Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Fuzzy Sets and Systems 152 (2005) 535–551

FUZZY sets and systems

www.elsevier.com/locate/fss

An asymmetry-similarity-measure-based neural fuzzy inference system

Cheng-Jian Lin*, Wen-Hao Ho

Department of Computer Science and Information Engineering, Chaoyang University of Technology, No. 168, Jifong E. Rd., Wufong Township, Taichung County 41349, Taiwan

Received 10 December 2002; received in revised form 25 March 2004; accepted 5 November 2004
Available online 8 December 2004

Abstract

In this paper, a new asymmetry-similarity-measure-based neural fuzzy inference system (ASM-NFIS) is proposed. A pseudo-Gaussian membership function can provide a neural fuzzy inference system which has a higher flexibility and can approach the optimized result more accurately. An on-line self-constructing learning algorithm is proposed to automatically construct the ASM-NFIS. It consists of structure learning and parameter learning that would create adaptive fuzzy logic rules. The structure learning is based on the similarity measure of asymmetric Gaussian membership functions, and the parameter learning is based on a supervised gradient descent method. Computer simulations were conducted to illustrate the performance and applicability of the proposed model.

Keywords: Asymmetric fuzzy similarity measure; Pseudo-Gaussian; Backpropagation; Prediction

1. Introduction

The main purpose of a fuzzy system is to achieve a set of local input–output relationships that describe a process. As is well known, the problem of system modeling requires two main stages: structure identification and parameter optimization. Structure identification deals with the problem of determining the input–output space partition and how many rules must be used by the fuzzy system. Parameter optimization finds the optimum value of all the parameters involved in the fuzzy system; that is, it locates the membership functions in the premise and consequent of each rule [1–4,6–8,10].

* Corresponding author. Fax: +886 43742375.
E-mail address: cjlin@mail.cyut.edu.tw (C.-J. Lin).

0165-0114/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2004.11.001

To prevent a newly generated membership function from being too similar to an existing membership function, the similarity measure has been widely researched and broadly applied [2,7,8,10]. They adopt the traditional symmetric Gaussian membership functions. Recently, many researchers [11,12] use the asymmetric Gaussian membership function, which is called Pseudo-Gaussian (PG), to act as the input term node. Because the asymmetric Gaussian membership function's variability and malleability are higher than those of the traditional membership function, the PG membership function can provide a neural fuzzy inference system which has a higher flexibility and can approach the true result more easily. In [11], the partitioning of input space is performed in advance; namely the number of fuzzy rules will be pre-set by the users. Then, an online learning algorithm is proposed to construct the fuzzy systems dynamically in order to overcome the aforementioned drawback. We also develop a new asymmetric similarity measure to check the similarities between a new membership function and existing ones. Therefore, in this paper the proposed asymmetric similarity measure method is different from the traditional symmetric similarity measure method [2,7,8,10].

In this paper, we present an asymmetry-similarity-measure-based neural fuzzy inference system (ASM-NFIS). It is a standard four-layer feedforward neural network. An on-line learning algorithm is proposed to automatically construct the ASM-NFIS. It consists of structure learning and parameter learning. We will add a new node to satisfy the fuzzy partitioning of the training data in structure learning. The similarity measure of asymmetric Gaussian membership functions is proposed to estimate the rule's similarity degree. The back-propagation learning is then used for tuning input/output membership functions. This method has the advantage of not requiring an expert's assistance since the input–output characteristics of the ASM-NFIS and its structure are obtained from the training examples. This is in contrast to [11] and [12], which the input space needs to be divided properly in advance. The proposed model has been used to identify the dynamic system and predict the chaotic time-series. The simulation results show that the proposed ASM-NFIS model has a better learning performance than other learning systems.

2. The structure of ASM-NFIS

The jth fuzzy if-then rule shown below is used by the ASM-NFIS:

$$\begin{aligned} R_j :\ & \text{IF } x_1 \text{ is } A_1^j \text{ and } \ldots\ldots \text{ and } x_n \text{ is } A_n^j, \\ & \text{THEN } y = b_j, \end{aligned} \tag{1}$$

where x_i and y are the input and output variables, respectively; A_i^j is the linguistic term of the precondition part with membership function $\mu_{A_i^j}$; b_j is the constant consequent; and n is the number of input dimensions.

The membership function of the precondition part discussed in this paper is different from the typical Gaussian membership function. We adopt the Pseudo-Gaussian (PG) membership function [11] to approximate desired results. The definition of PG membership function is as follows:

$$\mu_{A_i^j}(x_i) = \exp\left(-\frac{(x_i - m_{ji})^2}{\sigma_{ji,-}^2}\right) U(x_i; -\infty, m_{ji}) + \exp\left(-\frac{(x_i - m_{ji})^2}{\sigma_{ji,+}^2}\right) U(x_i; m_{ji}, \infty), \tag{2}$$

$$\text{where } U(x_i; a, b) = \begin{cases} 1 & \mathit{if}\ a \leqslant x_i < b \\ 0 & \mathit{otherwise}, \end{cases}$$

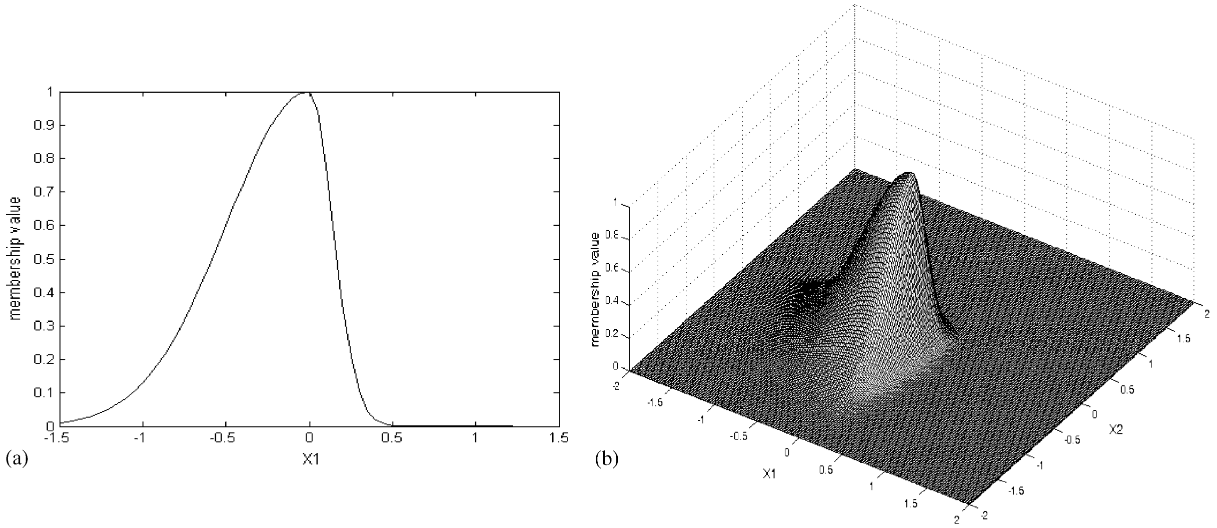


Fig. 1. (a) One-dimensional PG, (b) two-dimensional PG.

where m_{ji} is the mean of the PG membership function; $\sigma_{ji,-}$ is the negative deviation of the PG membership function; and $\sigma_{ji,+}$ is the positive deviation of the PG membership function. The PG membership function is asymmetric and has great flexibility. Fig. 1 shows the one- and two-dimensional PG membership functions.

