Incremental Hierarchical Fuzzy Model Generated from Multilevel Fuzzy Support Vector Regression Network

Ling Wang, DongMei Fu and LuLu Wu
School of Automation & Electrical Engineering, University of Science and Technology, Beijing, 100083, China
Key Laboratory of Advanced Control of Iron and Steel Process (Ministry of Education), Beijing, 100083, China

Keywords: FCM clustering, fuzzy association rules, incremental hierarchical fuzzy model, multilevel fuzzy support vector regression network

Received: March 23, 2014

Fuzzy rule-based systems are nowadays one of the most successful applications of fuzzy logic, but in complex applications with a large set of variables, the number of rules increases exponentially and the obtained fuzzy system is scarcely interpretable. Hierarchical fuzzy systems are one of the alternatives presented in the literature to overcome this problem. This paper presents a multilevel fuzzy support vector regression network (MFSVRN) model that learns incremental hierarchical structure based on the Takagi-Sugeno-Kang(TSK) fuzzy system with the aim of coping with the curse of dimensionality and generalization ability. From the input–output data pairs, the TS-type rules and its parameters are learned by a combination of fuzzy clustering and linear SVR in this paper. In addition, an efficient input variable selection method of the incremental multilevel network is proposed based on the FCM clustering and fuzzy association rules. To achieve high generalization ability, the consequence parameters of a rule are learned through linear SVR with a new TS-kernel. This paper demonstrates the capabilities of MFSVRN model by conducting simulations in function approximations and a chaotic time-series prediction. This paper also compares simulation results from the single-level counterparts-FSVRN and Jang's ANFIS model.

Povzetek:Predstavljen je hierarhični mehek model, zgrajen iz večnivojske mehke regresijske mreže.

1 Introduction

As is widely known, fuzzy rule-based systems have been proposed and successfully applied to solving problems such as classification, identification, control, etc. At present, one of the important issues in fuzzy rule-based systems is how to reduce the total number of involved rules and their corresponding computation requirements. In a single fuzzy system, in order to maintain a specified accuracy, the number of rules grows exponentially with the number of input variables or input fuzzy sets, but the interpretability of the fuzzy system is lost. Hence, to deal with the "curse of dimensionality" and the rule-explosion problem, hierarchical fuzzy system (HFS) was proposed [1] and have attracted much attention in recent years. Most hierarchical fuzzy systems [2-9] consist of a number of low-dimensional fuzzy systems in a hierarchical form, the number of rules can be reduced to as low as being a linear function of the number of input variables. A HFS is made up of a set of fuzzy subsystems or modules. These modules are linked such a way that the output of a module is the input of other ones. Fig.1 depicts some possible hierarchical models for 4 input variables and 2 hierarchical levels or 3 hierarchical levels. The incremental structure is shown in Fig.1(a), in which the inputs of a module is the output of the previous ones, along with external variables[2-6]. The aggregated structure is shown in Fig.1(b), in which the first level is made up of a set of modules receiving the input variables, and each variable is used as an input only in a single module. The outputs of the modules in the first level are the inputs of the modules which constitute the next level, and so on [7-9].

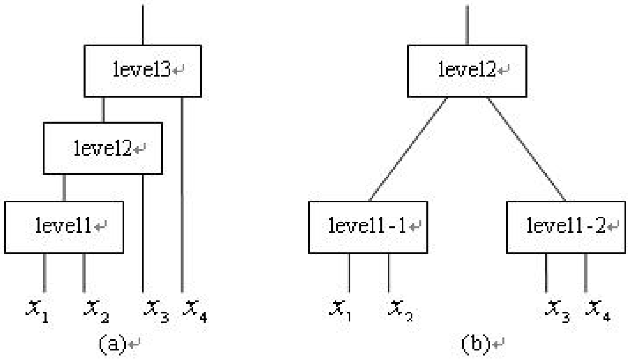

Figure1: Example of HFS Structure.

Our research focused on the incremental hierarchical structure. In this hierarchy, it is important to determine its input variables and their interactions in different levels. In general, by analyzing the relative importance of input variables, the most important input variables are assigned to the lowest level and the least important input variables are assigned to the highest level [12-13]. In order to assign the correlated or coupled input variables to the same level in hierarchical fuzzy system, Chung and Duan [13] introduced a correlation matrix to determine the correlated or coupled input variables. In[22], a multi-objective genetic algorithm (MOGA) is adopted to determine the input variables of each level module and the hierarchical architecture. In [6], an evolutionary algorithm (EA) is investigated to select the input variables and the topology in HFS. But GA and EA are all very time-consuming searching process.

