2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

Fuzzy Reasoning Models with Different Error Measure Criteria

Shinya Fukumoto† , Hiromi Miyajima‡ and Yoji Nagasawa†

†Department of Information and Computer Science, Kagoshima University
‡Department of Electrical and Electronics Engineering, Kagoshima University
1-21-40 Koorimoto, Kagoshima-shi, Kagoshima 890-0065 Japan
Email: †{fukumoto, nagasawa }@ics.kagoshima-u.ac.jp
‡miya@eee.kagoshima-u.ac.jp

Abstract—This paper proposes a learning algorithm of fuzzy reasoning systems with different error measure criteria. In the method, objective functions with reasoning error and regularization term are used in learning of the weights of the consequent part for fuzzy rules. Two functions as regularization terms are proposed. One is multi-peak function shaped like a hump and the other is a cone-shaped function. The former is the method that the absolute values of the weights approach 0.5. The latter is the method that minimizes the absolute of the sum of the weights. In order to demonstrate the validity of the proposed method, some numerical simulations are performed.

1. Introduction

Many studies have been done with self-tuning fuzzy systems. Their aims are to construct automatically fuzzy inference rules by input and output data based on the steepest descend method. These approaches based on the steepest descend method contain some problems that are increase of inference error or learning times. Because systems of these approaches are constructed by using local search. In order to resolve above problems, several techniques had been proposed[1]. The methods with changing the structure of fuzzy models like to increase or eliminate fuzzy rules in the learning process have been proposed[2]. On the other hand fuzzy systems are combined with other methods like ANN, GA, SOM and LVQ. Further another approaches improved the performance of systems using regularization have been proposed[3]. On the normal regularized technique, regularization parameter and function are added to the error measure function. In this paper, we take account of the weights of consequent part and adopt different types of regularization terms. One is a term derived by a multi-peak function and the other is derived by a cone-shaped function. These terms work so that the weights of consequent part approaches constant value. Hence, it can be expected that the fuzzy systems are constructed as the function of the system becomes smooth. Since this is an action to keep the contribution of each rule, it seems that the performance of parallel computation in the fuzzy models is improved. We would like to examine the validity of the proposed method through numerical simulations.

2. Fuzzy Reasoning Model Using Delta Rule

This section describes the fuzzy reasoning model using delta rule. This reasoning model is the basis for proposed method.

When the input data are expressed by $x_1, \cdots, x_m$ and its output is expressed by y^*, the rules of simplified fuzzy reasoning can be expressed as the following:

$$R_j \ : \ if \ x_1 \ is \ M_{1j} \ and \ x_2 \ is \ M_{2j} \cdots x_m \ is \ M_{mj} \ then \ y^* \ is \ w_j, \quad (1)$$

where j ($j = 1, \cdots, n$) is a rule number, i ($i = 1, \cdots, m$) is a variable number, $M_{1j}, \cdots, M_{mj}$ are membership functions of the antecedent part, and w_j is the weights of the consequent part.

A membership value of the antecedent part μ_i is expressed as the following:

$$\mu_j = \prod_{i=1}^{m} M_{ij}(x_i), \quad (2)$$

where M_{ij} is membership function of antecedent part, and of the center value c_{ij} and the width b_{ij} are parameters. It is expressed as the following:

$$M_{ij}(x_i) = \begin{cases} 1 - \frac{2\cdot\left|x_i - c_{ij}\right|}{b_{ij}} & (c_{ij} - \frac{b_{ij}}{2} \le x_j \le c_{ij} + \frac{b_{ij}}{2}) \\ 0 & \text{(otherwise)} \end{cases} \quad (3)$$

The output y^* of fuzzy reasoning can be derived from the Eq.(4)

$$y^* = \frac{\sum_{j=1}^{n} \mu_j \cdot w_j}{\sum_{j=1}^{n} \mu_j}. \quad (4)$$

The value c_{ij}, b_{ij}, w_j are regarded as the parameters of the fuzzy reasoning model.

The objective function E is defined to evaluate the reasoning error between the desirable output y^r and the output y^* of fuzzy reasoning:

$$E = \frac{1}{2}(y^* - y^r)^2. \tag{5}$$

In order to minimize the objective function E, the parameters c_{ij}, b_{ij}, and w_j are updated based on the descent method as follows:

$$c_{ij}(t+1) = c_{ij}(t) - K_c \cdot \frac{\partial E}{\partial c_{ij}}, \tag{6}$$

$$b_{ij}(t+1) = b_{ij}(t) - K_b \cdot \frac{\partial E}{\partial b_{ij}}, \tag{7}$$