A typical network consists of nodes with a finite number of fan-in connections from other nodes represented by weight values and a finite number of fan-out connections to other nodes. Associated with the fan-in of a node is an integration function which combines information, activation, or evidence from other nodes and provides the net input, i.e.,

$$\text{net input} = f(z_1^{(k)}, z_2^{(k)}, \ldots, z_p^{(k)}; w_1^{(k)}, w_2^{(k)}, \ldots, w_p^{(k)}), \tag{3}$$

where $z_i^{(k)}$ is the ith input variable to a node in layer k and $w_i^{(k)}$ is the weight of the associated link. The superscript in the above equation indicates the layer number. This notation will be also used in the following equations. Each node also outputs an activation value as a function of its net input

$$\text{output} = a[f(\cdot)], \tag{4}$$

where $a(\cdot)$ denotes the activation function.

The ASM-NFIS is a standard four-layer network [8], as shown in Fig. 2, where the functions of the nodes in each layer are described as follows:

Layer 1: The nodes in this layer are input nodes (i.e., input-linguistic nodes) which represent input-linguistic variables and which pass input signals to the next layer directly:

$$f(x_i^{(1)}) = x_i^{(1)} \tag{5}$$

and

$$a[f(\cdot)] = f(\cdot), \tag{6}$$

where $x_i^{(1)}$ is the ith input variable to a node in layer 1.

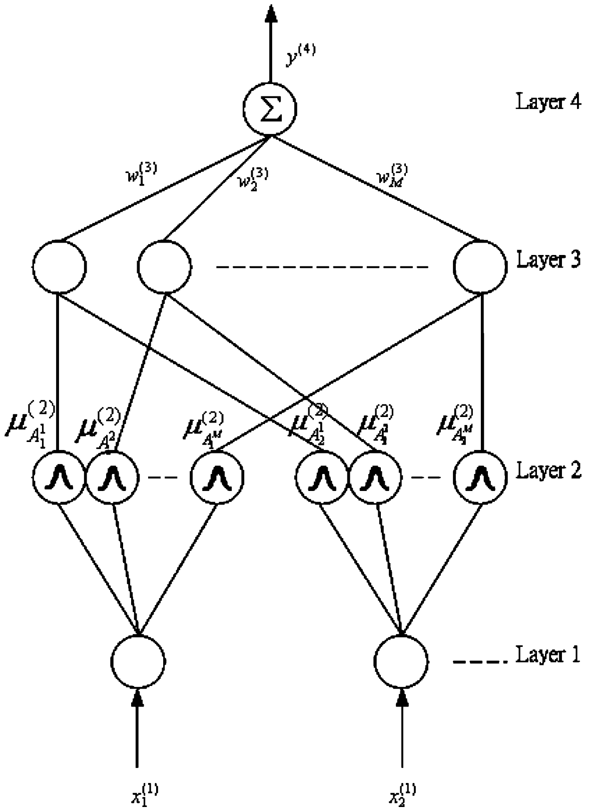


Fig. 2. The structure of ASM-NFIS.

Layer 2: The nodes in this layer are term nodes that act as the PG membership function. They can react to the terms of the respective input-linguistic variables. For the jth rule node,

$$f(z_i^{(2)}) = \exp\left(-\frac{(z_i^{(2)} - m_{ji})^2}{\sigma_{ji,-}^2}\right) U(z_i^{(2)}; -\infty, m_{ji}) + \exp\left(-\frac{(z_i^{(2)} - m_{ji})^2}{\sigma_{ji,+}^2}\right) U(z_i^{(2)}; m_{ji}, \infty), \tag{7}$$

where $U(z_i^{(2)}; a, b) = \begin{cases} 1 & \textit{if } a \leqslant z_i^{(2)} < b \\ 0 & \textit{otherwise} \end{cases}$

and

$$a[f(\cdot)] = f(\cdot). \tag{8}$$

The m_{ji}, $\sigma_{ji,-}$, and $\sigma_{ji,+}$ are the mean, negative deviation, and positive deviation, respectively of the PG membership functions of jth term associated with ith input variable x_i.

Layer 3: The nodes in this layer are compensatory fuzzy nodes. They represent the precondition part of the fuzzy logic rules which can input the multiple incoming signals and output the product result. For the jth rule node,

$$f(z_i^{(3)}) = \prod_i^n z_i^{(3)} \tag{9}$$

and

$$a[f(\cdot)] = f(\cdot) \tag{10}$$

where n is dimension number.

Layer 4: The nodes in this layer are denoted by Σ. That is, it receives the multiple incoming signals and outputs the result of summation. For the output y,

$$f(z_i^{(4)}) = \sum_j^M w_j^{(3)} z_i^{(4)} \tag{11}$$

and

$$a[f(\cdot)] = f(\cdot) \tag{12}$$

where M is the rule number and $w_j^{(3)}$ is the link weight.

3. The on-line learning algorithm

In this section, we propose an online learning algorithm which consists of the structure learning algorithm and the parameter learning algorithm. The structure learning algorithm is used to find proper fuzzy partitions in the input space and create fuzzy logic rules. The asymmetric similarity measure method is used to prevent the newly generated membership function from being too similar to the existing membership function. The parameter learning algorithm is the most general supervised learning scheme and is used to adjust the PG membership functions in the precondition part and to modify the link weight in the consequent part. As a result, the parameter learning algorithm is based on the back-propagation algorithm, which minimizes the cost function to approximate the desired results. The procedure of the structure and parameter learning algorithms is through inputting the training pattern to learn successively.

3.1. The structure learning algorithm

The proposed structure learning algorithm decides the proper fuzzy partitions by using the input patterns. The procedure of the structure learning algorithm uses the PG membership functions to find the fuzzy logic rules. However, the structure learning algorithm determines whether to add a new node in layer 2 via the input pattern data and whether to add the associated fuzzy logic rule in layer 3.