In the hierarchical fuzzy system design, each single-level module serves as individual reasoning level. As a fuzzy reasoning mechanism, Takagi-Sugeno-Kang (TSK) type is most widely used. The well-known Jang's ANFIS[21] model based on layered network architecture was firstly proposed. In [10] and [11], the neural-fuzzy network is also used to realize TSK fuzzy reasoning. Usually, the consequent in Takagi-Sugeno fuzzy model is a crisp function of antecedent variables, and the recursive least squares (RLS) is used to determine the parameters of nonlinear consequents. However, this further RLS tuning is based on empirical risk minimization and lack of the high generalization ability. The idea of solving this parameter estimation problem by incorporating SVR (support vector regression) into TSK model has recently attracted a lot of attention [23-24]. In [23], a SVR-based FNN was proposed that the fuzzy if–then rules are generated based on the extracted SVs. Since the number of SVs in an SVR is usually very large, the model size of a designed FS is equally large. In [24], fuzzy c -means (FCM) is used to generate fuzzy rules and their antecedent parameters, the consequent part parameters are obtained by SVR learning. In contrast to fuzzy neural network, the use of SVR for TSK learning has a smaller number of parameters and the use of kernel transformation retains the SVR's good generalization ability.

This paper proposes a multilevel fuzzy support vector regression network (MFSVRN) model to design a fuzzy system to solve the dimensionality problem and generalization performance. In order to achieve efficient input variables in each level, we adopt the FCM clustering algorithm [15] and fuzzy association rules mining method [19] to construct a MLFSVRN model with incremental architecture. In addition, based on neuro-fuzzy systems, a novel TSK type fuzzy reasoning system with support vector regression learning mechanism called fuzzy support vector regression network (FSVRN) is applied to the modules of hierarchical fuzzy system. To improve the prediction performance of hierarchical fuzzy system, the consequence parameters of a rule are learned through linear SVR with a new TS-kernel. In the view of single-level FSVRN, the overall network-Multilevel fuzzy support vector regression network (MLFSVRN) is formulated.

The paper is organized as follows: Section 2 briefly describes the data mining techniques used in this paper. Section 3 presents our research methodology including input selection algorithms and construction of the multilevel support vector regression network models. Section 4 shows the experimental results. Finally, Section 5 concludes the paper with our final remarks.

2 Related knowledge

2.1 Fuzzy C-Means (FCM) clustering

Fuzzy c-Means (FCM) clustering algorithm [15-16] is an iterative optimization algorithm that minimizes the cost function (1) subjecting to $u_{ij} \in [0,1]$;

$$J(U,v)=\sum_{i=1}^{n}\sum_{j=1}^{c}u_{ij}^{m}\left\|\vec{x}_i-\vec{v}_j\right\|^2 \tag{1}$$

Here, u_{ij} is the degree of membership of $\vec{x}_i$ in the cluster j, which is defined as (2)

$$u_{ij}=\frac{(1/\left\|\vec{x}_i-\vec{v}_j\right\|^2)^{1/m-1}}{\sum_{k=1}^{c}(1/\left\|\vec{x}_i-\vec{v}_k\right\|^2)^{1/m-1}} \tag{2}$$

where $\vec{v}_j$ is the center of the j th cluster,

$$\vec{v}_j=\frac{1}{\sum_{i=1}^{n}(u_{ij})^m}\sum_{i=1}^{n}(u_{ij})^m\cdot\vec{x}_i \tag{3}$$

$\vec{x}_i$ is the i th data sample, n is the number of data points; $m\in(1,\infty)$ is a weighting constant.

The optimal clusters are produced by minimizing the objective function.

2.2 Fuzzy association rules

For the numerical data, fuzzy association rules [19] are easily understood by humans because of the fuzzy termsets. In order to mine fuzzy association rules, we apply FCM clustering to transform each of the numeric variables into fuzzy sets (fuzzy partition) with its membership function u, and then these fuzzy partitions are used to generate fuzzy rules. Meanwhile, the centre of each fuzzy set and the maximum and minimum value for each partition of the input data points are determined by FCM. By finding the centre of each partition, we can label it very easily according to the data point at which the core occurs. The labelling of each partition is very important as it helps a lot in the -eventual generation of fuzzy association rules.

Given a set of records, n is the record number, two items $X=\{x_1,x_2,\ldots,x_p\}$ and $Y=\{y_1,y_2,\ldots,y_q\}$, p is the length of the itemset X, q is the length of the itemset Y. The fuzzy set membership value of variable x_j in the i th record is denoted $u(x_{ij})$. The Apriror approach is used for extracting fuzzy itemsets from a fuzzy data set based on interesting measures(support and confidence) and able to generate fuzzy association rules. A fuzzy association rule is an implication of the form $(X\ is\ A)\Rightarrow(Y\ is\ B)$. A and B are fuzzy sets that characterize X and Y respectively. The measures of support, confidence and correlation have been fuzzified for the purpose of fuzzy association rules. The fuzzy support of X is defined as follows.

