An Adaptive Fuzzy Rule Extraction Using Hybrid Model of the Fuzzy Self-Organizing Map and the Genetic Algorithm with Numerical Chromosomes

Tatsuya Nomura [1] **and Tsutomu Miyoshi** [2]

[1] ATR Human Information Processing Research Laboratories
2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan
TEL: +81-774-95-1010, FAX: +81-774-95-1008, E-mail: nomura@hip.atr.co.jp

[2] Image Media Research Laboratories, SHARP CORPORATION
1-9-2, Nakase, Mihama-ku, Chiba-shi, Chiba 261, Japan
TEL: +81-43-299-8702, FAX: +81-43-299-8709, E-mail: mijosxi@iml.mkhar.sharp.co.jp

Key Word: neural network, Kohonen map, fuzzy reasoning, genetic algorithm, Lamarckian evolution

Abstract

In this paper, we propose a hybrid model of the fuzzified Kohonen's Self-Organizing Map and the GA with numerical chromosomes, and automatic fuzzy rule extraction method that uses our model. It is shown that our hybrid model is superior to both of the individual models in cases where there is a tendency for data to change dynamically and quickly.

1 Introduction

Though fuzzy inference rules have usually been constructed through trial and error by humans, many methods with machine learning such as neural networks and genetic algorithms have recently been proposed for automatic rule extraction from a given set of input-output data examples. For the purpose of extracting fuzzy If-Then rules from input-output data, we have proposed the Fuzzy Self-Organizing Map (FSOM) [2], which has both the architecture of Kohonen's Self-Organizing Map (SOM) [1] and the structure of fuzzy If-Then rules. In addition, we have also proposed a genetic algorithm using numerical chromosomes and an appropriate crossover method for the numerical chromosomes, called the "Unfair Average Crossover" [3]. We have shown the superiority of our methods to conventional neural networks such as Radial Basis Function and genetic algorithms using bit string chromosomes in dynamic environment.

In this paper, we propose a hybrid model of the above FSOM and GA for adaptive fuzzy rule extraction in dynamic environments and verify its effectiveness. Although hybrid models of SOM-like neural networks and GAs have been proposed [4][5], our model is specialized for fuzzy clustering to some degree.

2 Fuzzy Self-Organizing Map

The FSOM executes the fuzzy partitioning of the input space based on fuzzy competitive learning. In the FSOM, the following one-dimensional Gaussian membership function in equation (1) is assigned as the weight between the j-th input node and the i-th output node:

$$F_{ij}(x) = \exp((-1/2)(\mu_{ij} - x)^2/\sigma_{ij}^2). \quad (i = 1, \ldots, m, j = 1, \ldots, n) \tag{1}$$

For n-dimensional input vector $I = (I_1, \ldots, I_n) \in \boldsymbol{R}^n$, the output value of each output node is given by the following equation (2):

$$\begin{aligned} F_i(I) &= \prod_{j=1}^{n} F_{ij}(I_j) \qquad (2) \\ &= \exp((-1/2)\sum_{j=1}^{n}(\mu_{ij} - I_j)^2/\sigma_{ij}^2). \\ &\quad (i = 1, \ldots, m) \end{aligned}$$

Equation (2) is an n-dimensional Gaussian membership function with center $\mu_i = (\mu_{i1}, \ldots, \mu_{in})$ and width $\sigma_i = (\sigma_{i1}, \ldots, \sigma_{in})$. We can regard the value of $F_i(I)$ as the degree of inclusion into the fuzzy set in n-dimensional Euclidean space represented by the membership function F_i.

FSOM performs fuzzy partitioning of the input space based on competitive learning like SOM. Unlike the original SOM which learns only the centers of clusters of the input vectors, FSOM learns both the centers of clusters and deviations around the centers using the following fuzzy competitive learning:

$$\begin{aligned}\Delta\mu_{ij} &= \gamma f_i(I_j - \mu_{ij}), && (3)\\ \Delta\sigma_{ij} &= 2\gamma f_i \sigma_{ij}((\mu_{ij} - I_j)^2 - \sigma_{ij}^2), && (4)\\ f_i &= F_i(I)^{1/h} / \sum_{d=1}^{m} (F_d(I))^{1/h}. && (5)\\ & (i = 1, \ldots, n, j = 1, \ldots, m) &&\end{aligned}$$

Here, $\gamma(>0)$ and $h(>0)$ are learning parameters. As the learning proceeds, h decreases and the fuzzy neibourhoods among FSOM output nodes are reduced. This learning rule is based on the Fuzzy c-Means method and maximum-entropy approach. (It has experimentally been shown that this learning rule is superior to the one in [2] in dynamic environments [6].)

