A Hybrid Fuzzy Genetics-based Machine Learning Algorithm: Hybridization of Michigan Approach and Pittsburgh Approach

Hisao Ishibuchi, Tomoharu Nakashima, and Tetsuya Kuroda
Department of Industrial Engineering, Osaka Prefecture University
{hisaoi, nakashi, tkuroda}@ie.osakafu-u.ac.jp

ABSTRACT

In this paper, we propose a hybrid genetics-based machine learning algorithm for designing a linguistic classification system that consists of a small number of fuzzy if-then rules with clear linguistic interpretation. Our task is to generate a small number of fuzzy if-then rules from numerical data for a high-dimensional pattern classification problem. We assume that a set of linguistic values is given for each attribute of the pattern classification problem by human experts. Thus our task is described as finding a small number of combinations of linguistic values, each of which is used as the antecedent part of a fuzzy if-then rule. While this task seems to be simple at a glance, it is very difficult especially in the case of high-dimensional problems because the number of possible combinations of antecedent linguistic values exponentially increases with the dimensionality of problems. That is, the search space for high-dimensional problems is terribly huge. In our approach, an individual in genetic algorithms is a set of fuzzy if-then rules. Each rule is coded as a string by its antecedent linguistic values. Thus an individual (i.e., a rule set) is denoted by a concatenated string. The fitness of a rule set, which is defined by its classification performance, is used in a selection operation. New rule sets are generated by a crossover operation from selected rule sets. A mutation operation modifies a part of each rule set generated by the selection and crossover. As a mutation operation, we use a rule generation mechanism of Michigan approach. In Michigan approach, new fuzzy if-then rules are generated from existing rules with high classification performance. The performance of non-hybrid algorithms as well as our hybrid algorithm is examined by computer simulations.

1. INTRODUCTION

Fuzzy systems based on fuzzy if-then rules have been successfully used in many application areas [1,2]. Fuzzy if-then rules were traditionally obtained from human experts. Recently various methods have been proposed for automatically generating and adjusting fuzzy if-then rules without human experts (for example, [3-6]). Genetic algorithms [7,8] have been used as rule generation and optimization tools in the design of fuzzy rule-based systems [9-14]. Those GA-based studies on the design of fuzzy rule-based systems are often referred to as fuzzy genetics-based machine learning methods (fuzzy GBML methods), each of which can be classified into Pittsburgh or Michigan approach as non-fuzzy GBML methods. Many fuzzy GBML methods [9-11] are categorized as Pittsburgh approach [15] where a set of fuzzy if-then rules is coded as a string (i.e., an individual is a fuzzy rule-based system). Some studies [12-14] are categorized as Michigan approach (i.e., classifier systems [7,8,16]) where a single fuzzy if-then rule is coded as a string (i.e., an individual is a single fuzzy if-then rule).

In Pittsburgh approach where a number of fuzzy if-then rules is coded as a string and handled as an individual, the performance of each rule set (i.e., each individual) is used as its fitness value. Thus the genetic search for finding rule sets with high fitness values is equivalent to the search for fuzzy rule-based systems with high performance. That is, the optimization of fuzzy rule-based systems is directly handled by genetic algorithms that try to maximize the fitness function. Some good rule sets in a current population are inherited to the next population with no modification as elite individuals. The performance of each rule is not explicitly evaluated in Pittsburgh approach. Thus even if good rules exist in the current population, they are not always used for generating new rule sets. Especially when good rules are included in poor rule sets, they easily disappear during the generation update. Since a population consists of a number of rule sets, long computation time and large memory storage are

0-7803-5731-0/99/$10.00 ©1999 IEEE

required in Pittsburgh approach.

