IEEE International Conference on Data Mining* (2005) 549-552
5. de la Iglesia, B., Philpott, M. S., Bagnall, A. J., Rayward-Smith, V. J.: Data Mining Rules using Multi-Objective Evolutionary Algorithms. *Proc. of 2003 Congress on Evolutionary Computation* (2003) 1552-1559
6. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester (2001)

7. Deb, K., Mohan, M., Mishra, S.: Evaluating the ε-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation 14*, 4 (2005) 501-525
8. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation 6*, 2 (2002) 182-197
9. Ishibuchi, H., Kuwajima, I., Nojima, Y.: Relation between Pareto-Optimal Fuzzy Rules and Pareto-Optimal Fuzzy Rule Sets. *Proc. of 1st IEEE Symposium on Computational Intelligence in Multicriteria Decision Making* (2007) (in press)
10. Ishibuchi, H., Murata, T., Turksen, I. B.: Single-Objective and Two-Objective Genetic Algorithms for Selecting Linguistic Rules for Pattern Classification Problems. *Fuzzy Sets and Systems 89*, 2 (1997) 135-150
11. Ishibuchi, H., Nakashima, T., Murata, T.: Three-Objective Genetics-based Machine Learning for Linguistic Rule Extraction. *Information Sciences 136*, 1-4 (2001) 109-133
12. Ishibuchi, H., Nakashima, T., Nii, M.: *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*. Springer, Berlin (2004)
13. Ishibuchi, H., Nozaki, K., Yamamoto, N., Tanaka, H.: Selecting Fuzzy If-Then Rules for Classification Problems Using Genetic Algorithms. *IEEE Trans. on Fuzzy Systems 3*, 3 (1995) 260-270
14. Ishibuchi, H., Yamamoto, T.: 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
15. Kaya, M.: Multi-Objective Genetic Algorithm based Approaches for Mining Optimized Fuzzy Association Rules. *Soft Computing 10* (2006) 578-586
16. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. *Evolutionary Computation 10*, 3 (2002) 263-282
17. Luke, S., Panait, L.: Lexicographic Parsimony Pressure, *Proc. of 2002 Genetic and Evolutionary Computation Conference* (2002) 829-836
18. Reynolds, A., de la Iglesia, B.: Rule Induction using Multi-Objective Metaheuristics: Encouraging Rule Diversity. *Proc. of 2006 International Joint Conference on Neural Networks* (2006) 6375-6382
19. Wang, H., Kwong, S., Jin, Y., Wei, W., Man, K. F.: Multi-Objective Hierarchical Genetic Algorithm for Interpretable Fuzzy Rule-based Knowledge Extraction. *Fuzzy Sets and Systems 149*, 1 (2005) 149-186
20. Wang, H., Kwong, S., Jin, Y., Wei, W., Man, K. F.: Agent-based Evolutionary Approach for Interpretable Rule-based Knowledge Extraction. *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews 35*, 2 (2005) 143-155