3 GA with Numerical Chromosomes and Unfair Average Crossover

In the framework of numerical coding, we directly code the chromosomes with an array of numerical parameters from the membership functions of fuzzy If-Then rules. For these chromosomes, we use the Unfair Average Crossover (UFAC). In our UFAC, two offsprings are made from two parents. We make four kinds of numerical values from two elements f and m at each location in the two parent arrays of the following way:

$$\left.\begin{array}{lcl} E_1(f,m) &=& (1 + (1/a))f - (1/a)m \\ E_2(f,m) &=& -(1/a)f + (1 + (1/a))m \\ M_1(f,m) &=& (1 - (1/a))f + (1/a)m \\ M_2(f,m) &=& (1/a)f + (1 - (1/a))m \end{array}\right\}. \qquad (6)$$
$$(a: \text{ real value}, a \geq 2)$$

$E_1(f,m)$ is the value far from the average in the direction of the father's value, $M_1(f,m)$ is the value near the average from the direction of the father's value, $E_2(f,m)$ is far from the average in the direction of the mother's value, and $M_2(f,m)$ is near the average from the direction of the mother's value.

Figure 1 shows the procedure of UFAC. We cut two parent arrays at one point and make two offspring arrays using E_1, E_2, M_1, and M_2. Offspring 1 heavily inherits the characteristics of the parents and offspring 2 inherits them on an average amount. If one of the parents has an appropriate characteristic, one of the offsprings inherits it to some degree. Furthermore, variety in the population is maintained by producing offsprings far from the average.

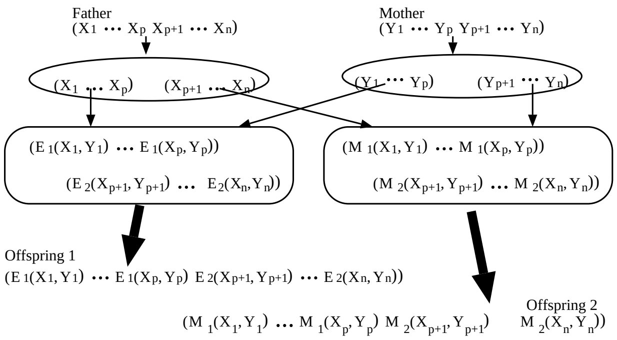
Father
$(X_1 \cdots X_p\ X_{p+1} \cdots X_n)$
Mother
$(Y_1 \cdots Y_p\ Y_{p+1} \cdots Y_n)$
$(X_1 \cdots X_p)$
$(X_{p+1} \cdots X_n)$
$(Y_1 \cdots Y_p)$
$(Y_{p+1} \cdots Y_n)$
$(E_1(X_1, Y_1) \cdots E_1(X_p, Y_p))$
$(E_2(X_{p+1}, Y_{p+1}) \cdots E_2(X_n, Y_n))$
$(M_1(X_1, Y_1) \cdots M_1(X_p, Y_p))$
$(M_2(X_{p+1}, Y_{p+1}) \cdots M_2(X_n, Y_n))$
Offspring 1
$(E_1(X_1, Y_1) \cdots E_1(X_p, Y_p)\ E_2(X_{p+1}, Y_{p+1}) \cdots E_2(X_n, Y_n))$
Offspring 2
$(M_1(X_1, Y_1) \cdots M_1(X_p, Y_p)\ M_2(X_{p+1}, Y_{p+1})$
$M_2(X_n, Y_n))$

Figure 1: Unfair Average Crossover

4 Hybrid Model

In this paper, we propose a method for extracting fuzzy If-Then rules from input-output data using a hybrid model of the above FSOM and GA.