$$f_{support}(X)=\frac{1}{n}\sum_{i=1}^{n}\prod_{j=1}^{p}u(x_{ij}) \qquad (4)$$

Where $X=\{x_i\,|\,x_i=(x_{ij}\,|\,j=1,\cdots,p),\,i=1,\cdots,n\}$ .The fuzzy support of $X\Rightarrow Y$ is defined as follows.

$$f_{support}(X\Rightarrow Y)=\frac{1}{n}\sum_{i=1}^{n}\left(\prod_{j=1}^{p}u(x_{ij})\cap\prod_{j=1}^{q}u(x_{ij})\right) \qquad (5)$$

The fuzzy confidence of $X\Rightarrow Y$ is defined as follows.

$$f_{confidence}(X\Rightarrow Y)=\frac{f_{support}(X\Rightarrow Y)}{f_{support}(X)} \qquad (6)$$

The fuzzy correlation measure of the association rule $X\Rightarrow Y$ can be measured by computing

$$Correlation(X,Y)=\frac{P(X\cup Y)}{P(X)P(Y)}=\frac{f_{conference}(X\Rightarrow Y)}{f_{support}(Y)} \qquad (7)$$

In order to mine fuzzy association rules, the definitions of fuzzy support and fuzzy confidence are used in Fuzzy Apriori instead of their crisp counterparts used in Apriori.

3 Multilevel fuzzy SVR network (MLFSVRN) modelling of incremental type

In order to determine a multilevel fuzzy SVR network model, we must determine the model structure and initial parameters. Based on an incremental type structure like the one shown in Fig.1(a), it is quite possible to consider some influential variables to the first level, the less influential ones to the next level, and so on. To do so, we must find a set of candidate variables, which play a significant role in determining the output.

3.1 Variable selection based on FCM clustering and fuzzy association rules

In this paper, fuzzy association rules are used to select more representative variables from the original ones to improve the later prediction performance. The different fuzzy confidence and fuzzy support values between input variables and output variables are examined by using fuzzy association rules. In this paper, cross-validation with best parameter grid search is adopted to obtain the best rules. we only report the best rules that are based on the fuzzy confidence value = 1 and the fuzzy support value = 0.03.

These rules are further ranked as their importance by

$$Importance(X,Y)=\log\left(\frac{f_{confidence}(Y\Rightarrow X)}{f_{support}(X)}\right)=\log\left(\frac{p(X/Y)}{P(X)}\right) \qquad (8)$$

In particular, according to (8), the rule that has the importance value less than 0.8 is excluded. As a result, the extracted rules that relate to the important variables will be obtained. In order to determine the most influential input variables, the influence degree of variables are calculated. The term $ID(x_i)$ is used to represent the influence degree (ID) of x_i ,i.e., $ID(x_i)=\frac{important(x_i)+correlation(x_i)}{SUM_{ID}}$. Here, the term $important(x_i)$ is used to represent the best result of the degree of importance of the fuzzy association rules with x_i, which is done by (8), In a similar way, the term $correlation(x_i)$ is used to represent the best result of the degree of correlation of the fuzzy association rules with x_i, which is done by (7). For all extracted variables from fuzzy association rules, these two items are added up and described as

$$SUM_{ID}=\sum_{i=1}^{n}(important(x_i)+correlation(x_i)).$$

3.2 Constructing a MLFSVRN model with incremental architecture

Based upon the analysis method just described in Section 3.1, the influential input variables can be obtained and consequently a MLFSVRN model with incremental architecture can be constructed as shown in Fig.2. The construction algorithm of MLFSVRN Model with Incremental Architecture can be summarized as follows.

Step 1- Initialization:

The number of levels is h. This identified model is called model h and its output is denoted by $y^{(h)}$. All n input variables are put in a set S . Let $SUM_{ID}=\sum_{x_i\in S}(important(x_i)+correlation(x_i))$. A threshold value T_{inc} is set to control the model structure(the number of levels). A large T_{inc} will set the combination of the representative input variables rigorously and hence generate complicated networks while a small value will set the combination of the representative input variables loosely and generate the networks with few sublevels. Such arrangement is used to make the first level reserve enough system information and let the first level contain at least two input variables.

Step 2- Determination of Level 1:

1) Choose n_1 most influential inputs as the input variables of the first level and write them as $x_i^{(1)}$'s. In order to make the first level contain enough system information, the value of n_1 is determined by

$$\sum_{i=1}^{n_1}ID(x_i^{(1)})=\sum_{i=1}^{n_1}\frac{important(x_i^{(1)})+correlation(x_i^{(1)})}{SUM_{ID}}<T_{inc} \quad \text{and} \quad n_1\geq 2$$

(9)

According to (9), the first level of the model architecture contain at least two input variables. Here, the term $important(x_i^{(1)})$ is used to represent the best result of the degree of importance of the fuzzy association rules with $x_i^{(1)}$, the term $correlation(x_i^{(1)})$ is used to represent the best result of the degree of correlation of the fuzzy association rules with $x_i^{(1)}$.

2) Set the level index $h=2$. Remove these n_1 input variables from S.