$$w_j(t+1) = w_j(t) - K_w \cdot \frac{\partial E}{\partial w_j}, \tag{8}$$

where t is iteration times and K_c, K_b, K_w are constants. The Eqs.(6),(7),(8) are calculated as follows:

$$\begin{aligned} c_{ij}(t+1) &= c_{ij}(t) - \frac{K_c \cdot \mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot \\ &\quad (w_j - y^*) \cdot \mathrm{sgn}(x_i - c_{ij}) \cdot \frac{2}{b_{ij} \cdot M_{ij}(x_i)}, \end{aligned} \tag{9}$$

$$\begin{aligned} b_{ij}(t+1) &= b_{ij}(t) - \frac{K_b \cdot \mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot \\ &\quad (w_j - y^*) \cdot \frac{1 - M_{ij}(x_i)}{M_{ij}(x_i)} \cdot \frac{1}{b_{ij}}, \end{aligned} \tag{10}$$

$$w_j(t+1) = w_j(t) - \frac{K_w \cdot \mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r), \tag{11}$$

where $\mathrm{sgn}(z)$ is shown in the following:

$$\mathrm{sgn}(z) = \begin{cases} -1 & ;\ z < 0 \\ 0 & ;\ z = 0 \\ 1 & ;\ z > 0 \end{cases} \tag{12}$$

In the fuzzy reasoning model using delta rule, the parameters c_{ij}, b_{ij}, w_j are updated according to Eqs.(9),(10),(11).

3. Fuzzy Reasoning Model with Regularization Term

3.1. Learning Rule

A learning algorithm added the regularization term is proposed. The generalized objective function S for fuzzy reasoning model is defined as follows:

$$S(\boldsymbol{w}) = \sum_{p} P(p) \cdot U_p(\boldsymbol{w}), \tag{13}$$

where $P(p)$ is the probability of occurrence for input-output data p, $\boldsymbol{w}$ is set of weight w_j.

This paper deals with the case where $P(p)$ is distributed uniformly, so $U_p(\boldsymbol{w})$ is only considered as an objective function. U_p is an objective function to be minimized for given input and output data. U_p is defined as follows:

$$U_p(\boldsymbol{w}) = E(\boldsymbol{w}) + \lambda G(\boldsymbol{w}), \tag{14}$$

where $E(\boldsymbol{w})$ is an error term shown as Eq.(5). $G(\boldsymbol{w})$ is the regularization term that denotes complexity measure of models. λ is a regularization parameter that is a small positive value. In the second term on the right-hand side in Eq.(14), we will consider the following two cases. In the first case, a multi-peak function is used and a cone-shaped function is used in the second case.

A. Multi-peak Function (Model I)

In this case, a multi-peak function is used as a regularization term. It is a function combined two normal distribution functions. A regularization term G is defined as follows:

$$G_{MP}(\boldsymbol{w}) = \sum_{j=1}^{n} \left(0.5 q_1(w_j) + 0.5 q_2(w_j)\right), \tag{15}$$

$$q_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right) \quad (\ k = \{1,2\}\ ), \tag{16}$$

where $\sigma_1 = \sigma_2 = 0.2$, $\mu_1 = -0.5$ and $\mu_2 = 0.5$. Parameters w_j is updated as follows:

$$w_j(t+1) = w_j(t) - K_w' \cdot \frac{\partial U_p}{\partial w_j}, \tag{17}$$

where K_w' is constants, t is iteration times. Partial differential term is calculated as follows:

$$\begin{aligned} \frac{\partial U_p}{\partial w_j} &= \frac{\mu_j}{\sum_{k=1}^{n} \mu_k} \cdot (y^* - y^r) + \lambda \Big(-\frac{w_j - \mu_1}{2\sqrt{2\pi}\sigma_1^3} \cdot \\ &\quad \exp(-\frac{(w_j-\mu_1)^2}{2\sigma_1^2}) - \frac{w_j - \mu_2}{2\sqrt{2\pi}\sigma_2^3} \cdot \exp(-\frac{(w_j-\mu_2)^2}{2\sigma_2^2}) \Big). \end{aligned} \tag{18}$$

B. Cone-shaped Function (Model II)

In this case, a cone-shaped function is used as a regularization term. G is defined as follows:

$$G_{CS}(\boldsymbol{w}) = \sum_{j=1}^{n} \frac{(\frac{w_j}{w_0})^2}{1 + (\frac{w_j}{w_0})^2}, \tag{19}$$

where w_0 is a fixed value as the normalization factor. In this case, the update equation becomes as follows:

$$\begin{aligned} \frac{\partial U_p}{\partial w_j} &= \frac{\mu_j}{\sum_{k=1}^{n} \mu_k} \cdot (y^* - y^r) + \\ &\quad \lambda \cdot \frac{2\frac{w_j}{w_0^2}}{(1 + (\frac{w_j}{w_0})^2)^2}. \end{aligned} \tag{20}$$

3.2. Regularization Parameter λ

Coefficient λ is regularization parameter for determining ratio in considering both terms that are first term and second term in Eqs.(18),(20). A following monotonically decreasing function is used for λ.

$$\lambda(t) = \lambda_0 \times (1 - \frac{t}{T_{max}}), \tag{21}$$

where t is iteration times and λ_0 is initial value of λ and T_{max} is maximum learning times for termination.