After the input pattern is entered in layer 2, the firing strength of the PG membership function will be obtained from Eq. (2), which is used to calculate the degree measure $\mu_{A_i^j}$. In layer 3, the firing

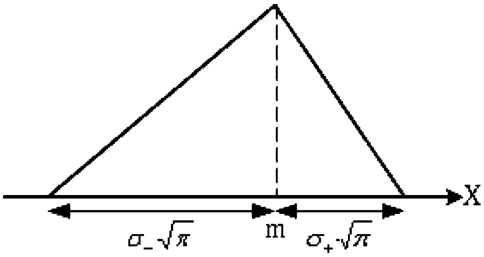


Fig. 3. An asymmetric triangle with the unity height and the length of bottom edge.

strength of the fuzzy logic rule is obtained from Eq. (4), used to obtain the degree measure of the precondition part

$$P_j = \prod_{i=1}^{n} \mu_{A_i^j}(x_i), \quad j = 1, 2, \ldots, M(t), \tag{13}$$

where $M(t)$ is the number of existing rules at time t. According to the degree measure, we can obtain the existing maximum P_j of the firing strength of the fuzzy logic rule

$$P_{\max} = \max_{1 \leqslant j \leqslant M(t)} P_j. \tag{14}$$

IF $P_{\text{th}} > P_{\max}$, the structure learning needs to add a new node in the ASM-NFIS. The P_{th} is the preset threshold, which should be decreased when the structure learning algorithm limits the rule of ASM-NFIS. The P_{th} is an important parameter and its value is set between zero and one. A low P_{th} value leads to the learning of coarse clusters, whereas a high P_{th} value leads to the learning of fine clusters. The new mean, positive deviation, and negative deviation of the PG membership function are preset values according to the input pattern or heuristic. The process of how the node increases is shown as follows:

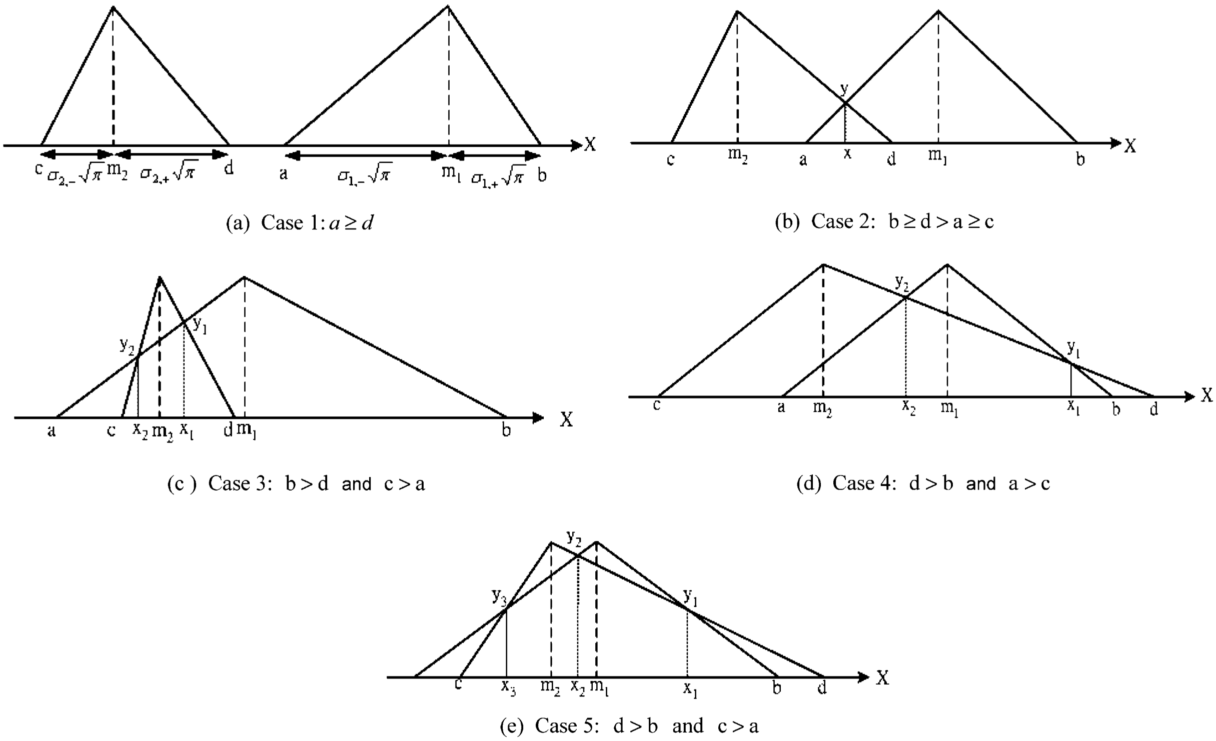
$$m_{ji}^{(\text{new})} = x_i, \tag{15}$$

$$\sigma_{ji,-}^{(\text{new})} = \sigma_{ji,+}^{(\text{new})} = \sigma_i, \tag{16}$$

where x_i is the new input pattern; σ_i is the preset constant.

To prevent the newly generated membership function from being too similar to the existing membership function, the similarities between the new membership function and the existing functions must be checked. If the new fuzzy rule is different from the existing fuzzy rule, we confirm that the new fuzzy rule will be added in the ASM-NFIS. It can cause the neural fuzzy inference system to perform better. Therefore, we use the similarity measure of asymmetric Gaussian membership functions to estimate the rule's similarity degree.

Since the area of the asymmetric Gaussian membership function, calculated from Eq. (2), is between $\sigma_+\sqrt{\pi}$ and $\sigma_-\sqrt{\pi}$, the height is always 1, and the center of the bottom-line at m_i is on the x-axis. We can approximate it by an asymmetric triangle $\Delta(m_i, \sigma_{i,-}, \sigma_{i,+})$ with a unity height and with the length of bottom edge $\sigma_+\sqrt{\pi} + \sigma_-\sqrt{\pi}$ (see Fig. 3). Assume that the two end-points of the bottom line of $\Delta(m_1, \sigma_{1,-}, \sigma_{1,+})$ are a and b on the x-axis, and another end-points of $\Delta(m_2, \sigma_{2,-}, \sigma_{2,+})$ are c and d on the x-axis. That is, $a = m_1 - \sigma_{1,-}\sqrt{\pi}$, $b = m_1 + \sigma_{1,+}\sqrt{\pi}$, $c = m_2 - \sigma_{2,-}\sqrt{\pi}$, and $d = m_2 + \sigma_{2,+}\sqrt{\pi}$.



(a) Case 1: $a \geq d$

(b) Case 2: b ≥ d > a ≥ c

(c) Case 3: b > d and c > a

(d) Case 4: d > b and a > c

(e) Case 5: d > b and c > a

Fig. 4. The five possible situations between two asymmetric triangles: (a) *Case* 1: $a \geqslant d$, (b) *Case* 2: b ⩾ d > a ⩾ c (c) *Case* 3: b > d and c > a (d) *Case* 4: d > b and a > c (e) *Case* 5: d > b and c > a.

