Article ID: 1000-8152(2001)03-0333-08

Theoretical Results on Applying EM-SFM Algorithm and Modular Network to Fuzzy Sugeno Model*

WANG Shitong

(Department of Computer, East China Shipbuilding Institute · Jiangsu Zhenjiang, 212003, P.R.China;
Advanced Computing Research Center, Bristol University · Bristol, U.K.)

J.F.Baldwin and T.Martin

(Advanced Computing Research Center, Bristol University · Bristol, U.K.)

Abstract: Based on modular network, the widely used Sugeno fuzzy model is newly re-explained. Accordingly, based on EM algorithm, a new algorithm EM-SFM for this model is presented, its linear convergence is proved, and its convergence rate is also analyzed.

Key words: Sugeno fuzzy model; EM algorithm; modular network; convergence

Document code: A

基于 Modular 网络和 EM-SFM 算法的模糊 Sugeno 模型

王士同

(华东船舶工业学院计算机系·江苏镇江,212003;布里斯托尔大学高级计算机研究中心·布里斯托尔,英国)

J.F.鲍尔蕴　T.马丁

(布里斯托尔大学高级计算机研究中心·布里斯托尔,英国)

摘要: 基于 Modular 网络重新解释了广为使用的模糊 Sugeno 模型.随后,基于 EM 算法,提出了该模型的新算法 EM-SFM,证明了该算法的线性收敛法,分析了它的收敛速度.

关键词: 模糊 Sugeno 模型; EM 算法; Modular 网络; 收敛

1 Introduction

Fuzzy Sugeno model is a well-known one, and it can express a highly nonlinear functional relation using a small number of fuzzy rules. Because conventional mathematical models may fail to give satisfactory results in describing the behavior of many complex nonlinear systems, the potential applications of fuzzy Sugeno model are wide. Many sucessful applications proved this viewpoint.

Prof. J. Buckly[1] proved fuzzy Sugeno model is a universal approximator. In practice, there exists a difficulty in building such a model. The parameter estimation of this model is a complex multiparameter optimization problem. For example, consider a model consisting of L rules with r input variables. There are $L^*(r+1)$ parameters to be estimated. Although some existing techniques such as the recursive least square method and the stochastic gradient search algorithm can be used to implement the parameter estimation, these techniques tend to be computationally complex and time-consuming when applied to the problem at hand.

Recently, Dr. L. Wang and R. Langri[2,3] presented a new method in which EM algorithm is applied to Sugeno model. However, they did not show whether and how the presented algorithm can theoretically converge.

In order to overcome this difficulty, we present a novel approach. Firstly, we explain fuzzy Sugeno model as a special modular network. Accordingly, we apply new EM algorithm to the special modular network. We will give important linear convergence results for new EM algorithm to fuzzy Sugeno model.

* Foundation item: supported by British Royal Society and National Natural Science Foundation of China (6983004).
Received date: 2000-08-28; Revised date: 2001-01-10.

The modular network[4] is a connectionist architecture that learns to partition a global task into several simpler subtasks and allocates distinct networks to learn each subtask. The modular network offers several advantages over a single neural network in terms of learning speed, generalization capabilities, representation capabilities and their ability to satisfy contraints imposed by hardware limitations.

In this paper, we will point out that modular network has an inherent similar relation with fuzzy Sugeno model, that is, Sugeno model can be viewed as a special modular network. In this manner, we can re-explain fuzzy Sugeno model by using modular netowrk. Based on this assertion, we present a new EM algorithm, which is used to optimize the parameters of fuzzy Sugeno model. Further theoretical analysis shows this new algorithm can linearly converge. This important result will make the new EM algorithm become a practical tool, which is used to build Sugeno model.

2 Re-explanation of fuzzy Sugeno model based on modular network

The Sugeno model consists of the following fuzzy IF-THEN rules:

R_l: IF x_1 is A_{1l} and $\cdots$ and x_p is A_{pl},

THEN $y_l = b_{0l} + b_{1l}x_1 + \cdots + b_{pl}x_p$,

where $b_{il}(i = 0,1,\cdots,p)$ are real-valued parameters, y_l is the local output of the model due to rule R_l and $l = 1,2,\cdots,K$.

We should note that the most important characteristic of the above rule is that the consequent of each rule is described by a linear regression model.

The total output of Sugeno model is a crisp value, defined by the weighted average:

$$y = \sum_{l=1}^{K} h_l y_l / \sum_{l=1}^{K} h_l, \tag{1}$$

where

$$h_l = \mu_{A_{1l}}(x_1)\mu_{A_{2l}}(x_2)\cdots\mu_{A_{pl}}(x_p),$$

$$\mu_{A_{il}}(x_i) = \exp(-(x_i - m_{il})^2/(2\sigma_{il})),$$

$$\boldsymbol{m}_l = (m_{1l}, m_{2l}, \cdots, m_{pl})^{\mathrm{T}},$$

$$\boldsymbol{\sigma}_l = (\sigma_{1l}, \sigma_{2l}, \cdots, \sigma_{pl})^{\mathrm{T}}.$$

Fig. 1 shows the basic configuration of a modular network. A modular network often consists of several expert netwroks and a gating network.