We use the following form of input-output data that has n numerical input attributes and a single output value:

$$\left.\begin{array}{lcl} S &=& \{S^{(1)}, S^{(2)}, \ldots, S^{(N)}\} \\ S^{(l)} &=& (a_{l1}, a_{l2}, \ldots, a_{ln} \ ;\ c_l), \\ a_l &=& (a_{l1}, a_{l2}, \ldots, a_{ln}) \in \boldsymbol{R}^n \\ c_l &\in& C = \{C_1, C_2, \ldots, C_L\} \\ && \quad (l = 1, \ldots, N) \\ S_k &=& \{S^{(l)} \in S :\ c_l = C_k\} \\ && \quad (k = 1, \ldots, L) \end{array}\right\} \qquad (7)$$

Here, C is a set of discrete or symbolic value. This input-output data $S^{(l)}$ is also regarded as an $n-$ dimensional numerical vector a_l labeled with c_l.

Moreover, we use the following form of fuzzy If-Then rules to be extracted from the above input-output data:

$$\begin{array}{l} \text{If } I_1 \text{ is } F_{ki1} \text{ and } \ldots \text{ and } I_n \text{ is } F_{kin}, \\ \qquad \text{Then } c \text{ is } C_k \text{ with weight } \omega_{ki} \qquad (8) \\ \qquad\qquad \displaystyle\sum_{i=1}^{m} \omega_{ki} = 1 \ , \end{array}$$

$$F_{kij}(x) = \exp((-1/2)(\mu_{kij} - x)^2/\sigma_{kij}^2).$$
$$(k = 1, \ldots, L,\ i = 1, \ldots, m,\ j = 1, \ldots, n)$$

For input vector $I = (I_1, \ldots, I_n)$, these rules with Gaussian membership functions do product–sum simplified inference with a degree of confidence in the following way:

$$\begin{aligned} P_k(I) &= p_k(I) / \sum_{r=1}^{L} p_r(I) \ ; \qquad (9) \\ p_k(I) &= \sum_{i=1}^{m} \left\{ \prod_{j=1}^{n} F_{kij}(I)\omega_{ki} \right\}. \\ & (k = 1, \ldots, L) \end{aligned}$$

The value $P_k(I)$ represents the degree where a data with the numerical input attribute I has output C_k.

We correspond one fuzzy rule to one numerical chromosome using an array of numerical parameters from the Gaussian membership functions:

$$\begin{aligned} Ch_i^{(k)} &= (\mu_{ki1}\ \sigma_{ki1}\ \ldots\ \mu_{kin}\ \sigma_{kin}). \qquad (10) \\ & (k = 1, \ldots, L,\ i = 1, \ldots, m) \end{aligned}$$

We gather all the chromosomes corresponding to the rules with the same Then-part "c is C_k" in one population POP_k and construct a total of L populations. We apply the GA described in Section 3 to each population individually. We define the evaluation value e_{ki} and the fitness value f_{ki} of the i-th chromosome in POP_k, $Ch_i^{(k)}$, in the following way:

$$e_{ki} = (1/2) \sum_{l=1}^{N} \left(\prod_{j=1}^{n} F_{kij}(a_{lj}) - T_k(c_l) \right)^2, (11)$$

$$T_k(c_l) = \delta_{kl} \quad (k = 1, \ldots, L,\ i = 1, \ldots, m),$$

$$f_{ki} = \max_{r=1,\ldots,m}(e_{kr}) - e_{ki} + \alpha \min_{r=1,\ldots,m}(e_{kr}). (12)$$

$$(k = 1, \ldots, L,\ i = 1, \ldots, m)$$

Here, $\alpha (> 0)$ is the selection pressure, e_{ki} in equation (12) means the squared error of the if-part of the corresponding rule for all the input-output data.