First, if $m_1 = m_2$, when $a \geqslant c$ and $b \leqslant d$, $|A \cap B| = \frac{1}{2}(\sigma_{2,+} + \sigma_{2,-})\sqrt{\pi}$, $c \geqslant a$ and $d \leqslant b$, $|A \cap B| = \frac{1}{2}(\sigma_{1,+} + \sigma_{1,-})\sqrt{\pi}$, $a \geqslant c$ and $b \geqslant d$, $|A \cap B| = \frac{1}{2}(\sigma_{2,+} + \sigma_{1,-})\sqrt{\pi}$, $a \leqslant c$ and $b \leqslant d$, $|A \cap B| = \frac{1}{2}(\sigma_{1,+} + \sigma_{2,-})\sqrt{\pi}$.

In the following discussion, we assume that $m_1 > m_2$. Let us consider the following five possible situations (see Fig. 4):

Case 1: If $a \geqslant d$, then $|A \cap B| = 0$, since the two membership functions do not overlap.

Case 2: If $b \geqslant d > a \geqslant c$, then

$$\begin{aligned} |A \cap B| &= \frac{1}{2}(d-a)y \\ &= \frac{1}{2}\,\frac{(m_2 + \sigma_{2,+}\sqrt{\pi} - m_1 + \sigma_{1,-}\sqrt{\pi})^2}{\sigma_{1,-}\sqrt{\pi} + \sigma_{2,+}\sqrt{\pi}}. \end{aligned} \tag{17}$$

Case 3: If $b > d$ and $c > a$, then

$$\begin{aligned} |A \cap B| &= \frac{1}{2}(x_2 - c)y_2 + \frac{1}{2}(y_1 + y_2)\cdot(x_1 - x_2) + \frac{1}{2}(d - x_1)y_1 \\ &= \frac{1}{2}\,\frac{(m_2 - \sigma_{2,-}\sqrt{\pi} - m_1 + \sigma_{1,-}\sqrt{\pi})^2}{-\sigma_{1,-}\sqrt{\pi} + \sigma_{2,-}\sqrt{\pi}} + \frac{1}{2}\,\frac{(m_2 + \sigma_{2,+}\sqrt{\pi} - m_1 + \sigma_{1,-}\sqrt{\pi})^2}{\sigma_{1,-}\sqrt{\pi} + \sigma_{2,+}\sqrt{\pi}}. \end{aligned} \tag{18}$$

Case 4: If $d > b$ and $a > c$, then

$$\begin{aligned}|A \cap B| &= \frac{1}{2}(x_2 - a)y_2 + \frac{1}{2}(y_1 + y_2)\cdot(x_1 - x_2) + \frac{1}{2}(b - x_1)y_1 \\ &= \frac{1}{2}\,\frac{(m_2 + \sigma_{2,+}\sqrt{\pi} - m_1 + \sigma_{1,-}\sqrt{\pi})^2}{\sigma_{1,-}\sqrt{\pi} + \sigma_{2,+}\sqrt{\pi}} + \frac{1}{2}\,\frac{(-m_2 - \sigma_{2,+}\sqrt{\pi} + m_1 + \sigma_{1,+}\sqrt{\pi})^2}{\sigma_{1,+}\sqrt{\pi} - \sigma_{2,+}\sqrt{\pi}}. \end{aligned} \tag{19}$$

Case 5: If $d > b$ and $c > a$, then

$$\begin{aligned}|A \cap B| &= \frac{1}{2}(x_3 - c)y_3 + \frac{1}{2}(y_2 + y_3)\cdot(x_2 - x_3) + \frac{1}{2}(y_1 + y_2)(x_1 - x_2) + \frac{1}{2}(b - x_1)y_1 \\ &= \frac{1}{2}\,\frac{(m_2 - \sigma_{2,-}\sqrt{\pi} - m_1 + \sigma_{1,-}\sqrt{\pi})^2}{-\sigma_{1,-}\sqrt{\pi} + \sigma_{2,-}\sqrt{\pi}} + \frac{1}{2}\cdot\frac{(m_2 + \sigma_{2,+}\sqrt{\pi} - m_1 + \sigma_{1,-}\sqrt{\pi})^2}{\sigma_{1,-}\sqrt{\pi} + \sigma_{2,+}\sqrt{\pi}} \\ &\quad + \frac{1}{2}\,\frac{(-m_2 - \sigma_{2,+}\sqrt{\pi} + m_1 + \sigma_{1,+}\sqrt{\pi})^2}{\sigma_{1,+}\sqrt{\pi} - \sigma_{2,+}\sqrt{\pi}}. \end{aligned} \tag{20}$$

We can conclude a general formula for $|A \cap B|$:

$$\begin{aligned}|A \cap B| &= \frac{1}{2}\,\frac{h^2(m_2 + \sigma_{2,+}\sqrt{\pi} - m_1 + \sigma_{1,-}\sqrt{\pi})}{\sigma_{1,-}\sqrt{\pi} + \sigma_{2,+}\sqrt{\pi}} + \frac{1}{2}\,\frac{h^2(m_2 - \sigma_{2,-}\sqrt{\pi} - m_1 + \sigma_{1,-}\sqrt{\pi})}{-\sigma_{1,-}\sqrt{\pi} + \sigma_{2,-}\sqrt{\pi}} \\ &\quad + \frac{1}{2}\,\frac{h^2(m_2 + \sigma_{2,+}\sqrt{\pi} - m_1 - \sigma_{1,+}\sqrt{\pi})}{\sigma_{1,+}\sqrt{\pi} - \sigma_{2,+}\sqrt{\pi}}, \end{aligned} \tag{21}$$

where $h(x) = \max\{0, x\}$. Thus, the approximate similarity measure of fuzzy sets is

$$\begin{aligned}E(A, B) &= \frac{|A \cap B|}{|A \cup B|} \\ &= \frac{|A \cap B|}{\frac{1}{2}\,\sigma_{1,+}\sqrt{\pi} + \frac{1}{2}\,\sigma_{1,-}\sqrt{\pi} + \frac{1}{2}\,\sigma_{2,+}\sqrt{\pi} + \frac{1}{2}\,\sigma_{2,-}\sqrt{\pi} - |A \cap B|}. \end{aligned} \tag{22}$$

The similarity measure E between the new membership function and all existing membership functions are calculated, and the maximum E, $E_{\max}$, is calculated as follows:

$$E_{\max} = \max_{1 \leqslant j \leqslant M(t)} E(\mu(m_1^{(\mathrm{new})}, \sigma_{1,+}^{(\mathrm{new})}, \sigma_{1,-}^{(\mathrm{new})}), \mu(m_1^j, \sigma_{1,+}^j, \sigma_{i,-}^j)). \tag{23}$$

If $E_{\max} \leqslant E_{\mathrm{th}}$, where $E_{\mathrm{th}} \in (0, 1)$ is a prespecified threshold, then the new fuzzy logic rule is adopted and the rule number is incremented.

$$M = M + 1. \tag{24}$$

Therefore, the new mean, deviation, and link weight are generated randomly.