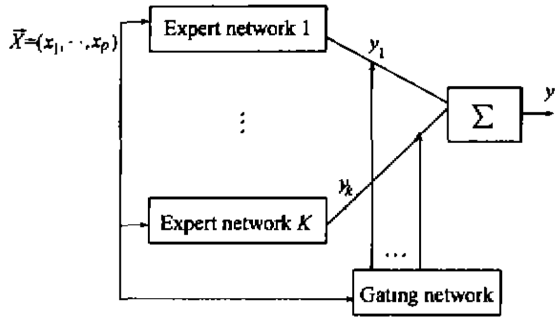


Fig. 1 Basic configuration of a modular network

Suppose each expert network of modular network represents a conclusion part of fuzzy rule of Sugeno model, i.e., for lth expert network

$$y_l = b_{0l} + b_{1l}x_l + \cdots + b_{pl}x_p,$$

it can be illustrated by Fig. 2.

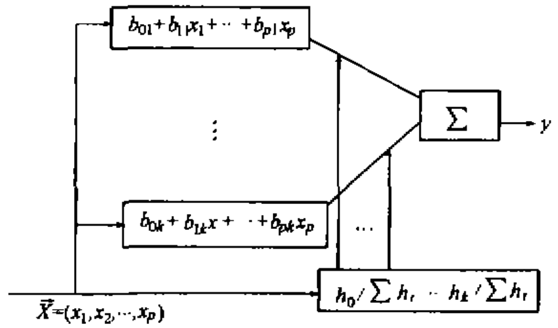


Fig. 2 Fuzzy sugeno model

For such expert network, we associate probabilistic model that relates input vector $x \in \mathbb{R}^p$ to output $y \in \mathbb{R}$. We denote these probabilistic models as follows:

$$P(x \mid \boldsymbol{x}, \boldsymbol{b}_l),$$

where $\boldsymbol{b}_l = (\boldsymbol{b}_{0l}, \boldsymbol{b}_{1l}, \cdots, \boldsymbol{b}_{pl})^{\mathrm{T}}$. As Dr. Jordan et al point out, each of these probability densities can be generally assumed to belong to the expoential family of densities. In this paper, we restrict out analysis to Gaussian model for simplicity.

For Gaussian $P(y \mid \boldsymbol{x}, \boldsymbol{b}_l)$, y_l is simply the mean. We also associate a covariance δ_l with lth expert network, yielding the following probabilistic model for expert l.

$$P(y \mid \boldsymbol{x}, \boldsymbol{b}_l, \delta_l) = (2\pi\delta_l)^{-1/2}\exp\{-(y - y_l)^2/(2\delta_l)\}. \tag{1'}$$

For the gating network of the modular network, in terms of formula (1), we define

$$g_l(\boldsymbol{x}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) = \frac{h_l}{\sum_{l=1}^{K} h_l} =$$

$$\frac{\left[\prod_{i=1}^{p} \exp\{-(x_i - m_{il})^2/(2\sigma_l^2)\}\right]}{\sum_{l=1}^{K}\left[\prod_{i=1}^{p} \exp\{-(x_i - m_{il})^2/(2\sigma_l^2)\}\right]},$$

where $l = 1,2,\cdots,K$.

Note: this definition means that $\sum_{l=1}^{K} g_l = 1$.

We suppose that training data is generated according to the following probability model. We assume that for a given $\boldsymbol{x}$, a label l is selected with probability $P(l \mid \boldsymbol{x}) = g_l(\boldsymbol{x}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l)$. Thus, the total probability of observing y from $\boldsymbol{x}$ is given by the following finite mixture density

$$P(y \mid \boldsymbol{x}) = \sum_{l=1}^{K} P(l \mid \boldsymbol{x}) P(y \mid \boldsymbol{x}, \boldsymbol{b}_l, \delta_j) = \sum_{l=1}^{K} g_l(\boldsymbol{x}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) P(y \mid \boldsymbol{x}, \boldsymbol{b}_l, \delta_j). \quad (2)$$

A training set $\Psi = \{(\boldsymbol{x}^{(t)}, y^{(t)}), t = 1,2,\cdots,N\}$ is assumed to be generated as an independant set based on the above mixture density. Thus, the total probability of the training set, for a special set of input vectors $\{x^{(t)}|_{t=1}^{N}$, is given by the following likelihood function:

$$L = P(\{y^{(t)}\}_{t=1}^{N} \mid (\{\boldsymbol{x}^{(t)}|_{t=1}^{N}) = \prod_{t=1}^{N} P(\{y^{(t)}\} \mid \{\boldsymbol{x}^{(t)}\}) = \quad (3)$$

$$\prod_{t=1}^{N} \sum_{j=1}^{K} g_j(\boldsymbol{x}^{(t)}, \boldsymbol{m}_j, \boldsymbol{\sigma}_j) P(y^{(t)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_j, \delta_j). \quad (4)$$

We design the learning algorithm as the maximum likelihood estimator. In other words, we treat learning as the problem of finding parameters $\boldsymbol{b}_l, \boldsymbol{m}_l, \boldsymbol{\sigma}_l$ to maximize L, or, more conventionally, to maximize the log likelihood $l_c = \ln L$.

$$l_c = \ln L = \sum_{t=1}^{N} \ln \sum_{j=1}^{K} g_j(\boldsymbol{x}^{(t)}, \boldsymbol{m}_j, \boldsymbol{\sigma}_j) P(y^{(t)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_j, \delta_j).$$

Given the probability model in formula (3), the expected value of the output is given as follows:

$$y = \sum [y \mid \boldsymbol{x}] = \sum_{l=1}^{K} g_l(\boldsymbol{x}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) y_l. \quad (5)$$

We should note that formula (4) is just the same as formula (1). This sufficiently shows fuzzy Sugeno model is a special case of modular networks. In other words, we re-explain the fuzzy Sugeno model, based on modular network.