3.3. Learning Algorithm

A flow of learning algorithm is shown in the following:

[STEP 1]

The initial number of rules, the central coordinates of the membership function c_{ij}, the base width of the membership function b_{ij} and weight w_j are set randomly.

The threshold θ_1 of reasoning error and the threshold θ_2 of change of rate of reasoning error are given. Input and output data $(x_1^p, \cdots, x_m^p, y_p^r)$ for $p = 1, \cdots, P$ is selected randomly. λ_0 is set to a fixed number.

[STEP 2]

Let $t = 1$. t is iteration times.

[STEP 3]

Let $p = 1$.

[STEP 4]

Input and output data $(x_1^p, \cdots, x_m^p, y_p^r)$ is given.

[STEP 5]

Fitness of each rule is calculated by Eqs.(2), (3).

[STEP 6]

Reasoning output y_p^* is calculated by Eq.(4).

[STEP 7]

Weight w_j is updated by Eqs.(17),(18), (20).

[STEP 8]

c_{ij} and b_{ij} are updated by Eqs.(9), (10).

[STEP 9]

If $p = P$ then go to STEP 10.

If $p < P$ then go to STEP 4 with $p \leftarrow p + 1$.

[STEP 10]

Reasoning error $E_r(t)$ is calculated, where

$$E_r(t) = \frac{1}{P} \sum_{p=1}^{P} |y^{*p} - y^{rp}|. \tag{22}$$

If $E_r(t) \le \theta_1$ then go to STEP 11.

If $E_r(t) > \theta_1$ then go to STEP 3 with $t \leftarrow t + 1$.

[STEP 11]

The rate of change of reasoning error $\triangle E_r(t)$ and threshold θ_2 are compared, where $\triangle E_r(t)$ is defined as Eq.(23). Then

if $\triangle E_r(t) \le \theta_2$ then learning is completed.

If $\triangle E_r(t) > \theta_2$ then go to STEP 3 with $t \leftarrow t + 1$.

$$\Delta E_r(t) = |E_r(t) - E_r(t-1)|. \tag{23}$$

Learning algorithm is operated according to the above procedure.

4. Numerical Experiment

4.1. Function Approximation

We perform an experiment to show the validity of the proposed method using learning rule described in the previous section. We perform function approximation to investigate basic feature of the proposed method and compare the performance of the proposed method with the delta rule model. Here are two systems which is specified by the following functions:

$$(a)\ \ y = \frac{\sin(\exp(3x)) + 1}{2}, \tag{24}$$

$$(b)\ \ y = \{\frac{\sin(\sqrt{(20x_1 - 10)^2 + (20x_2 - 10)^2})}{\sqrt{(20x_1 - 10)^2 + (20x_2 - 10)^2}} + 0.22\}/1.22. \tag{25}$$

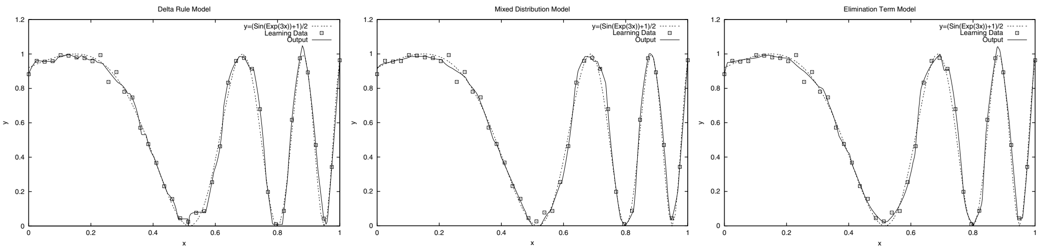
The output data for training are set to add noise that conform to the normal distribution with $\mu = 0, \sigma^2 = 0.1$, after y is calculated by Eqs.(24),(25). The conditions of the simulation are set as the following table (Table 1).