3.2. The parameter learning algorithm

After the structure network has been accordingly adjusted to the current training pattern, the network enters the parameter learning stage. The parameter learning algorithm adjusts the parameter of ASM-NFIS optimally with the same training pattern. The back-propagation phase is used for this supervised learning to find the output errors of the node in each layer and to analyze the error in order to adjust the parameter. The goal is to minimize the error function

$$E = \frac{1}{2}(y^d(t) - y(t))^2, \tag{25}$$

where $y^d(t)$ is the desired output and $y(t)$ is the ASM-NFIS output (Fig. 5).

The parameter learning algorithm based on back-propagation is then as follows:

Assuming that w is the adjustable parameter in a node, the generally-used learning rule is

$$w(t+1) = w(t) - \eta\left(\frac{\partial E}{\partial w}\right), \tag{26}$$

$$\begin{aligned}\frac{\partial E}{\partial w} &= \frac{\partial E}{\partial f}\frac{\partial f}{\partial w},\\ &= \frac{\partial E}{\partial a}\frac{\partial a}{\partial f}\frac{\partial f}{\partial w},\end{aligned} \tag{27}$$

where η is the learning rate.

To show the learning rules, we derive the parameter learning layer by layer.

Layer 4: There is no parameter to be adjusted in this layer.

Layer 3: Using (13) and (14), the link weight is adjusted by the amount

$$w_j^{(3)}(t+1) = w_j^{(3)}(t) - \eta_{w^{(3)}}\left(\frac{\partial E}{\partial w_j^{(3)}}\right), \tag{28}$$

$$\begin{aligned}\frac{\partial E}{\partial w_j^{(3)}} &= -\frac{\partial E}{\partial y^{(4)}}\frac{\partial y^{(4)}}{\partial w_j^{(3)}}\\ &= (y^{(4)} - y^d)p_j^{(3)}.\end{aligned} \tag{29}$$

Layer 2: The m_{ji} of the PG membership function is adjusted by the amount

$$m_{ji}(t+1) = m_{ji}(t) - \eta_m\left(\frac{\partial E}{\partial m_{ji}}\right) \tag{30}$$

$$\begin{aligned}\frac{\partial E}{\partial m_{ji}} &= \left[\frac{\partial E}{\partial y^{(4)}}\right]\left[\frac{\partial y^{(4)}}{\partial P_j^{(3)}}\right]\left[\frac{\partial P_j^{(3)}}{\partial \mu_{A_i^j}^{(2)}}\right]\left[\frac{\partial \mu_{A_i^j}^{(2)}}{\partial m_{ji}}\right]\\ &= (y^{(4)} - y^d)w_j^{(3)}\prod_{\substack{l=1\\ l\neq i}}^{n}\mu_{A_l^j}^{(2)}\\ &\quad\cdot\left[\frac{2(x_i^{(1)} - m_{ji})}{\sigma_{ji,-}^2}\exp\left(-\frac{(x_i^{(1)} - m_{ji})^2}{\sigma_{ji,-}^2}\right)U(x_i^{(1)}; -\infty, m_{ji})\right.\\ &\quad\left.+\frac{2(x_i^{(1)} - m_{ji})}{\sigma_{ji,+}^2}\exp\left(-\frac{(x_i^{(1)} - m_{ji})^2}{\sigma_{ji,+}^2}\right)U(x_i^{(1)}; m_{ji}, \infty)\right]\end{aligned} \tag{31}$$

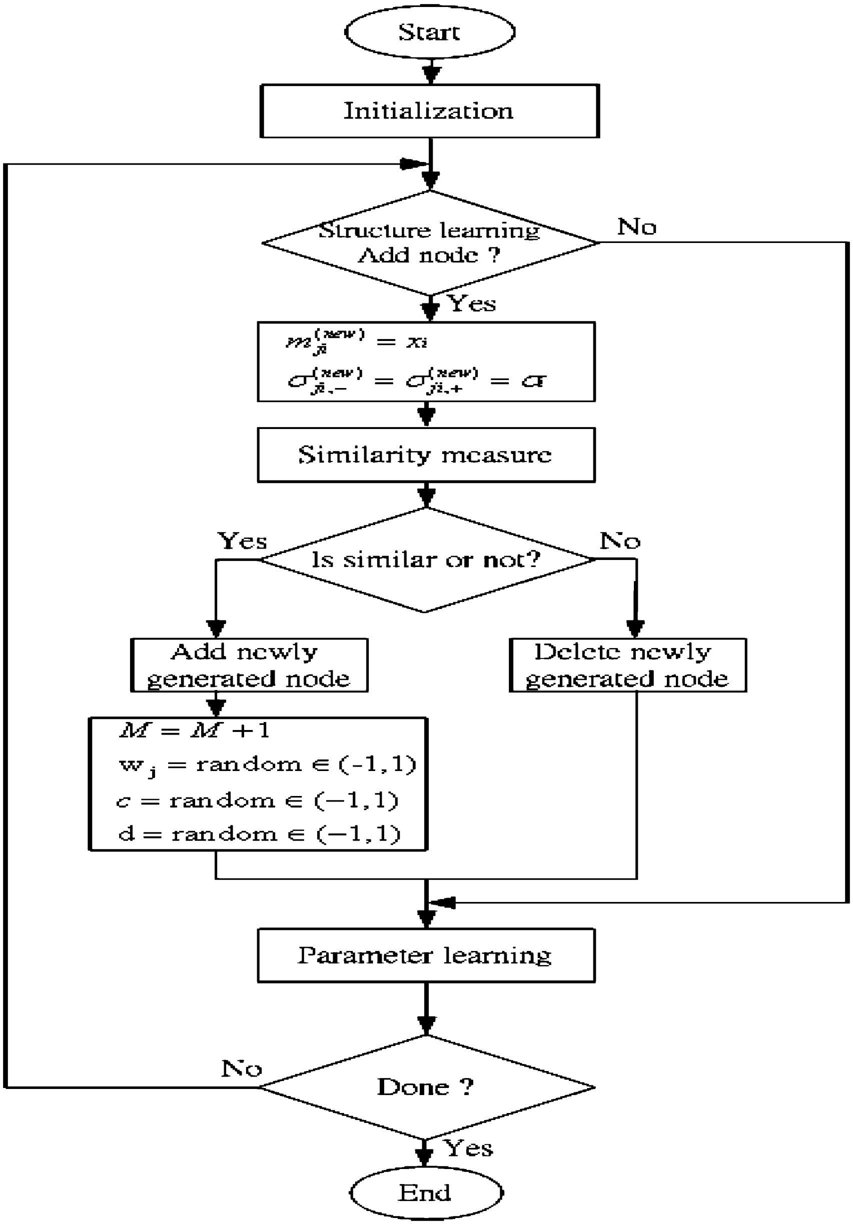


Fig. 5. The flow diagram of the structure/parameter learning for the ASM-NFIS.