We should also note: The special modular network here for fuzzy Sugeno model is quite different from the modular network model given by Dr. M. J. Jordan and Dr. L. Xu[5]. By comparing these two models, we will find that $P(y \mid \boldsymbol{x}, \boldsymbol{b}_l)$ here is a special case of $P(y \mid \boldsymbol{x}, \boldsymbol{\theta})$[5], where $g_l(\boldsymbol{x}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l)$ is a non-linear extension of $g_j(\boldsymbol{x}, \boldsymbol{\theta}_0)$[5] in which $s_j(\boldsymbol{x}, \boldsymbol{\theta}_0)$ is a linear function.

3 Algorithm EM-SFM for training fuzzy Sugeno model

Based on the modular network in the above section, this section will give a new EM algorithm for the above special modular network, i.e., fuzzy Sugeno model. We first introduce the basic idea of EM algorithms.

EM algortihm is a general method to iterative computation of maximum likelihood estimates given incomplete data. It originates from statics, but also finds successful applications in many other fields such as mixture density estimation, signal processing, neural network. The term incomplete data has two applications: 1) The existence of two sample spaces Y and X represented by the observed data y and the complete data x, respectively; 2) The one-to-many mapping $x \rightarrow y(x)$ for space X to space Y. The complete data x can not be observed directly, but only through the incomplete data y. The use of EM algorithm can lead to a significant reduction in computational complexity for estimating the parameters of the model.

Each iteration of the EM algorithm is composed of two steps: an expectation (E) step and a maximization (M) step - hence the name of the algorithm.

E step: Compute the expected value of the complete data log likelihood l_c, given the observed data y and the current model represented by the parameter vector $\theta^{(p)}$:

$$Q(\theta, \theta^{(p)}) = \mathrm{E}[l_c(y \mid \theta)] = \mathrm{E}[\ln(y \mid \theta)] = \mathrm{E}[\ln(\int P(y, y_{\mathrm{mis}} \mid \theta) \mathrm{d}y_{\mathrm{mis}})], \quad (6)$$

where y_{mis} represents missing or hidden variables, $\theta^{(p)}$ is the value of the parameter vector at iteration p.

This step yields a deterministic function Q.

M step: Maximize the deterministic function Q with respect to θ to find the new parameter estimates $\theta^{(p+1)}$:

$$\theta^{(p+1)} = \arg\max_{\theta} Q[\theta, \theta^{(p)}]. \tag{7}$$

For fuzzy Sugeno model, i.e., the special modular network, we choose the missing data to be a set of indicator random variables $y_{\text{mis}} = \{I_j^{(t)}, j = 1, \cdots, K, t = 1, \cdots, N\}$ with

$$I_j^{(t)} = \begin{cases} 1, & \text{if } y^{(t)} \text{ is generated from the } j\text{th model by formula } (1'); \\ 0, & \text{otherwise} \end{cases}$$

and $$\sum_{j=1}^{K} I_j^{(t)} = 1, \text{ for each } t.$$

We assume that the distribution of the complete data is given as

$$P(y, y_{\text{mis}} \mid \theta) = \prod_{t=1}^{N} \prod_{j=1}^{K} [g_j(\boldsymbol{x}^{(t)}, \boldsymbol{m}_j, \boldsymbol{\sigma}_j P(y \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_j, \delta_j]^{I(t)j}.$$

It is easy to verify that this distribution satisfies formula (6). Thus, in terms of formula (6),

$$Q(\theta \mid \theta^k) = E[\ln P(y, y_{\text{mis}} \mid \theta)] = \sum_{t=1}^{N} \sum_{l=1}^{K} h_l^{(k)}(t) \ln[g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) P(y \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_l, \delta_l)] = \sum_{t=1}^{N} \sum_{l=1}^{K} h_l^{(k)}(t) \ln g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) + \sum_{t=1}^{N} h_1^{(k)}(t) \ln P(y^{(t)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_1, \delta_1) + \cdots + \sum_{t=1}^{N} h_K^{(k)}(t) \ln P(y^{(t)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_K, \delta_K), \tag{8}$$

$$h_l^{(k)}(t) = p(l \mid \boldsymbol{x}^{(t)}, y^{(t)}) = \frac{[g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l^{(k)}, \boldsymbol{\sigma}_l^{(k)}) P(y \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_l^{(k)}, \delta_l^{(k)})]}{\sum_{i=1}^{K} [g_i(\boldsymbol{x}^{(t)}, \boldsymbol{m}_i^{(k)}, \boldsymbol{\sigma}_i^{(k)}) P(y \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_i^{(k)}, \delta_i^{(k)})]}, \tag{9}$$

where $P(l \mid \boldsymbol{x}^{(t)}, y^{(t)})$ denotes the probability that the pair $\{\boldsymbol{x}^{(t)}, y^{(t)}\}$ comes from the lth probability model. Note: $h_l^{(k)}(t) > 0$ for any time.