Step 3— Recalculation:

Recalculating SUM_{ID} and the influence degree values of the variables left in S, i.e.,

$$ID(x_i)=\frac{important(x_i)+correlation(x_i)}{SUM_{ID}}\ \forall x_i\in S \qquad (10)$$

Step 4—Determination of level h:

Choose n_h most influential input variables, i.e., $x_i^{(h)}$'s, from S and assign them to level h. The number n_h is determined by $\sum_{i=1}^{n_h} ID(x_i^{(h)})>T_{inc}$ and $n_h\geq 2$.

Step 5—Termination:

Remove these n_h input variables from S. If S is empty, the algorithm terminates, otherwise go back to Step 3, let $h=h+1$.

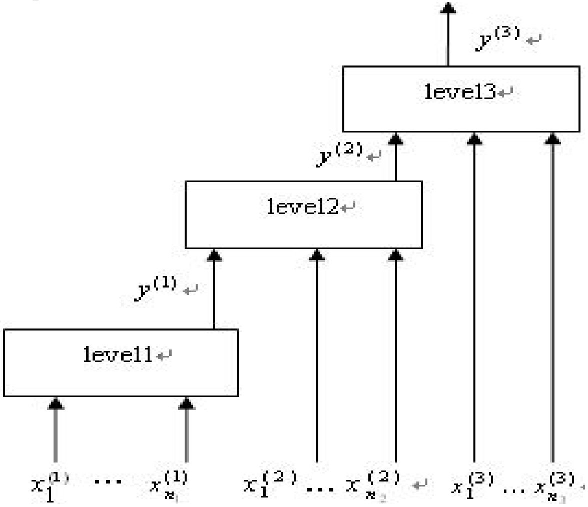
$y^{(3)}$
level13
$y^{(2)}$
level12
$y^{(1)}$
level11
$x_1^{(1)}$ ... $x_{n_1}^{(1)}$
$x_1^{(2)}$ ... $x_{n_2}^{(2)}$
$x_1^{(3)}$ ... $x_{n_3}^{(3)}$

Figure 2: The most parsimonious incremental architecture for a three-level-input system.

3.3 TSK type MLFSVRN model structure and learning

3.3.1 TSK fuzzy inference algorithm

In this section, a TSK fuzzy inference algorithm in its each single-level module of MLFSVRN model is presented. Consider a TSK fuzzy system, the j th fuzzy rule in level h will have the form

$\text{Rule}_j^{(h)}$: IF $\left(x_1^{(h)}\text{is}A_{1,j}^{(h)},\cdots,x_{n_h}^{(h)}\text{is}A_{n_h,j}^{(h)}\right)$ and $\left(y^{(h-1)}\text{is}B_j^{(h-1)}\right)$

$$\text{THEN}\qquad y^{(h)}=a_{0,j}^{(h)}+\sum_{i=1}^{n_h+1}a_{i,j}^{(h)}z_i^{(h)}. \qquad (11)$$

Where

n_h is the total number of input variables (ordinary system inputs and input from previous level).

$x_i^{(h)}(i=1,\cdots,n_h)$ is input variables being determined by the proposed method in Section 3 and assigned to this level;

$y^{(h)}(h\leq H)$ is intermediate variables which represent the output from level h as well as the input to level $h+1$, H is the total number of levels;

$y^{(H)}$ is output variable;

$z_1^{(h)}=x_1^{(h)}$, $z_2^{(h)}=x_2^{(h)}$, … , $z_{n_h}^{(h)}=x_{n_h}^{(h)}$, $z_{n_h+1}^{(h)}=y^{(h-1)}$;

$a_{0,j}^{(h)},a_{i,j}^{(h)}$ are crisp consequent part parameters;

Fuzzy set $A_{i,j}^{(h)}$ is employed with the following Gaussian-type membership function:

$$M_{ij}^{(h)}=\exp\left\{-\left(\frac{(z_i^{(h)}-m_{ij}^{(h)})^2}{\pm_j^2}\right)\right\} \qquad (12)$$

Here, $m_{ij}^{(h)}$ and $\pm_j$ correspond to the center and width of the fuzzy set respectively, which are determined by FCM clustering.

The firing strength $«_j^{(h)}(\vec{z})$ of rule j is calculated by

$$«_j^{(h)}(\vec{z})=\prod_{i=1}^{n_h+1}M_{ij}^{(h)}=\exp\left\{-\sum_{i=1}^{n_h+1}\left[\frac{(z_i^{(h)}-m_{ij}^{(h)})^2}{\pm_j^2}\right]\right\}$$
$$=\exp\left\{-\frac{\left\|\vec{z}^{(h)}-\vec{m}_j^{(h)}\right\|^2}{\pm_j^2}\right\} \qquad (13)$$

Where $\vec{z}^{(h)}=[x_1^{(h)},x_2^{(h)},\cdots,x_{n_h}^{(h)},y^{(h-1)}]$.