The $\sigma_{ji,-}$ of the PG membership function is adjusted by the amount

$$\sigma_{ji,-}(t+1) = \sigma_{ji,-}(t) - \eta_{\sigma_-}\left(\frac{\partial E}{\partial \sigma_{ji,-}}\right) \tag{32}$$

$$\frac{\partial E}{\partial \sigma_{ji,-}} = \left[\frac{\partial E}{\partial y^{(4)}}\right]\left[\frac{\partial y^{(4)}}{\partial P_j^{(3)}}\right]\left[\frac{\partial P_j^{(3)}}{\partial \mu_{A_i^j}^{(2)}}\right]\left[\frac{\partial \mu_{A_i^j}^{(2)}}{\partial \sigma_{ji,-}}\right]$$
$$= (y^{(4)} - y^d) w_j^{(3)} \prod_{\substack{l=1 \\ l\neq i}}^{n} \mu_{A_l^j}^{(2)}$$
$$\times \left[\frac{2(x_i^{(1)} - m_{ji})^2}{\sigma_{ji,-}^3} \exp\left(-\frac{(x_i^{(1)} - m_{ji})^2}{\sigma_{ji,-}^2}\right) U(x_i^{(1)}; -\infty, m_{ji})\right]. \tag{33}$$

The $\sigma_{ji,+}$ of the PG membership function is adjusted by the amount

$$\sigma_{ji,+}(t+1) = \sigma_{ji,+}(t) - \eta_{\sigma+}\left(\frac{\partial E}{\partial \sigma_{ji,+}}\right) \tag{34}$$

$$\frac{\partial E}{\partial \sigma_{ji,+}} = \left[\frac{\partial E}{\partial y^{(4)}}\right]\left[\frac{\partial y^{(4)}}{\partial P_j^{(3)}}\right]\left[\frac{\partial P_j^{(3)}}{\partial \mu_{A_i^j}^{(2)}}\right]\left[\frac{\partial \mu_{A_i^j}^{(2)}}{\partial \sigma_{ji,+}}\right]$$
$$= (y^{(4)} - y^d) w_j^{(3)} \prod_{\substack{l=1 \\ l\neq i}}^{n} \mu_{A_l^j}^{(2)}$$
$$\times \left[\frac{2(x_i^{(1)} - m_{ji})^2}{\sigma_{ji,+}^3} \exp\left(-\frac{(x_i^{(1)} - m_{ji})^2}{\sigma_{ji,+}^2}\right) U(x_i^{(1)}; m_{ji}, \infty)\right], \tag{35}$$

where η_m, η_{σ_+} and η_{σ_-} represent the learning rate parameters of the PG membership function respectively.

4. Illustrative examples

In this section, we simulate some popular problems. The first example is to identify a dynamic system [6,7,9]. The second example is to predict time-series [1,3,4,6].

Example 1. Identification of the dynamic system

The plant to be identified is guided by the difference equation

$$y(k+1) = \frac{y(k)}{1 + y^2(k)} + u^3(k), \tag{36}$$

$$u(k) = \sin(2\pi k/100). \tag{37}$$

The output of the plant depends nonlinearly on both its pass output values and input values, but the effects of the input and output values are additive. Two hundred training patterns are generated by the plant Eqs. (36) and (37). The initial parameters are $\eta = 0.01$, $\sigma_i = 0.2$, $P_{\text{th}} = 0.15$, and $E_{\text{th}} = 0.6$. Starting at zero, the numbers of clusters grow dynamically for the incoming training data. The training process is continued

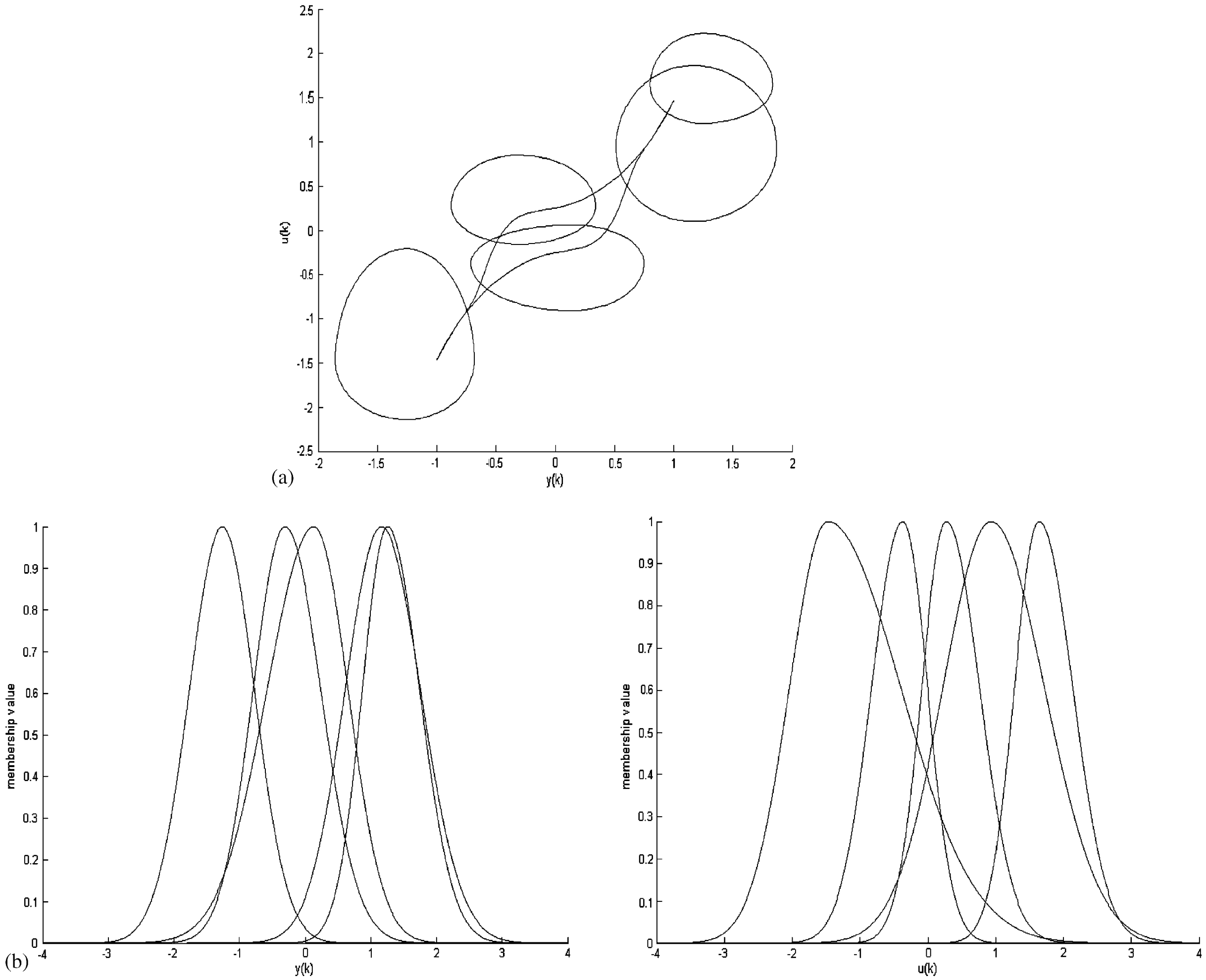


Fig. 6. Simulation results of the ASM-NFIS on the PG membership functions of each input variable in Example 1. (a) The input training patterns and the final assignment of rules. (b) The distribution of the membership functions on the $u(k)$ and $y(k)$ dimensions.