On the other hand, in Michigan approach where a single fuzzy if-then rule is coded as a string and handled as an individual, the performance of each rule is used as its fitness value. That is, the performance of rule sets is not utilized in the genetic search for finding fuzzy rule-based systems with high performance. Thus the optimization of fuzzy rule-based systems is indirectly performed by searching for good fuzzy if-then rules. Some good fuzzy if-then rules in the current population (i.e., in the current rule set) are inherited to the next population with no modification as elite individuals. The performance of the current rule set is not explicitly evaluated in the genetic search of Michigan approach. Thus a good rule set can be destroyed by the generation update (i.e., the performance of the current population can be decreased). Since a population includes only a single rule set, computation time and memory storage in Michigan approach are much smaller than those in Pittsburgh approach where a population consists of a number of rule sets.

In Michigan approach, good fuzzy if-then rules in the current population (i.e., in the current rule set) are inherited with no modification to the next population. The generation update in Michigan approach can be viewed as a partial change of the current population where bad rules are replaced with newly generated rules. Thus once good fuzzy if-then rules are found, they are not likely to disappear. On the other hand, genetic operations in Pittsburgh approach are not directly based on the performance of fuzzy if-then rules. Thus even good fuzzy if-then rules (especially included in poor rule sets) can easily disappear by generation update. Of course, good rule sets are inherited to the next population as elite individuals in Pittsburgh approach. In Table 1, we summarize characteristic features of these two approaches.

Table 1. Characteristic features of each approach.

Approach	Pittsburgh	Michigan
Individual	A rule set	A rule
Fitness definition	For a rule set	For a rule
Optimization	Direct	Indirect
Elite	Rule sets	Rules
Inheritance of good rules	×	○
Inheritance of good rule sets	○	×
Computation time	Long	Short
Memory storage	Large	Small

In this paper, we first compare these two approaches of fuzzy GBML. Then we combine them into a single hybrid algorithm for simultaneously utilizing advantages of each approach.

2. TWO APPROACHES OF FUZZY GBML

2.1 Fuzzy Rule-Based Classification Systems

We use fuzzy if-then rules of the following form for an n-dimensional pattern classification problem:

$$\text{Rule } R_j : \text{If } x_1 \text{ is } A_{j1} \text{ and ... and } x_n \text{ is } A_{jn} \text{ then Class } C_j \text{ with } CF_j, \quad (1)$$

where R_j is the label of the j-th fuzzy if-then rule, j indexes the number of rules, $\mathbf{x} = (x_1, x_2, ..., x_n)$ is an n-dimensional pattern vector, A_{ji} is an antecedent fuzzy set with a linguistic label (i.e., a linguistic value such as *small* and *large*) on the i-th axis, C_j is a consequent class, and CF_j is a certainty grade. In this paper, we assume that the pattern space is the n-dimensional unit cube $[0,1]^n$. In computer simulations, all attribute values are normalized into real numbers in the unit interval [0,1]. As the antecedent fuzzy sets A_{ji}'s, we use five linguistic values in Fig. 1 and *"don't care"*. Thus the total number of combinations of the antecedent fuzzy sets is 6^n, which is terribly large in the case of high-dimensional problems.

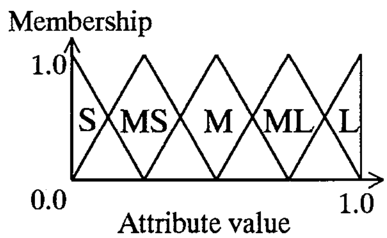

Fig. 1 Antecedent fuzzy sets.

As shown in Fig. 1, the meaning of each linguistic value is specified by a triangular membership function on the unit interval [0,1]. We handle *"don't care"* as a special linguistic value with the following membership function:

$$\mu_{don't\ care}(x) = \begin{cases} 1, & 0 \le x \le 1, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

In this paper, we assume that m training patterns $\mathbf{x}_p = (x_{p1}, ..., x_{pn})$, $p = 1,2,...,m$ are given. Our task is to generate a small number of fuzzy if-then rules from the training patterns. When antecedent fuzzy sets of a fuzzy if-then rule are specified, its consequent class and certainty grade are determined by a heuristic procedure [4]. Thus our task is viewed as finding a small number of combinations of antecedent fuzzy sets. Whereas this task does not involve the adjustment of membership functions or certainty grades, it is a very difficult task especially in the case of high-dimensional problems (note that the number of combinations of antecedent fuzzy sets is 6^n).