Moreover, we correspond one rule to one output node of the FSOM by regarding the fuzzy membership funcions in the corresponding rule $\{F_{ij1}, \ldots, F_{kin}\}$ as the weights toward the output node. Then, we construct a total of L FSOMs, $FS_1, \ldots, FS_L$, by gathering all the output nodes corresponding to the rules with the same Then-part. Thus, the i-th chromosome in POP_k corresponds to the i-th output node in FS_k $(i = 1, \ldots, m,\ k = 1, \ldots, L)$. We regard the above chromosomes as genotype and the results of FSOM learning as phenotype. In other words, we make each individual learn the tendency of input-output data using the fuzzy competitive learning by regarding the chromosome as an initial point of learning, then evaluate the fitness value of the individuals based on the learning result. Furthermore, we take a Lamarckian stance [7]; that is, learning results are directly incorporated into chromosomes and inherited by offsprings.

We execute rule extraction with the following procedures:

0: Initialize the arrays of real values in equation (10) randomly.

1: In each population POP_k $(k = 1, \ldots, L)$, do roulette selection, unfair average crossover, and mutation by exchanging a randomly selected element in the array for a random value.

2: In each FS_k $(k = 1, \ldots, L)$, transfer the parameters of the rules in the chromosomes in POP_k into the weights of FS_k and modify the weights based on the fuzzy competitive learning for D times (D is a given integer).

Here, the learning data given in FS_k are the input attribute vectors a_l of $S^{(l)} \in S_k$ and that of $S^{(l)} \notin S_k$ is ignored (*i.e.*, only the numerical vectors labeled with C_k).

3: After learning in **2**, transfer the parameters of the rules in the FS_k into the chromosomes in POP_k.

4: repeat **1**–**3** for T times (T is a given integer)

5: determine the degree of confidence in the rules by normalizing the final fitness values among the chromosomes in the same population.

5 Simulations

We performed comparative simulations by using data with three input attributes and three kinds of output values (in equation (7), $n = L = 3$). We assumed that the distributions of input attribute vectors gradually changed as shown in Table 1 and applied the FSOM [2], the GA [3], and the hybrid method to those data with these distributions of the Gaussian type. For each distribution shown in Table 1, we prepared a total of 90 learning data and 90 testing data. For the learning data of each step in Table 1, we executed the learning for 30 times in the FSOM, 30 generations in the GA, and the learning for 9 times and 3 generations in the hybrid method. Then, for the fuzzy rules extracted from the learning data, we evaluated the correctness rate of the rules for the testing data. Note that the initial state of adaptation for each step is the final state of adaptation of the previous step in all the methods. The learning parameter of the FSOM and the hybrid methods is 0.01, mutation and

Table 1: Transition of the distributions of input attribute vectors in the simulation

step num	Center of input attribute vectors with C_1	Center of input attribute vectors with C_2	Center of input attribute vectors with C_3	deviation of each distribution
1	(2, 0, 0)	(0, 0, 2)	(0, 2, 2)	1
2	(1.9, 0.1, 0.1)	(0.1, 0, 1.9)	(0, 1.9, 2)	0.95
3	(1.8, 0.2, 0.2)	(0.2, 0, 1.8)	(0, 1.8, 2)	0.9
4	(1.7, 0.3, 0.3)	(0.3, 0, 1.7)	(0, 1.7, 2)	0.85
5	(1.6, 0.4, 0.4)	(0.4, 0, 1.6)	(0, 1.6, 2)	0.8
6	(1.5, 0.5, 0.5)	(0.5, 0, 1.5)	(0, 1.5, 2)	0.75
7	(1.4, 0.6, 0.6)	(0.6, 0, 1.4)	(0, 1.4, 2)	0.7
8	(1.3, 0.7, 0.7)	(0.7, 0, 1.3)	(0, 1.3, 2)	0.65
9	(1.2, 0.8, 0.8)	(0.8, 0, 1.2)	(0, 1.2, 2)	0.6
10	(1.1, 0.9, 0.9)	(0.9, 0, 1.1)	(0, 1.1, 2)	0.55
11	(1, 1, 1)	(1, 0, 1)	(0, 1, 2)	0.5
12	(0.9, 1.1, 1.1)	(1.1, 0, 0.9)	(0, 0.9, 2)	0.52
13	(0.8, 1.2, 1.2)	(1.2, 0, 0.8)	(0, 0.8, 2)	0.54
14	(0.7, 1.3, 1.3)	(1.3, 0, 0.7)	(0, 0.7, 2)	0.56
15	(0.6, 1.4, 1.4)	(1.4, 0, 0.6)	(0, 0.6, 2)	0.58
16	(0.5, 1.5, 1.5)	(1.5, 0, 0.5)	(0, 0.5, 2)	0.61
17	(0.4, 1.6, 1.6)	(1.6, 0, 0.4)	(0, 0.4, 2)	0.68
18	(0.3, 1.7, 1.7)	(1.7, 0, 0.3)	(0, 0.3, 2)	0.76
19	(0.2, 1.8, 1.8)	(1.8, 0, 0.2)	(0, 0.2, 2)	0.84
20	(0.1, 1.9, 1.9)	(1.9, 0, 0.1)	(0, 0.1, 2)	0.92
21	(0, 2, 2)	(2, 0, 0)	(0, 0, 2)	1