500 times. After training, the final root mean square (rms) error of the identification output approximates 0.001652. There are five fuzzy logic rules generated in ASM-NFIS. Fig. 6 illustrates the distribution of the training patterns and the final assignment of the rules (i.e., distribution of the membership functions) in $[u(k), y(k)]$ plain (input space). Fig. 7 shows the output of the plant and the identification model after the 500 training steps are finished. In this figure, the output of the ASM-NFIS model are represented by "$*$" while the plant output values are represented as "o". The results show the good identification capability of the trained ASM-NFIS model.

The selection of P_{th} will crucially affect the simulation results. A low P_{th} value leads to the learning of coarse clusters, whereas a high P_{th} value leads to the learning of fine clusters. In this simulation, we adopt the different P_{th} values to perform the identification problem. The simulation results are tabulated

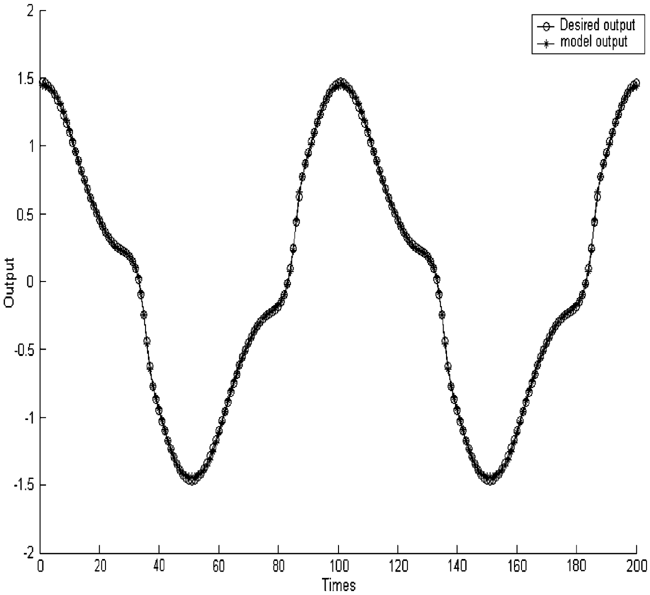


Fig. 7. Simulation results of the ASM-NFIS model in Example 1.

Table 1
Performance comparison of various P_{th} values on the identification problem

P_{th} values	0.05	0.15	0.2
Rule numbers	3	5	8
RMS error	0.0027	0.0016	0.00072

in Table 1. Table 1 shows if we choose a high P_{th} value, the numbers of fuzzy rules will be increased but the rms errors will be decreased.

Example 2. Prediction of the chaotic time-series

Let $P(k)$, $k = 1, 2, \ldots$, be a time series. The problem of the time-series prediction can be formulated in the following way: given $P(k-m+1),\ P(k-m+2), \ldots ,\ P(k)$, determine $P(k+l)$, where m and l are fixed positive integers. (i.e., determine a mapping from $[P(k-m+1),\ P(k-m+2), \ldots ,\ P(k)] \in \Re^m$ to $[P(k+l)] \in \Re$). To illustrate the online learning ability, the ASM-NFIS model is used to predict the Mackey–Glass chaotic time-series. The Mackey–Glass chaotic time-series is generated from the following delay differential equation:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t), \tag{38}$$

where $\tau > 17$.

In our simulation, we chose the series with $\tau = 30$. Fig. 8 shows 1000 points of this chaotic series used to test the ASM-NFIS model. We chose $m = 9$ and $l = 1$ (i.e., nine point values in the series are used to predict the value of the next time point). The 200 points of the series from $x(501) - x(700)$ are used as

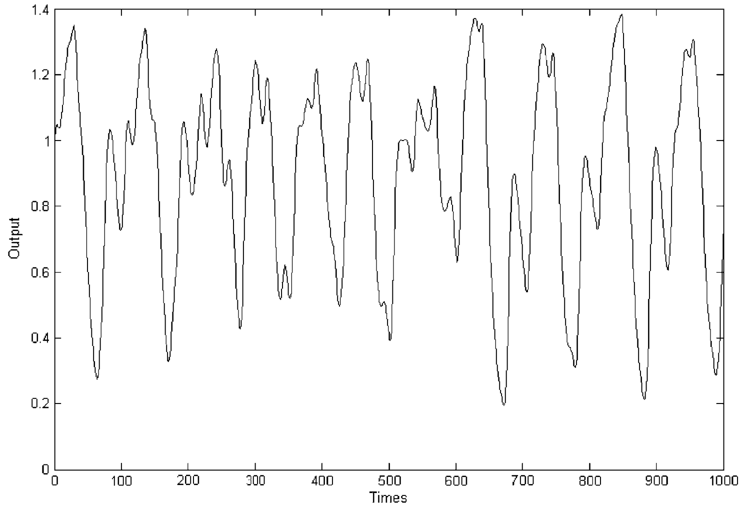


Fig. 8. The Mackey–Glass chaotic time series.

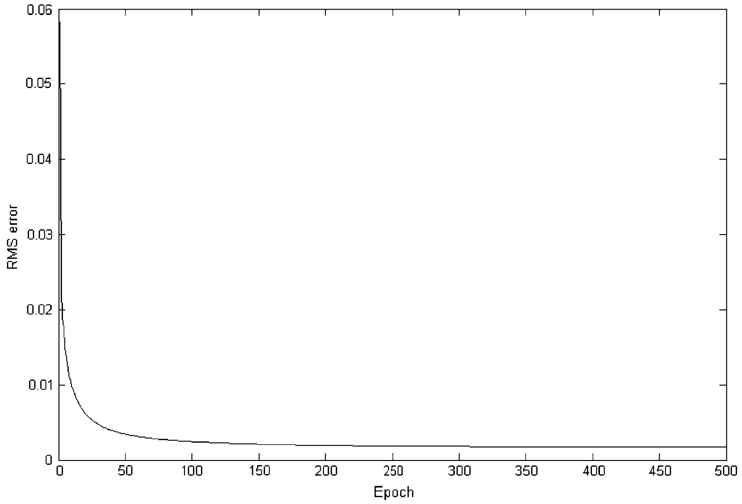


Fig. 9. Learning curve of the proposed ASM-NFIS model.

training data, and the final 300 points from $x(701) - x(1000)$ are used as test data. The initial parameters are $\eta = 0.01$, $\sigma_i = 0.2$, $P_{\text{th}} = 0.15$, and $E_{\text{th}} = 0.6$. After the structure and parameter learning, three fuzzy logic rules were generated in our model. The learning curve is shown in Fig. 9. The rms error of the prediction output approximates 0.0016083. Fig. 10 shows the prediction results of the trained ASM-NFIS. We compared the performance of our system with that of other existing methods that can generate fuzzy rules from numerical data automatically. The comparison results are tabulated in Table 2. The simulation results show that our system has better learning performance than other learning systems.