When a set of fuzzy if-then rules is given, its performance is evaluated by classifying the given training patterns using the rule set. We use a fuzzy reasoning method based on a single winner rule [4] where an input pattern is classified by the winner rule with the maximum product of the compatibility grade and the certainty grade.

2.2 Pittsburgh Approach

Let us denote the fuzzy if-then rule R_j in (1) by its n antecedent fuzzy sets as $R_j = A_{j1} \cdots A_{jn}$. That is, R_j is coded as a string of the length n. Let S be a set of N fuzzy if-then rules (i.e., $S = \{R_1, ..., R_N\}$). We denote S by a concatenated string of the length $n \times N$ where each sub-string of the length n corresponds to a single fuzzy if-then rule. That is, the rule set S is denoted as

$$S = A_{11} \cdots A_{1n}\, A_{21} \cdots A_{2n}\, \cdots\, A_{N1} \cdots A_{Nn}. \qquad (3)$$

The fitness of the rule set S is measured as

$$fitness(S) = NCP(S), \qquad (4)$$

where $NCP(S)$ is the number of correctly classified training patterns by S.

Let Ψ be a current population including N_{set} rule sets. We define the selection probability of each rule set S_i as

$$P(S_i) = \frac{fitness(S_i) - f_{\min}(\Psi)}{\sum_{S_k \in \Psi} \{fitness(S_k) - f_{\min}(\Psi)\}}, \qquad (5)$$

where $f_{\min}(\Psi)$ is the fitness value of the worst rule set in Ψ (i.e., the smallest fitness value in Ψ).

We use the uniform crossover where each sub-string is handled as a block. That is, the crossover replaces some rules in one parent with rules in the other parent. A mutation operation randomly replaces an antecedent fuzzy set with another one. By the selection, crossover and mutation operations, we generate $(N_{\text{set}} - 1)$ rule sets. The best rule set in the current population is added to the $(N_{\text{set}} - 1)$ rule sets as an elite individual to form a new population with N_{set} rule sets.

2.3 Michigan Approach

As in Pittsburgh approach, a fuzzy if-then rule is denoted by its antecedent fuzzy set: $R_j = A_{j1} \cdots A_{jn}$. In Michigan approach, a single fuzzy if-then rule is handled as an individual, which is coded as a string of the length n. A population S consists of N fuzzy if-then rules: $S = \{R_1, ..., R_N\}$. The fitness value of a fuzzy if-then rule R_i in the current population S (i.e., in the rule set S) is defined as follows after all the given training patterns are classified by the rule set S.

$$fitness(R_j) = w_1 \times NCP(R_j) - w_2 \times NMP(R_j), \qquad (6)$$

where $NCP(R_j)$ is the number of correctly classified training patterns by R_j, $NMP(R_j)$ is the number of misclassified training patterns by R_j, and w_1 and w_2 are positive constants.

The selection probability of each rule is defined by a roulette wheel selection with a linear scaling as in (5). The uniform crossover is used for generating two fuzzy if-then rules from a pair of parent rules. A mutation operation randomly replaces an antecedent fuzzy set with another one. By the selection, crossover and mutation operations, we generate N_{replace} fuzzy if-then rules. The worst N_{replace} fuzzy if-then rules in the current population are replaced with the newly generated rules to form the next population (i.e., the next rule set) with N rules. Since the number of replaced rules is the same as that of newly added rules, the size of rule sets is kept constant throughout the iterative execution of Michigan approach as well as Pittsburgh approach.

2.4 Computer Simulations

The two approaches of fuzzy GBML were applied to wine data with 13 attributes (available from UC Irvine Database). This data set was often used in the literature. Since the wine data set involves 13 attributes and we use six antecedent fuzzy sets (i.e., five linguistic values and "*don't care*") for each attribute, the total number of possible fuzzy if-then rules is $6^{13} \cong 1.3 \times 10^{10}$. The wine data set is not a difficult classification problem because there are no class overlaps. But the search for a small number of fuzzy if-then rules is very difficult due to a huge number of combinations of antecedent fuzzy sets (i.e., $6^{13} \cong 1.3 \times 10^{10}$ combinations).