Table 1: Simulation conditions

	4.1 function approx.		4.2 torus
	(a)	(b)	
θ_1	6.0×10^{-3}	3.0×10^{-2}	3.0×10^{-2}
θ_2	2.0×10^{-7}	3.0×10^{-6}	3.0×10^{-6}
K_c	0.2	0.07	0.07
K_b	0.3	0.06	0.06
$K_w(K'_w)$	0.7	0.5	0.5
λ_0	0.001	0.01	0.01
T_{max}	1000	10000	10000
# rules	22	25	25
# training data	40	225	200
# test data	400	2000	2500
initial c_{ij}, b_{ij}	random	random	random
initial b_{ij}	0.1	0.7	0.1

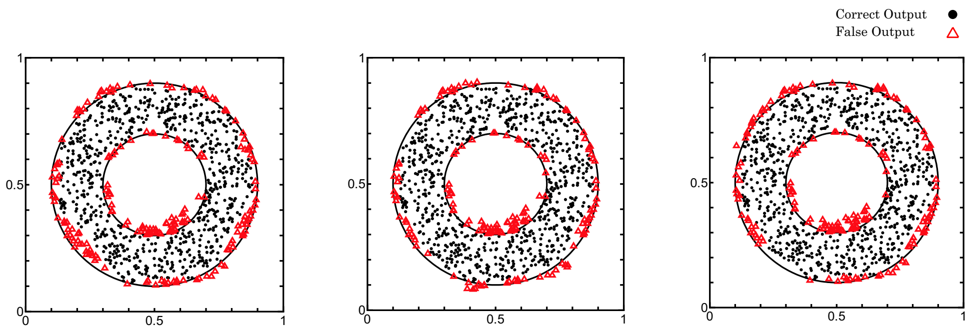
Fig.(1) shows the reasoning output of system (a) for test data. A dotted line denotes the function shown in Eq.(24) and '□' mark denotes learning data and a solid line denotes the reasoning output for test data. On the delta rule model in Fig.(a), the reasoning output are over fitted to the learning data. On the other hand, the reasoning output of proposed model in Fig.(b),(c) is smooth.

Table 2 shows the result of function approximation. Each value in the table expresses average of ten trials. Mean square error (MSE) in the table denotes an error for test data. MSE and iteration times are improved in comparison with the delta rule model. Especially, the Model II shows good result.



(a) Delta Rule Model (b) Model I (c) Model II

Figure 1: Result of function approximation.



(a) Delta Rule Model (b) Model I (c) Model II

Figure 2: Result of torus division problem.

4.2. Torus Division Problem

This problem is that points on $[0,1]\times[0,1]$ are classified into two classes with torus inside and outside. The inside data and outside data for torus are assigned by class 1 and class 0, respectively. The center of torus is $(0.5, 0.5)$, with a minor diameter of 0.4 and a major diameter of 0.8. The conditions of the simulation are shown in Table 1. The results of reasoning output are shown in Table 3. Each value in the table expresses average of ten trials. From the results, MSE is small compared with the delta rule model. The results of reasoning output using constructed rules are shown in Fig.2. The mark of '•' denotes reasoning output discriminated correctly, and '△' denotes discriminated incorrectly. From the figure, incorrect point is great by delta rule model within torus, however, it is improved by proposed models. In the simulation, the "Model II" was more effective than the "Model I". It considers that this reason is because "Model II" has a control function that w_j is approached to small value.

Table 2: Result of function approximation.

		Delta Rule Model	Proposed Model	
			Model I	Model II
(a)	MSE($\times10^{-4}$)	3.8	3.6	3.4
	Iteration	1834.7	994.7	620.2
(b)	MSE($\times10^{-4}$)	10.6	9.8	7.6
	Iteration	3595.1	2501.6	1148.8

Table 3: Result of torus division problem.

	Delta Rule Model	Proposed Model	
		Model I	Model II
MSE($\times10^{-2}$)	5.4	4.6	4.0
Iteration	5834.6	1458.1	1482.0

5. Conclusion

In this paper, we proposed a self-tuning fuzzy system adopt two types of regularization terms to the objective function. It was found through numerical experiments that the proposed method shows good result. On the function approximation, the function became smooth in comparison with the delta rule model. The performance of the proposed method was superior to the delta rule model.

References

[1] M.M. Gupta, L. Jin, and N. Homma, *"Static and Dynamic Neural Networks,"* IEEE Press, 2003.

[2] H. Miyajima, et al., "Constructive, Destructive and Simplified Learning Methods of Fuzzy Inference," *IEICE Trans. Fundamentals*, vol.E78-A, no.10, pp.1331–1338, 1995.

[3] S. Fukumoto, et al., "A Learning Method of Fuzzy Reasoning Model with Regularization," *Proc. of International Workshop of Fuzzy Systems & Innovational Computing 2004*, pp.275–279, 2004.