If the single-level fuzzy system contains r rules, then according to the simple weighted sum method [20], the output would be

$$y(\vec{x})=\sum_{j=1}^{r}\left[«_j^{(h)}(\vec{z})\cdot\left(a_{0,j}^{(h)}+\sum_{i=1}^{n_h+1}(a_{i,j}^{(h)}z_i^{(h)})\right)\right]$$
$$=\sum_{j=1}^{r}\left\{(\vec{z}^{(h)}\cdot\vec{a}_j^{(h)}+a_{0,j}^{(h)})\cdot\exp\left[-\frac{\left\|\vec{z}^{(h)}-\vec{m}_j^{(h)}\right\|^2}{\pm_j^2}\right]\right\} \qquad (14)$$

Where $\vec{a}_j^{(h)}=[a_{1,j}^{(h)},\ldots,a_{n_h,j}^{(h)}]$. By setting

$$\vec{a}_j^{(h)}=w_j^{(h)}[m_{1j}^{(h)}\cdots m_{n_hj}^{(h)}]=w_j^{(h)}\vec{m}_j^{(h)} \qquad (15)$$

Eq.(14) becomes

$$y(\vec{x})=\sum_{j=1}^{r}\left\{(w_j^{(h)}\vec{z}^{(h)}\cdot\vec{m}_j^{(h)}+a_{0,j}^{(h)})\cdot\exp\left[-\frac{\left\|\vec{z}^{(h)}-\vec{m}_j^{(h)}\right\|^2}{\pm_j^2}\right]\right\}$$
$$=\sum_{j=1}^{r}\left\{(w_j^{(h)}\vec{z}^{(h)}\cdot\vec{m}_j^{(h)})\cdot\exp\left[-\frac{\left\|\vec{z}^{(h)}-\vec{m}_j^{(h)}\right\|^2}{\pm_j^2}\right]\right\}$$
$$+\sum_{j=1}^{r}a_{0,j}^{(h)}\exp\left[-\frac{\left\|\vec{z}^{(h)}-\vec{m}_j^{(h)}\right\|^2}{\pm_j^2}\right] \tag{16}$$

By adopting the kernel trick, a TS-kernel is integrated as

$$K_{TS}(\vec{z}^{(h)},\vec{m}_j^{(h)})=(\vec{z}^{(h)}\cdot\vec{m}_j^{(h)})\cdot\exp\left[-\frac{\left\|\vec{z}^{(h)}-\vec{m}_j^{(h)}\right\|^2}{\pm_j^2}\right].$$

And setting $\sum_{j=1}^{r}a_{0,j}^{(h)}\exp\left[-\frac{\left\|\vec{z}^{(h)}-\vec{m}_j^{(h)}\right\|^2}{\pm_j^2}\right]=b^{(h)}$

then (16) can be further rewritten as:

$$y(\vec{x})=\sum_{j=1}^{r}w_j^{(h)}K_{TS}(\vec{z}^{(h)},\vec{m}_j^{(h)})+b^{(h)} \tag{17}$$

Eq.(17) is the output of TSK-type fuzzy system, which is equivalent to the output of the SVR.

For each level module, the parameters $\vec{m}_j$ and $\pm_j$ in the TS-kernels and the number of rules are determined by FCM clustering, the weighting parameters $w_j^{(h)}$ and bias $b^{(h)}$ are determined by the linear SVR. Each input data $\vec{z}^{(h)}$ is transformed to the vector $\vec{V}(\vec{z}^{(h)})=[V_1(\vec{z}^{(h)}),V_2(\vec{z}^{(h)}),\ldots,V_r(\vec{z}^{(h)})]$, where $V_j(\vec{z}^{(h)})=K_{TS}(\vec{z}^{(h)},\vec{m}_j^{(h)})$ is the output of the j th TS-kernel. The vector is fed as input to a linear SVR, and the training data pairs are represented by

$$S=\left\{(\vec{V}(\vec{z}_1^{(h)}),y_1^{(h)}),(\vec{V}(\vec{z}_2^{(h)}),y_2^{(h)}),\ldots,(\vec{V}(\vec{z}_N^{(h)}),y_N^{(h)})\right\} \tag{18}$$

The optimal linear SVR function is

$$y^{(h)}(\vec{z}^{(h)})=\sum_{k=1}^{N}(\sim_k^{(h)}-\hat{\sim}_k^{(h)})\left\langle V(\vec{z}^{(h)}),V(\vec{z}_k^{(h)})\right\rangle+b^{(h)} \tag{19}$$