We applied the two approaches of fuzzy GBML (i.e., Pittsburgh approach and Michigan approach) to the wine data set with the following parameter specifications:

Number of rules in each rule set: $N = 20$,
Number of rule sets in each generation:
$N_{\text{rule}} = 50$ (in Pittsburgh approach),
$N_{\text{rule}} = 1$ (in Michigan approach),
Weights w_1 and w_2 in Michigan approach:
$w_1 = 1,\ w_2 = 5$,
Crossover probability: 0.9,
Mutation probability: 0.1,
Stopping condition: 500 generations,
Number of replaced rules in Michigan approach:
N_{replace} : 20% of the current rule set.

Each approach was applied to the wine data set ten times using different initial populations. The following average classification rate on the training patterns was obtained at the 500th generation:

Pittsburgh approach: 63.1%,
Michigan approach: 95.1% at the 500th generation,
96.3% (the best result among 500 generations).

From these results, we can see that Michigan approach can efficiently search for good fuzzy if-then rules among a huge number of possible rules. Similar results were reported in Ishibuchi et al.[17] with different conditions about genetic operations and parameter specifications. Higher classification rates (i.e., almost 100% classification rates) can be obtained by Michigan approach by increasing the number of fuzzy if-then rules and/or the number of generations (see Ishibuchi et al.[13, 14, 17]).

For identifying the essential elements of Michigan approach, we applied it to the wine data with various conditions. First we examined the effect of the number of replaced rules (i.e., N_{replace}) on the performance of Michigan approach. We used three specifications of N_{replace} : 20 (100% of the current population), 10 (50%) and 4 (20%). In computer simulations, we used two specifications of the weight values: $(w_1, w_2) = (1,5), (5,1)$. Average classification rates at the 500th generation are summarized in Table 2. From this table, we can see that the performance of Michigan approach was significantly deteriorated by replacing all rules in the current population (i.e., by specifying N_{replace} as 20).

Next we examined the effect of the selection operation on the performance of Michigan approach. We performed computer simulations with a random selection operation where parent rules were randomly selected from the current population. Simulation results are summarized in Table 3. From the comparison between Table 2 and Table 3, we can see that the good result (i.e., a 97.1% average classification rate) was obtained when the parameter values were appropriately specified. We can also see from Table 3 that the selection of parent rules has a large effect on the performance of Michigan approach when the parameter values were inappropriately specified.

We also examined the effect of the selection of replaced rules on the performance of Michigan approach. We performed computer simulations by randomly selecting replaced rules from the current population. Simulation results are summarized in Table 4. From the comparison between Table 2 and Table 4, we can see that good results were not obtained in this case (i.e., the best average classification rate was 70.0%).

Table 2. Classification rates at the 500th generation.

Weights (w_1, w_2)	Number of replaced rules (N_{replace})		
	20	10	4
(5, 1)	70.4%	96.4%	95.1%
(1, 5)	27.0%	97.7%	95.1%

Table 3. Simulation results with a random selection of parent rules.

Weights (w_1, w_2)	Number of replaced rules (N_{replace})		
	20	10	4
(5, 1)	0.2%	95.7%	95.9%
(1, 5)	0.1%	97.1%	89.2%

Table 4. Simulation results with a random selection of replaced rules.

Weights (w_1, w_2)	Number of replaced rules (N_{replace})		
	20	10	4
(5, 1)	69.7%	66.3%	70.0%
(1, 5)	26.0%	21.1%	17.0%