crossover rates of the GA and the hybrid methods are respectively 0.01 and 0.8.

Figure 2 shows the results of the experiments. Each value represents the average of the above correctness rate for ten trials with different initial states. Because of the low learning parameter for the fast change of the tendency of the data, the correctness rate of the FSOM was lower at the former steps and higher at the latter steps; and that of the GA changed less drastically and shows its stabilty in the state of non-equilibrium. The correctness rate of the hybrid method showed the aspect including those of both the FSOM and the GA. Although it was lower than that of the GA at the initial step in the same way as that of the FSOM, it became almost the same as that of the GA at step 3. This is caused by the effect of the GA. Then, caused by the effect of learning of the FSOM, it became higher than that of the GA at the latter steps in the same way as that of the FSOM. This result shows that the hybrid model gives play to both the FSOM's and the GA's abilities in case where the tendency of data dynamically and quickly changes.

6 Conclusion

We proposed an automatic fuzzy rule extraction method using the hybrid model of the FSOM and the GA with numerical chromosomes. It was shown that our hybrid model is superior to both of the individual models in cases where there is a tendency for data to change dynamically and quickly.

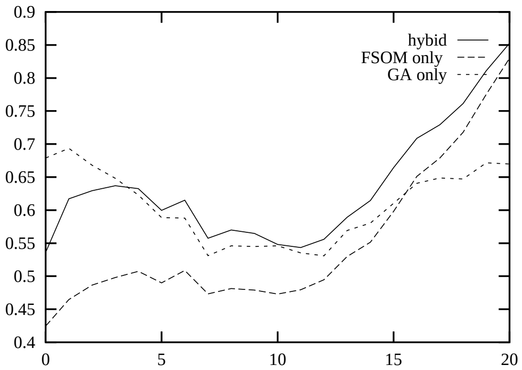

Figure 2: Average Correctness Rate of Each Method for the Simulation Data

References

[1] T. Kohonen, "Self-Organizing and Associative Memory." Springer-Verlag, 1989.

[2] T. Nomura and M. Miyoshi, "An Adaptive Rule Extraction with the Fuzzy Self-Organizing Map and a Comparison with Other Methods," in *Proc. ISUMA-NAFIPS'95*, pp. 311–316, Sept 1995.

[3] T. Nomura and M. Miyoshi, "Numerical Coding and Unfair Average Crossover in GA for Fuzzy Clustering and Their Applications for Automatic Fuzzy Rule Extraction," in *Proc. IEEE/Nagoya University WWW'95*, pp. 13–21, Nov 1995.

[4] S. A. Harp and T. Samad, "Genetic Optimization of Self-Organizing Feature Maps," in *Proc. IJCNN'91*, vol. I, pp. 341–346, 1991.

[5] G. J. Toth and A. Lorincz, "Genetic Algorithm with Migration on Topology Conserving Maps," in *Proc. WCNN'93*, vol. 3, pp. 168–171, 1993.

[6] T. Nomura and T. Miyoshi, "Construction of a Neural Network and a Genetic Algorithm for Fuzzy If-Then Rule Extraction," Tech. Rep. IIS-95-19, IEE Japan, 1995.

[7] D. H. Ackley and M. L. Littman, "A Case for Lamarckian Evolution," in *Artificial Life III* (C. G. Langton, ed.), pp. 3–10, Addison-Wesley, 1994.