With function Q, we now give *M* step of EM algorithm. In terms of formula (8) and (1'), we have

$$\partial Q / \partial \boldsymbol{m}_l = \sum_{t=1}^{N} h_l^{(k)}(t) [\partial g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) / \partial \boldsymbol{m}_l] / g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) = \sum_{t=1}^{N} h_l^{(k)}(t) [1 - g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l)] [(\boldsymbol{x}^{(t)} - \boldsymbol{m}_l) / \boldsymbol{\sigma}_l^{\mathrm{T}} \boldsymbol{\sigma}_l], \tag{10}$$

$$\partial Q / \partial \boldsymbol{\sigma}_l = \sum_{t=1}^{N} h_l^{(k)}(t) [1 - g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l)] [(\boldsymbol{x}^{(t)} - \boldsymbol{m}_l)^{\mathrm{T}} (\boldsymbol{x}^{(t)} - \boldsymbol{m}_l) / (\boldsymbol{\sigma}_l^{\mathrm{T}} \boldsymbol{\sigma}_l)^2], \tag{11}$$

Similarly, we also have

$$\partial Q / \partial \boldsymbol{b}_l = \sum_{t=1}^{N} h_l^{(k)}(t) (y - y_l) \boldsymbol{x}'^{(t)}, \tag{12}$$

where $\boldsymbol{x}'^{(t)} = (I, x_1, \cdots, x_p)$.

$$\partial Q / \partial \delta_l = \sum_{t=1}^{N} h_l^{(k)}(t) [(y - y_l)^2 / (2\delta_l \delta_l) - I / (2\delta_l)]. \tag{13}$$

Let $\partial Q / \partial \boldsymbol{b}_l = 0$, where $\boldsymbol{b}_l = \boldsymbol{b}_l^{(k+1)}$, i.e.

$$\sum_{t=1}^{N} h_l^{(k)}(t) (y - y_l) \boldsymbol{x}'^{(t)} = 0,$$

i.e.

$$\sum_{t=1}^{N} h_l^{(k)}(t) y \boldsymbol{x}'^{(t)} = \sum_{t=1}^{N} h_l^{(k)}(t) \boldsymbol{b}_l (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)}.$$

Thus, we obtain the update for $\boldsymbol{b}_l$

$$\boldsymbol{b}_l^{(k+1)} = \sum_{t=1}^{N} h_l^{(k)}(t) y \boldsymbol{x}'^{(t)} / \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)}. \tag{14}$$

Let $\partial Q / \partial \delta_l = 0$, where $\delta_l = \delta_l^{(k+1)}$, we have

$$\delta_l^{(k+1)} = \sum_{t=1}^{N} h_l^{(k)}(t) (y - y_l)^2 / \sum_{t=1}^{N} h_l^{(k)}(t). \tag{15}$$

Let $\partial Q / \partial \boldsymbol{m}_l = 0$, where $\boldsymbol{m}_l = \boldsymbol{m}_l^{(k+1)}$, we have

$$\boldsymbol{m}_l^{(k+1)} = \frac{\sum_{t=1}^{N} h_l^{(k)}(t) [1 - g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l)] \boldsymbol{x}^{(t)}}{\sum_{t=1}^{N} h_l^{(k)}(t) [1 - g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l)]}. \tag{16}$$

From formula (11), we know, $\partial Q / \partial \boldsymbol{\sigma}_l > 0$, therefore, we can not use the above method to get $\boldsymbol{\sigma}_l^{(k+1)}$. However, $\partial Q / \partial \boldsymbol{\sigma}_l > 0$ shows that $\boldsymbol{\sigma}_l$ is monotonically increasing. Hence, there must exist some positive define matrix $U^{(k)}$ such that

$$\boldsymbol{\sigma}_l^{(k+1)} = \boldsymbol{\sigma}_l^{(k)} + U^{(k)} (\partial Q / \partial \boldsymbol{\sigma}_l). \tag{17}$$

In practice, $U^{(k)}$ is defined by user. Generally speaking, it usually takes a value which approximates 0.

In summary, the parameter update algorithm EM-SFM (EM algorithm for Sugeno fuzzy model) for the model formula (2), i.e., Sugeno fuzzy model, is given as follows:

Algorithm EM-SFM

1) *E* step: Compute the $h_l^{(k)}(t)$ by formula (9).

2) M step: Compute $\boldsymbol{b}_l^{(k+1)}$ by formula (14); compute $\delta_l^{(k+1)}$ by formula (15); compute $\boldsymbol{m}_l^{(k+1)}$ by formula (16); compute $\boldsymbol{\sigma}_l^{(k+1)}$ by formula (17).

4 Theoretical linearly convergence proof

In this section, we will theoretically show that EM-SFM algorithm can converge linearly. In the meantime, we will investigate the convergence rate of this algorithm. We begin with a convergence theorem which estabilshes a relationship between EM-SFM algorithm and gradient ascent.