Where $\sim_k^{(h)}$ and $\hat{\sim}_k^{(h)}$ are solved by SVR. Eq.(19) can be represented as

$$y^{(h)}(\vec{z})=\sum_{k=1}^{N}(\sim_k^{(h)}-\hat{\sim}_k^{(h)})\sum_{j=1}^{r}V_j(\vec{z}^{(h)})V_j(\vec{z}_k^{(h)})+b^{(h)}$$
$$=\sum_{j=1}^{r}\left(\sum_{k=1}^{N}(\sim_k^{(h)}-\hat{\sim}_k^{(h)})V_j(\vec{z}_k^{(h)})\right)V_j(\vec{z}^{(h)})+b^{(h)} \tag{20}$$
$$=\sum_{j=1}^{r}w_j^{(h)}V_j(\vec{z}^{(h)})+b^{(h)}=\sum_{j=1}^{r}w_j^{(h)}K_{TS}(\vec{z}^{(h)},\vec{m}_j^{(h)})+b^{(h)}$$

Where $w_j^{(h)}=\sum_{k=1}^{N}(\sim_k^{(h)}-\hat{\sim}_k^{(h)})V_j(\vec{z}_k^{(h)})$ (21)

According to Eq.(21), the weighting parameters $w_j^{(h)}$ are obtained.

3.3.2 Structure of single-level modules

The structure of each single-level FSVRN module is based on the structure like the four-layered Fuzzy neural network in Fig.3, which consists of the membership function, fuzzy rules, weighted consequent and output layers, and their functions are briefly described below in the our context. In the following descriptions, $O_{i,j}^{(h)}$ represents the output of the i th node in j th layer of level h, $u_l^{(h)}$ is the l th input of a certain node in current layer of level h.

Layer 1: Each node in this layer corresponds to a membership function of ordinary system inputs and input from previous level [see (22) and (23)].

$$O_{i,1}^{(h)}=M_{i,1}^{(h)}(x_k^{(h)}),$$
$$\forall i=1,2,\cdots,N_1^{(h)}-1,h=2,\cdots,H \tag{22}$$

and

$$O_{N_1^{(h)},1}^{(h)}=M_{N_1^{(h)},1}^{(h)}(y^{(h-1)})\quad\forall h=2,\cdots,H \tag{23}$$

Here, $M_{i,1}^{(h)}(x_k^{(h)})$ and $M_{N_1^{(h)},1}^{(h)}(y^{(h-1)})$ are the corresponding membership value of the input (or intermediate) variable connected to it by (12). $N_1^{(h)}$ is equal to the total number of inputs to level h, including the membership value of system inputs and intermediate input. When $h=1$, there will not exist any intermediate inputs and only (22) is applied.

Layer 2: The output of each node in this layer represents the firing strength of a rule

$$O_{i,2}^{(h)}=\prod_{l=1}^{r_i}\left(u_l^{(h)}\right)\quad\forall i=1,2,\cdots,N_2^{(h)}. \tag{24}$$

The determination of (24) is similar with (13). r_i denotes the number of preconditions in rule node i and $N_2^{(h)}$ indicates the total number of rules in level h.

Layer 3: The output of each rule is computed in this layer.

In level $h=1$ where no intermediate variable appears, the function has the form

$$O_{i,3}^{(1)} = u_l^{(1)}\left(\sum_{j=0}^{n_1} a_{j,i}^{(1)} x_j^{(1)}\right) \qquad \forall i = 1,2,\cdots,N_3^{(1)} \tag{25}$$

and when $h > 1$

$$O_{i,3}^{(h)} = u_l^{(h)}\left(\sum_{j=0}^{n_h+1} a_{j,i}^{(h)} z_j^{(h)}\right) \qquad \forall i = 1,2,\cdots,N_3^{(h)} \tag{26}$$

where $N_3^{(1)}$ and $N_3^{(h)}$ correspond to the total number of nodes in the third layer for Level 1and Level 3,respectively. $\left\{a_{j,i}^{(h)}\right\}$is the consequent parameter set in level h, $z_0^{(h)} = 1$;.

Layer 4: The final output is computed by summing the outputs of all rules

$$O_{1,4}^{(h)} = \sum_{j=1}^{N_R^{(h)}} u_l^{(h)} \tag{27}$$

$N_R^{(h)}$ is the total number of fuzzy rules in level h, which equals to $N_2^{(h)}$. Eqs.(14), (16), (17), (18) and (20) determine the output in (27).

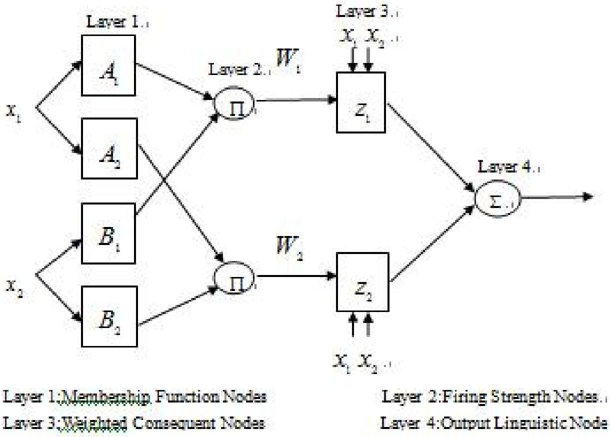
Layer 1.
Layer 2.
Layer 3.
Layer 4.
W_1
W_2
x_1
x_2
A_1
A_2
B_1
B_2
Π
Π
z_1
z_2
Σ
x_1 x_2
x_1 x_2
Layer 1:Membership Function Nodes
Layer 2:Firing Strength Nodes.
Layer 3:Weighted Consequent Nodes
Layer 4:Output Linguistic Node

Figure 3: The structure of the four-layered network.