While Michigan approach has high ability to efficiently find good fuzzy if-then rules (see Table 2), this does not always mean its high ability to find good rule sets because Michigan approach does not directly search for the best rule set. This indirect search nature causes a difficulty in finding good rule sets when the number of fuzzy if-then rules is very small. We applied Michigan approach with five fuzzy if-then rules (i.e., $N = 5$) to the wine data in the same manner as in Table 2. The weight values were specified as $w_1 = 1$ and $w_2 = 5$, and N_{replace} as $N_{\text{replace}} = 1$. The average classification rate at the 500th generation over ten independent trials was 38.3%. In each trial, there were many ups and downs of the classification rate over 500 generations (see Fig. 2 where simulation results of a single trial are shown). This is because the evolution of fuzzy if-then rules in Michigan approach is not driven by the performance of rule sets. On the other hand, the classification rate of the elite individual (i.e., the best rule set at each generation) in Pittsburgh approach does not decrease because the fitness function in (4) is the same as the classification rate. Pittsburgh approach, however, does not have high search ability to efficiently find good fuzzy if-then rules. In the next section, we try to combine the two approaches for utilizing advantages of each approach into a single hybrid algorithm.

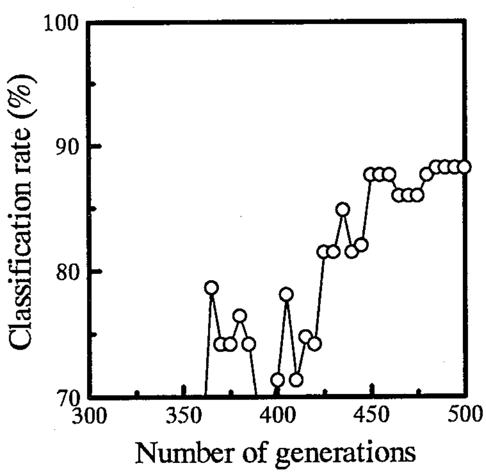

Fig. 2 Classification rate at each generation obtained by Michigan approach with five fuzzy if-then rules.

3. HYBRIDIZATION

From the simulation results in the previous section, it seems that the following points are essential in designing a high performance hybrid algorithm:

(1) To inherit good rules (see Table 2).
(2) To generate new rules from good rules (see Table 3).
(3) To remove bad rules (see Table 4).
(4) To directly maximize the performance of rule sets.

Our hybrid algorithm is based on Pittsburgh approach. Only its mutation operation is modified. The rule generation mechanism of Michigan approach is used as a mutation operation in our hybrid algorithm. That is, the worst rules in each rule set (i.e., in each individual) are replaced with new rules that are generated from good rules in that rule set. In our hybrid approach, the search for good fuzzy if-then rules is mainly driven by the rule generation mechanism of Michigan approach. Thus we use a small probability for the crossover operation in Pittsburgh approach.

We applied our hybrid algorithm to the wine data with the following parameter specifications:

Number of rules in each rule set: $N = 20$,
Number of rule sets in each generation: $N_{\text{rule}} = 50$,
Crossover probability: 0.3 in Pittsburgh approach,
0.9 in Michigan approach,
Mutation probability: 0.1,
Stopping condition: 500 generations,
Number of replaced rules in Michigan approach:
N_{replace} : 20% of the current rule set.

The average classification rate over ten independent trials was 97.9%. This is better than the average results by Pittsburgh approach and Michigan approach in the previous section. That is, our hybrid algorithm has high search ability to efficiently find good fuzzy if-then rules. We also applied our hybrid algorithm to the wine data set by specifying the number of fuzzy if-then rules in each rule set as $N = 5$. The average classification rate over ten trials was 85.2% at the 500th generation, which is much better than the results by Pittsburgh approach (45.6%) and Michigan approach (38.3%).

4. CONCLUSION

In this paper, we compared the two approaches of fuzzy GBML to the design of linguistic rule-based systems for high-dimensional pattern classification problems. We demonstrated high search ability of Michigan approach to find good fuzzy if-then rules through computer simulations on wine data with 13 attributes. We also showed that its high search ability was deteriorated when all rules were replaced with newly generated rules, parent rules were randomly selected, and existing rules were randomly removed from the current rule set. These results suggested that good rules in the current rule set should be inherited to the next rule set, new rules should be generated from existing good rules, and bad rules should be replaced with newly generated rules. A difficulty related to the indirect search for good rule sets in Michigan approach was also demonstrated.