Theorem 1 For algorithm EM-SFM, the following formulas hold:

$\boldsymbol{b}_l^{(k+1)} - \boldsymbol{b}_l^{(k)} = R^{(k)}(\partial \boldsymbol{l}_c / \partial \boldsymbol{b}_l)$, where $\boldsymbol{b}_l = \boldsymbol{b}_l^{(k)}$,

$\delta_l^{(k+1)} - \delta_l^{(k)} = S^{(k)}(\partial \boldsymbol{l}_c / \partial \delta_l)$, where $\delta_l = \delta_l^{(kH)}$,

$\boldsymbol{m}_l^{(k+1)} - \boldsymbol{m}_l^{(k)} = T^{(k)}(\partial \boldsymbol{l}_c / \partial \boldsymbol{m}_l)$, where $\boldsymbol{m}_l = \boldsymbol{m}_l^{(k)}$,

$\boldsymbol{\sigma}_l^{(k+1)} - \boldsymbol{\sigma}_l^{(k)} = U^{(k)}(\partial \boldsymbol{l}_c / \partial \boldsymbol{\sigma}_l)$, where $\boldsymbol{\sigma}_l = \boldsymbol{\sigma}_l^{(kH)}$,

where $R^{(k)}, S^{(k)}, T^{(k)}, U^{(k)} > 0$, i.e. positive definite matrix.

Proof We know,

$$l_c = \ln L = \sum_{t=1}^{N} \ln \sum_{j=1}^{K} g_j(\boldsymbol{x}^{(t)}, \boldsymbol{m}_j, \boldsymbol{\sigma}_j) P(y^{(t)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_j, \sigma_j),$$

therefore, we have

$$\partial l_c / \partial \boldsymbol{m}_l = \sum_{t=1}^{N} \{ [P(y^{(t)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_l, \delta_l)] / \sum_{j=1}^{K} g_j(\boldsymbol{x}^{(t)}, \boldsymbol{m}_j, \boldsymbol{\sigma}_j) P(y^{(t)} \mid \boldsymbol{x}^{(t)}, \boldsymbol{b}_j, \delta_j)] \times (\partial g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) / \partial \boldsymbol{m}_l) \} = \sum_{t=1}^{N} h_l^{(k)}(t) [\partial g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) / \partial \boldsymbol{m}_l] / g_l(\boldsymbol{x}^{(t)}, \boldsymbol{m}_l, \boldsymbol{\sigma}_l) = \partial Q / \partial \boldsymbol{m}_l.$$

Similarly, we also have

$$\partial \boldsymbol{l}_c / \partial \boldsymbol{b}_l = \partial Q / \partial \boldsymbol{b}_l,$$
$$\partial \boldsymbol{l}_c / \partial \delta_l = \partial Q / \partial \delta_l,$$
$$\partial \boldsymbol{l}_c / \partial \sigma_l = \partial Q / \partial \boldsymbol{\sigma}_l.$$

In terms of formula (14), we have

$$\boldsymbol{b}_l^{(k+1)} = \sum_{t=1}^{N} h_l^{(k)}(t) y \boldsymbol{x}'^{(t)} / \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)} = \boldsymbol{b}_l^{(k)} + \sum_{t=1}^{N} h_l^{(k)}(t) y \boldsymbol{x}'^{(t)} - \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)} \boldsymbol{b}_l^{(k)} =$$
$$\boldsymbol{b}_l^{(k)} + [\sum_{t=1}^{N} h_l^{(k)}(t) y \boldsymbol{x}'^{(t)} / \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)} - \boldsymbol{b}_l^{(k)}] / \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)} =$$
$$\boldsymbol{b}_l^{(k)} + [\sum_{t=1}^{N} h_l^{(k)}(t) (y - (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{b}_l^{(k)} \boldsymbol{x}'^{(t)}] / \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)}.$$

In terms of Sugeno model, we know

$$y_l^{(k)} = (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{b}_l^{(k)}.$$

Thus, the above formula becomes

$$\boldsymbol{b}_l^{(k+1)} = \boldsymbol{b}_l^{(k)} + [\sum_{t=1}^{N} h_l^{(k)}(t)(y - y_l^{(k)}) \boldsymbol{x}'^{(t)}] / \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)}.$$

In terms of formula (12), we furtherly have

$$\boldsymbol{b}_l^{(k+1)} = \boldsymbol{b}_l^{(k)} + (\partial Q / \partial \boldsymbol{b}_l) / \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)} = \boldsymbol{b}_l^{(k)} + (\partial \boldsymbol{l}_c / \partial \boldsymbol{b}_l) / \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)},$$

where $\boldsymbol{b}_l = \boldsymbol{b}_l^{(k)}$.

Corresponding to the state of this theorem, we take

$$R^{(k)} = 1 / \sum_{t=1}^{N} h_l^{(k)}(t) (\boldsymbol{x}'^{(t)})^{\mathrm{T}} \boldsymbol{x}'^{(t)},$$

Because $h_l^{(k)}(t) > 0$, we must have $R^{(k)} > 0$.