In [13], Wang and Mendel tried to improve the prediction accuracy by using an updating fuzzy rule base procedure and dividing the domain interval into finer regions in their system. In the end, their system

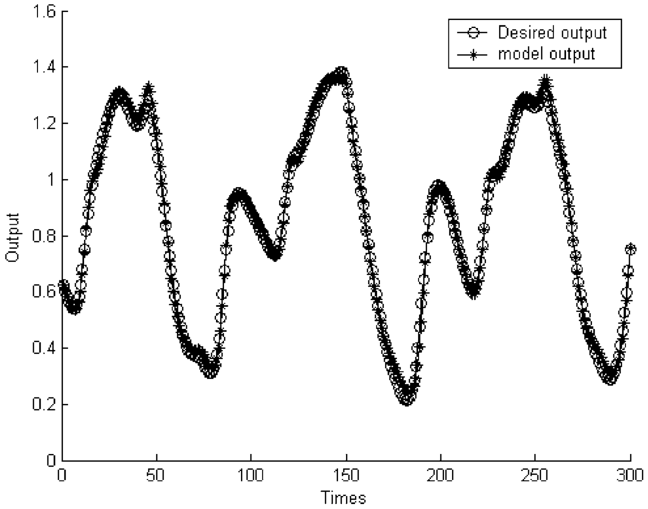


Fig. 10. The solid line denotes the output of ASM-NFIS and the dotted line denotes the true output.

Table 2
Performance comparison of various rule generation methods on the time-series prediction problem

	ASM-NFIS	FNN [8]	SONFIN [2]	FALCON-ART [6]	Wang and Mendel [13]	Kosko (AVQ) [5]	
						UCL	DCL
Rule numbers	3	4	4	22	121	100	100
RMS error	0.0016	0.0023	0.018	0.08	0.08	0.09	0.09

achieved perfect prediction capability when the domain interval was divided into 29 regions. Kim and Kasabov [4] proposed a hybrid neural fuzzy inference system (HyFIS) for building fuzzy models. In the HyFIS model, fuzzy rules were extracted by using fuzzy techniques proposed by [13] and parameters were adjusted by using a gradient descent learning algorithm. In [4] and [13], the input space needs to be divided properly in advance. Jang [1] proposed an adaptive-network-based fuzzy inference system (ANFIS) model for learning and tuning a fuzzy predictor. By using a hybrid learning procedure, the proposed ANFIS can construct an input/output mapping based on both human knowledge and stipulated input/output pairs. The ANFIS also has perfect prediction capability of the chaotic time series prediction problem. However, the proper fuzzy rules and space partition must be given in advance by experts.

5. Discussion

In this section, we summarize the features of the proposed ASM-NFIS model. First, distributed representation is used to represent the input patterns in the ASM-NFIS model. The input space is divided into overlapping smaller regions and the partitioning is not performed in advance, but is dynamically and appropriately adjusted during the learning process. As a result, each region varies in size, and the degree of overlap between regions is also adjustable. This is in contrast to [1,4,5,13] in which the input space needs to be divided properly in advance. The second feature of the proposed ASM-NFIS model prevents the newly generated membership function from being too similar to the existing membership function. The asymmetric similarity measure between the new membership function and the existing functions must be checked. This is in contrast to the symmetric similarity measure [2,4,8,10]. The third feature of the proposed ASM-NFIS model is that it flexibly partitions the input space. This is in contrast to the grid-type space partition [1]. In [1], as the number of input variables increases, the number of the partitioned grids will grow exponentially. As a result, the required size of memory or hardware may become impractically huge.

6. Conclusions

In this paper, we introduced a neural fuzzy system called ASM-NFIS. An on-line learning algorithm was proposed to automatically construct the ASM-NFIS. We addressed the automatic determination of the structure of the ASM-NFIS and the simultaneous optimization of both membership functions and fuzzy rule conclusions. The PG membership function was used to construct the general neural fuzzy system and to make the variability and the flexibility of the ASM-NFIS higher. The similarity measure of asymmetric Gaussian membership functions was proposed to estimate the rule's similarity degree. The proposed ASM-NFIS can obtain a smaller RMS error and generate less fuzzy logic rules.

Acknowledgement

This work was supported by the National Science Council, R.O.C., under grant NSC 92-2213-E-324-002.

References

[1] J.S.R. Jang, ANFIS: adaptive-network-based fuzzy inference systems, IEEE Trans. on Syst. Man, and Cybern. 23 (3) (1993) 665–685.

[2] C.-F. Juang, C.-T. Lin, An on-line self-constructing neural fuzzy inference network and its applications, IEEE Trans. on Fuzzy System 6 (1) (1998) 12–32.

[3] N. Kasabov, Q. Song, DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, IEEE Trans. on Fuzzy Systems 10 (2) (2002) 144–154.

[4] J. Kim, N. Kasabov, HyFIS: adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems, Neural Networks 12 (1999) 1301–1319.

[5] B. Kosko, Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, NJ, 1992.

[6] C.-J. Lin, C.-T. Lin, An ART-based fuzzy adaptive learning control network, IEEE Trans. Fuzzy System 5 (4) (1997) 477–496.

[7] C.-T. Lin, C.-J. Lin, C.-S.G. Lee, Fuzzy adaptive learning control network with on-line neural learning, Fuzzy Sets and Systems 71 (1995) 22–45.

[8] F.-J. Lin, C.-H. Lin, P.-H. Shen, Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive, IEEE Trans. Fuzzy System 9 (5) (2001) 751–759.

[9] K.-S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Trans. on Neural Networks 1 (1) (1990) 4–27.

[10] S. Paul, S. Kumar, Subsethood-product fuzzy neural inference system (SuPFuNIS), IEEE Trans. on Neural Networks 13 (3) (2002) 578–599.

[11] I. Rojas, H. Pomares, F.-J. Fernandez, J.-L. Bernier, F.-J. Pelayo, A. Prieto, A new methodology to obtain fuzzy systems autonomously from training data, IEEE Internat. Conf. on Fuzzy Systems 1 (1999) 527–532.

[12] I. Rojas, H. Pomares, F.-J. Fernandez, J.-L. Bernier, F.-J. Pelayo, A. Prieto, A new radial basis function networks structure: application to time series prediction, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, vol. 4, 2000, pp. 449–454.

[13] L.-X. Wang, J.-M. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. on Man, and Cybern. 22 (6) (1992) 1414–1427.