3.3.3 Learning algorithm

As we have found in 3.3.1, the consequent part of the TSK fuzzy inference rule is a linear combination of all consequent parameters, which can be reconstructed in the form of linear SVR. So, the linear SVR algorithm is first applied to evaluate the optimal values of all consequent parameters of TSK-type MLFSVRN model. Given a set of training data pairs and setting the desired output of each single-level module as same as the final system output. The linear SVR learning algorithm of a TSK-type MLFSVRN model with incremental architecture is listed as follows.

Step 1—Initialization:

1) Divide the input variables into H subsets $\left\{x_1^{(h)}, x_2^{(h)}, \cdots, x_{n_h}^{(h)} \middle| h = 1,\cdots,H\right\}$according to the variable selection method proposed in Section 3, each of them attached to a single-level reasoning network module.

2) Set the level index $h = 1$ and initialize appropriate membership function parameters based on the FCM clustering method. The membership function parameters of all intermediate variables are fixed according to the final outputs.

Step 2—Apply Linear SVR to Level h:

In order to evaluate the optimal value of all unknown consequent parameters $a_{j,i}^{(h)}$, (25),(26) and(27) can be rewritten in the form of linear SVR according to Eqs.(14), (16), (17), (18) and (20) . The consequent parameters then can be evaluated using Linear SVR method with (15) and (21). Here, the output value y will be used as the desired value of $y^{(h)}$.

Step 3—Forward Computation:

The output of Level h, $y^{(h)}$, can be computed using the evaluated $a_{j,i}^{(h)}$'s.

Step 4—Termination:

Set $h = h + 1$. If $h \le H$, go to Step 2; otherwise, the training process stops.

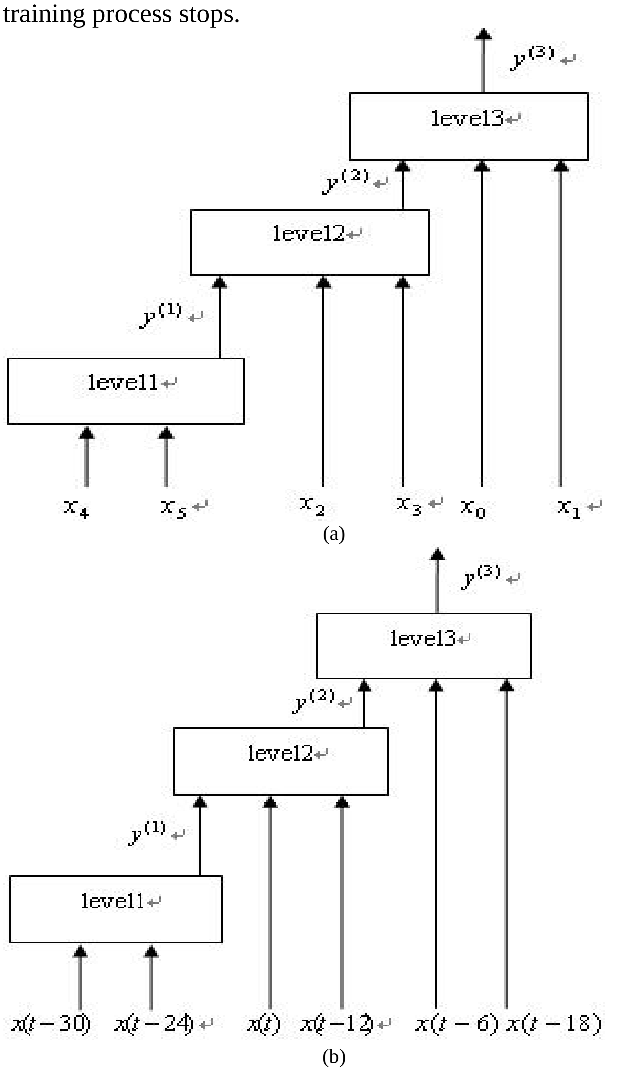

Figure 4: MLFSVRN model with incremental architecture (a) six-dimensional example (b) Mackey-Glass time series.

4 Simulation results

In this section, the proposed method has been evaluated for nonlinear system identification and Mackey-Glass chaotic time-series prediction. Section 4.1 discusses a six-dimensional example, which is used to validate the variable selection analysis method described in Sections 3. Section 4.2 discusses the Mackey–Glass chaotic time series prediction, aiming at demonstrating the satisfactory learning behavior and good generalization ability of the MLFSVRN models.