Similarly, we have

$$\delta_l^{(k+1)} = \sum_{t=1}^{N} h_l^{(k)}(t)(y - y_l)^2 / \sum_{t=1}^{N} h_l^{(k)}(t) =$$
$$\delta_l^{(k)} + \sum_{t=1}^{N} h_l^{(k)}(t)(y - y_l)^2 / \sum_{t=1}^{N} h_l^{(k)}(t) - \delta_l^{(k)} =$$
$$\delta_l^{(k)} + [\sum_{t=1}^{N} h_l^{(k)}(t)(y - y_l)^2 - \sum_{t=1}^{N} h_l^{(k)}(t) \delta_l^{(k)}] / \sum_{t=1}^{N} h_l^{(k)}(t) =$$
$$\delta_l^{(k)} + [\sum_{t=1}^{N} h_l^{(k)}(t) \{(y - y_l)^2 / (2\delta_l^{(k)} \delta_l^{(k)}) - 1/(2\delta_l^{(k)})\}] 2\delta_l^{(k)} \delta_l^{(k)} / \sum_{t=1}^{N} h_l^{(k)}(t) =$$
$$\delta_l^{(k)} + \{2\delta_l^{(k)} \delta_l^{(k)} / \sum_{t=1}^{N} h_l^{(k)}(t)\} \partial Q / \partial \delta_l =$$
$$\delta_l^{(k)} + \{2\delta_l^{(k)} \delta_l^{(k)} / \sum_{t=1}^{N} h_l^{(k)}(t)\} \partial l_c / \partial \delta_l,$$

where $\delta_l = \delta_l^{(k)}$.

Let $S^{(k)} = \{2\delta_l^{(k)}\delta_l^{(k)}/\sum_{t=1}^{N} h_l^{(k)}(t)\}$. Obviously, $S^{(k)} > 0$.

We also have

$$m_l^{(k+1)} = \sum_{t=1}^{N} h_l^{(k)}(t)[1 - g_l(x^{(t)}, m_l, \sigma_l)]x^{(t)} / \sum_{t=1}^{N} h_l^{(k)}(t)[1 - g_l(x^{(t)}, m_l, \sigma_l)] =$$
$$m_l^{(k)} + \sum_{t=1}^{N} h_l^{(k)}(t)[1 - g_l(x^{(t)}, m_l, \sigma_l)]x^{(t)} / \sum_{t=1}^{N} h_l^{(k)}(t)[1 - g_l(x^{(t)}, m_l, \sigma_l)] - m_l^{(k)} =$$
$$m_l^{(k)} + \{\sigma_l^{\mathrm{T}}\sigma_l / \sum_{t=1}^{N} h_l^{(k)}(t)[1 - g_l(x^{(t)}, m_l, \sigma_l)]\}\partial Q/\partial m_l =$$
$$m_l^{(k)} + \{\sigma_l^{\mathrm{T}}\sigma_l / \sum_{t=1}^{N} h_l^{(k)}(t)[1 - g_l(x^{(t)}, m_l, \sigma_l)]\}\partial l_c/\partial m_l,$$

where $m = m_l^{(k)}$.

Let

$$T^{(k)} = \sigma_l^{\mathrm{T}}\sigma_l / \sum_{t=1}^{N} h_l^{(k)}(t)[1 - g_l(x^{(t)}, m_l, \sigma_l)].$$

In terms of the definition of $g_l(x^{(t)}, m_l, \sigma_l)$, we know $g_l(x^{(t)}, m_l, \sigma_l) < 1$, therefore, $T^{(k)} > 0$.

For the case of σ_l, this theorem obviously holds. This completes the proof of this theorem.

Theorem 1 is very important, and it implies the following Theorem 2.

Theorem 2 For Sugeno fuzzy model and algorithm EM-SFM, the search direction of this algorithm has a positive projection on the gradient ascent search direction of $l_c = \ln L$.

This theorem shows EM-SFM algorithm can be viewed as a modified gradient ascent algorithm for maximizing. $l_c = \ln L$. This algorithm searches in an uphill direction, so if the learning rate is appropriate, the search process will converge to a local maximum or a saddle point of Ln L.

Now, we investigate the convergence problem of EM-SFM algorithm.

Theorem 3 Suppose the training set is generated by Sugeno fuzzy model shown in this paper, and the number N of elements in the training set is sufficiently large. Assume

$$A = (b_l^{\mathrm{T}}, \delta_l^{\mathrm{T}}, m_l^{\mathrm{T}}, \sigma_l^{\mathrm{T}})^{\mathrm{T}},$$
$$B = \mathrm{diag}(R^{(k)}, S^{(k)}, T^{(k)}, U^{(k)}),$$
$$H(A) = \partial^2 l_c(A)/\partial A \partial A^{\mathrm{T}}.$$

Assume that on a given domain D_A,

1) Hessian matrix $H(A)$ is negative definite;

2) A^* is a local maximum of $l_c(A)$ and $A^* \in D_A$.

Then,

1° Letting $-M$, $-m (M > m > 0)$ be the minimum and maximum eigenvalues of the negative definite matrix $(B^{1/2})^{\mathrm{T}}H(A)B^{1/2}$, we have

$$l_c(A^*) - l_c(A^{(k)}) \leqslant r^k[l_c(A^*) - l_c(A_0)], \quad (18)$$
$$\| B^{-1/2}(A^{(k)} - A^* \| \leqslant | r |^{k/2}[2(l_c(A^*) - l_c(A_0))/m]^{1/2}, \quad (19)$$

where $r = 1 - (1 - M/2)m^2/M < 1$.

2° For any initial point $A_0 \in D_A$, $\lim_{k\to\infty} A^{(k)} = A^*$ when $M < 2$.