We combined the two approaches for utilizing their advantages in a single hybrid algorithm. In our hybrid algorithm, the rule generation mechanism of Michigan approach is used as a mutation operation in Pittsburgh approach. Its performance was examined through computer simulations on the wine data. Our hybrid algorithm has high search ability to find good fuzzy if-then rules as in Michigan approach. The evolution of fuzzy if-then rules in our hybrid algorithm is driven by the performance of rule sets as in Pittsburgh approach. In this sense, our hybrid algorithm has the advantages of the two approaches. On the other hand, our hybrid algorithm has the same disadvantages as Pittsburgh approach with respect to computation time and memory storage. This is because a set of fuzzy if-then rules is coded as a string and handled as an individual in our hybrid algorithm as in Pittsburgh approach. For overcoming these disadvantages, the development of hybrid algorithms based on Michigan approach is left for future work.

REFERENCES

[1] M. Sugeno, "An introductory survey of fuzzy control," *Information Sciences*, vol. 36, no. 1/2, pp. 59-83, 1985.

[2] C. C. Lee, "Fuzzy logic in control systems: fuzzy logic controller Part I and Part II," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404-435, March/April, 1990.

[3] L. X. Wang, and J. M. Mendel, "Generating fuzzy rules by learning from examples," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, no. 6, pp. 1414-1427, November/ December, 1992.

[4] H. Ishibuchi, K. Nozaki, and H. Tanaka, "Distributed representation of fuzzy rules and its application to pattern classification," *Fuzzy Sets and Systems*, vol. 52, no. 1, pp. 21-32, November, 1992.

[5] S. Abe, and M.-S. Lan, "A method for fuzzy rules extraction directly from numerical data and its application to pattern classification," *IEEE Trans. on Fuzzy Systems*, vol. 3, no. 1, pp. 18-28, February, 1995.

[6] S. Mitra, and S. K. Pal, "Self-organizing neural network as a fuzzy classifier," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 24, no. 3, pp. 385-399, March, 1994.

[7] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[9] C. L. Karr, and E. J. Gentry, "Fuzzy control of pH using genetic algorithms," *IEEE Trans. on Fuzzy Systems*, vol. 1, no. 1, pp. 46-53, February, 1993.

[10] Herrera, M. Lozano, and J. L. Verdegay, "Tuning fuzzy logic controllers by genetic algorithms," *International J. of Approximate Reasoning*, vol. 12, no. 3/4, pp. 299-315, April/May, 1995.

[11] B. Carse, T.C. Fogarty, and A. Munro, "Evolving fuzzy rule based controllers using genetic algorithms," *Fuzzy Sets and Systems*, vol. 80, no. 3, pp. 273-293, June, 1996.

[12] M. Valenzuela-Rendon, "The fuzzy classifier system: A classifier system for continuously varying variables," *Proc. of 4th International Conference on Genetic Algorithms*, pp. 346-353, University of California, San Diego, CA, July, 1991.

[13] H. Ishibuchi, T. Nakashima, and T. Murata, "A fuzzy classifier system that generates fuzzy if-then rules for pattern classification problems," *Proc. of 2rd IEEE International Conf. on Evolutionary Computation*, pp. 759-764, Perth, Australia, November, 1995.

[14] H. Ishibuchi, T. Nakashima, and T. Murata, "Performance evaluation of fuzzy classifier systems for multi-dimensional pattern classification problems," *IEEE Trans. on Systems, Man, and Cybernetics* (in press).

[15] S. F. Smith, "A learning system based on genetic algorithms," Ph.D. Dissertation, University of Pittsburgh, Pittsburgh, PA, 1980.

[16] L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier systems and genetic algorithms," *Artificial Intelligence*, vol. 40, no. 1-3, pp. 235-282, September, 1989.

[17] H. Ishibuchi, T. Nakashima, and T. Murata, "Genetic-algorithm-based approaches to the design of fuzzy systems for multi-dimensional pattern classification problems," *Proc. of 3rd IEEE International Conference on Evolutionary Computation*, pp. 229-234, Nagoya, Japan, May, 1996.