Proof According to Sugeno fuzzy model, we can easily verify that $H(A)$ exists and remains continous. We now expand the log likelihood in a Taylor expansion.

$$l_c(A) - l_c(A^*) = (A - A^*)(\partial l_c/\partial A)|_{A=A^*} + (A - A^*)^{\mathrm{T}}H(A^* + \xi(A - A^*))(A - A^*))/2$$

with $0 < \xi < 1$. Because

$$\partial l_c/\partial A|_{A=A^*} = 0.$$

We furtherly have

$$l_c(A) - l_c(A^*) = (A - A^*)^{\mathrm{T}}H(A^* + \xi(A - A^*))(A - A^*)/2. \quad (20)$$

From Theorem 1, we know that B is positive definite. Because $H(A)$ is negative definite. This implies that $B^{1/2}$ exists and $(B^{1/2})^{\mathrm{T}}H(A)B^{1/2}$ is negative definite on D_A. By using the Rayleigh quotient, for any u we have

$$-M\|u\|^2 \leqslant u^{\mathrm{T}}(B^{1/2})^{\mathrm{T}}H(A)B^{1/2}u \leqslant -m\|u\|^2. \quad (21)$$

Substituting formula (21) into formula (20), we have

$$l_c(A) - l_c(A^*) = (A - A^*)^{\mathrm{T}}(B^{-1/2})^{\mathrm{T}}(B^{1/2})^{\mathrm{T}}H(A^* + \xi(A - A^*))B^{1/2}B^{-1/2}(A - A^*)/2, \quad (22)$$

$$l_c(A) - l_c(A^*) \geqslant - M \| (B^{-1/2})(A - A^*) \|^2/2. \tag{23}$$

Moreover, we have

$$- m \| (B^{-1/2})(A - A^*) \|^2 \geqslant$$
$$| (A - A^*)^T[\partial l_c/\partial A - \partial l_c/\partial A |_{A=A^*}] | \geqslant$$
$$\| B^{1/2}\partial l_c/\partial A \| \ \| (B^{-1/2})(A - A^*) \| .$$

Thus,

$$\| (B^{-1/2})(A - A^*) \| \leqslant \| B^{1/2}\partial l_c/\partial A \| / m.$$

Together with formula (23), we also have

$$- \| B^{1/2}\partial l_c/\partial A \| \leqslant 2m^2[l_c(A) - l_c(A^*)]/M. \tag{24}$$

On the other hand, we have

$$l_c(A) - l_c(A^{(k)}) =$$
$$(A - A^{(k)})(\partial l_c/\partial A) |_{A=A^*} +$$
$$(A - A^{(k)})^T H(A^{(k)} + \xi'(A - A^{(k)}))(A - A^{(k)})/2,$$

with $0 \leqslant \xi' < 1$. In terms of Theorem 1, for algorithm EM-SFM, we have

$$A^{(k+1)} - A^{(k)} = B(\partial l_c/\partial A) |_{A=A^*}.$$

In terms of the above two formulas, we can derive

$$l_c(A^{(k+1)}) - l_c(A^{(k)}) =$$
$$\| B^{1/2}\partial l_c/\partial A \|^2_{A=A^*} +$$
$$(B^{1/2}\partial l_c/\partial A |_{A=A^*})^T \times (B^{1/2})^T H(A^{(k)} +$$
$$\xi(A - A^{(k)}))B^{1/2}(B^{1/2}\partial l_c/\partial A |_{A=A^*})/2 \geqslant$$
$$(1 - M/2) \| B^{1/2}\partial l_c/\partial A \|^2_{A=A^*}. \tag{25}$$

Combining formula (25) and (24), we have

$$l_c(A^{(k+1)}) - l_c(A^{(k)}) \geqslant$$
$$- (- 1 - M/2)2m^2[l_c(A^{(k)} - l_c(A^*)]/M \geqslant$$
$$- \{(1 - M/2)2m^2/M\}^k[l_c(A_0) - l_c(A^*)]. \tag{26}$$

Let $r = - (1 - M/2)2m^2/M$. Thus, the above formula become formula (18) by multiplying both sides by -1. Furthermore, it is easy to verify that $0 < | r | < 1$ when $M < 2$.

In terms of formula (20) and (21),

$$l_c(A^{(k)}) - l_c(A^*) \leqslant -m \| (B^{-1/2})(A^{(k)} - A^*) \|^2/2,$$

furtherly, by formula (26), we have

$$-m \| (B^{-1/2})(A^{(k)} - A^*) \|^2/2 \geqslant r^k[l_c(A_0) - l_c(A^*)],$$

i.e.

$$\| B^{-1/2}(A^{(k)} - A^*) \| \leqslant | r |^{k/2}[2(l_c(A^*) - l_c(A_0))/m]^{1/2}.$$

In addition, because B is positive defined, when $M < 2$ and $-1 < r < 1$, we have $\lim_{k\to\infty} A^{(k)} = A^*$. This theorem is completely proved.

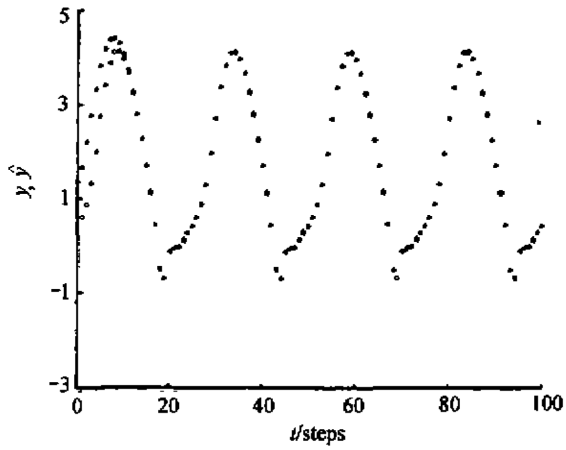
This theorem is very important, and shows that algorithm EM-SFM for Sugeno fuzzy model can converge linearly and that the convergence rate depends on the difference between M and m. The smaller the difference, the faster the convergence. Therefore, algorithm EM-SFM is a well-defined one for practical applications of Sugeno fuzzy model.

Example 1 We use fuzzy Sugeno model with EM-SFM to identify the following nonlinear system:

$$y(t+1) = \frac{y(t)y(t-1)(y(t)+2.5)}{1 + y^2(t) + y^2(t-1)} + u(t),$$

where activation function $u(t) = \sin\frac{2\pi t}{25}$.

We divide the fuzzy subsets of $u(t), y(t), y(t-1)$ into {NS, ZO, PB}, where NS, ZO, PB represent Negative Small, Zero, Positive Big, respectively. Gaussian membership functions are used, as shown in (1). With EM-SFM, after 130 iterations the mean variance of all samples is 0.000279. Fig. 3 shows the curves of the actual system and the fuzzy system at different initial states, respectively.



The hollow dots — the actual system
Initial state values are $y(-1)=1.5, y(0)=0.8$
The solid dots — the fuzzy system
Initial state values are $\hat{y}(-1)=1.5, \hat{y}(0)=0.0$

Fig. 3 Curves of actual system and the fuzzy system

5 Conclusion

Sugeno fuzzy model has been increasingly popular and is being widely used and sucessfully. In this paper, we have contributed to the theory of combining EM algorithm, modular network and Sugeno fuzzy model. We have presented the new effective EM-SFM algorithm of this model and given the proof of the linear convergence of this algorithm.

References

[1] Buckley J J. Sugeno-type controllers are universal controllers [J]. Fuzzy Sets and Systems, 1993,53(6):299 - 303

[2] Langari R and Wang Liang. Fuzzy models, modular networks, and hybrid learning [J]. Fuzzy Sets and Systems, 1996,79(2):141 - 150

[3] Wang Liang and Langari R. Sugeno model, fuzzy discretization, and the EM algorithm [J]. Fuzzy Sets and Systems, 1996,82(3):279 - 288

[4] Wang shitong. Neural Fuzzy Systems and Their Applications [M]. Beijing: The Press House of Beijing Aeronautical University, 1998

[5] Jordan M and Xu L. Convergence results for the EM approach to the mixtures of expert architectures [J]. Neural Networks, 1995,9(4):1409 - 1431

本文作者简介

王士同　1964年生.教授,博士生导师.曾两次留学英国.正式发表过学术论文100余篇,论著6本.研究方向:模糊系统,人工智能,神经网络.

J.F.鲍尔温　1947年生.教授.Bristol大学高级计算机研究中心主任.研究方向:模糊系统,人工智能.

T.马 丁　1955年生.研究方向:模糊人工智能.

Main Contents of the Next Issue

A New Fault Diagnosis Method Based on Parameter Estimation ·························· XU Sichun and XIAO Deyun

Estimate of Exponential Convergence Rate for Hopfield-Type Continuous Neural Networks ·························· WEI Gang and TIAN Chuanjun

Analysis on the Performance and Properties of Interacting Multiple Models Algorithm ·························· LIANG Yan, PAN Quan, ZHANG Hongcai and DAI Guanzhong

Fast Fuzzy Orthogonal Least Squares Algorithm ·························· YANG Yi, WANG Jiang and WANG Xianlai

Stability Analysis of a Class of Time-Varying Systems with State Feed Back Control ·························· ZHAO Xiaohui

Wiener State Deconvolution Filters ·························· DENG Zili, SUN Shuli and GUO Jinzhu

State Feedback H_∞ Control Design Problem for Linear Singular Systems ·························· MA Shuping and CHENG Zhaolin

Two-Degree Optimal Robust Tracking Control Based on l_∞ and l_1 Norm Minimization ·························· LI Shengping

Self-Crossover Operator ·························· REN Qingsheng, ZENG Jin and QI Feihu

The Optimal Model of Hybrid Control for the Laminar Cooling Process of Hot Rolled Slab ·························· GUAN Shouping, CHAI Tianyou and GAO Jun

Controller for Active Power Distribution Based on Discrete Variable Structure Control Theory ·························· SONG Lizhong, PU Leping and YAO Qionghui

Preview Predictive Control for a Class of Following Systems ·························· WU Hai, ZONG Ming, YU Xichun and XU Xinhe

The Robotic Assembly System and Its Control for Motorbicycle Engine's Gas Valve Units ·························· ZHANG Tie, XIE Cunxi and SHAO Ming

One Optimum Design Method of Servo-System and Application Research ·························· YAO Yu and ZHANG Guojian