4.1 Six-dimensional example

The six-dimensional nonlinear system was given by the following equation:

$$y = (-4 + x_0^{0.5} + x_1^{-1})^2 + 5\sin(x_2 + x_3) + \exp(1 + x_4 + x_5) \qquad (28)$$

A data set of 1000 pair was prepared by drawing the inputs uniformly from the six-dimensional unit hypercube. To construct the MLFSVRN with incremental architecture, using the variable selection analysis method proposed in Section 3, the input variables are grouped into three subsets $\{x_4, x_5\}$ 、 $\{x_2, x_3\}$ 、 $\{x_0, x_1\}$ .The influence degree of each input variable is evaluated and listed as follows:

x_i	x_0	x_1	x_2	x_3	x_4	x_5
ID(x_i)	0.109	0.107	0.176	0.178	0.273	0.270

It can be seen that x_4 is the most influential input among the six and this corresponds very well to what we can deduce from (28). Furthermore, the influence degree of x_0 and x_1, x_2 and x_3, x_4 and x_5 is about the same and this again matches with our expectation from (28). Thus, the effectiveness of the variable selection method is demonstrated by this example. Here, the threshold is chosen as T_{inc} =0.6, so the incremental architecture can have x_4 and x_5 assigned to the first level as inputs. For the other four input variables, x_2 and x_3 are put to the second level, x_0 and x_1 are put to the third level, as shown in Fig.4(a).

	Rules		Error-Train	Error-Test
TSK MLFSVRN With Incremental Architecture	Level 1	4	0.0168	0.0135
	Level 2	8		
	Level 3	8		
	Total	20		
FSVRN	48		0.0003	0.0126
Jang's ANFIS	64		0.0005	0.0157

Table 1: Comparison of the TSK-type MLFSVRN models from their single-level counterpart FSVRN and Jang's ANFIS in Function prediction.

The performance of the proposed TSK-type MLFSVRN models with incremental architecture has been evaluated with the Root Mean Square Error (RMSE). After the liner SVR learning phase with 400 training data pairs, the models were validated by the testing data set consisting of 600 points. For the purpose of comparison, the performance of the TSK-type MLFSVRN models, their single level counterpart, i.e., FSVRN model and Jang's ANFIS model [21] with 400 training data are listed in Table 1. Obviously, the TSK-type MLFSVRN models has 3 levels, in which the number of terms for each input is 2, 4, 4, respectively. The total number of rules is 20, which use much less fuzzy rules and adjustable parameters than single-level FSVRN. Furthermore, although the single-level counterparts-FSVRN has the smallest training RMSE and testing RMSE, the number of fuzzy rules is more than our proposed model. The testing RMSE of Jang's ANFIS model is biggest among these three models because it cannot have correct response to unforeseen inputs when the training samples are limited. So, the TSK-type MLFSVRN model with incremental architecture shows relatively better generalization ability.

4.2 Prediction of a chaotic time-series

The Mackey-Glass chaotic differential delay equation is recognized as a benchmark problem for time-series prediction which frequently used in the study of chaotic dynamics and defined as follows [14]:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-{}^2)}{1+x^{10}(t-{}^2)} - 0.1x(t) \qquad (29)$$

When ${}^2 > 17$, the equation shows chaotic behavior. In our simulations, we set ${}^2 = 30$. In this paper we used $x(t-30)$, $x(t-24)$, $x(t-18)$, $x(t-12)$, $x(t-6)$ and $x(t)$ as input variables to predict the value of $x(t+6)$.

To construct the MLFSVRN with incremental architecture, using the variable selection method proposed in section 3, the input variables are grouped into three subsets $\{x(t-24), x(t-30)\}$ 、 $\{x(t-18), x(t-6)\}$ 、 $\{x(t-12), x(t)\}$ .The influence degree of each input variable is computed and listed as follows:

x_i	$x(t-30)$	$x(t-24)$	$x(t-18)$	$x(t-12)$	$x(t-6)$	$x(t)$
ID(x_i)	0.253	0.327	0.087	0.165	0.074	0.173

It is found that $x(t-30)$ and $x(t-24)$ are two most influential input variables. Among three clusters, the combination of the influence degree of $x(t-30)$ and $x(t-24)$ is biggest and less than the threshold T_{inc} =0.6 which consistently follow the algorithm in section 3. So the incremental architecture can have $x(t-30)$ and $x(t-24)$ assigned to the first level as inputs. For the other four input variables, $x(t-18)$ and $x(t-6)$ are put to the second level, $x(t-12)$ and $x(t)$ are put to the third level, as shown in Fig.4(b).

	Rules		Error-Train	Error-Test
TSK MLFSVRN With Incremental Architecture	Level 1	9	0.0253	0.0262
	Level 2	9		
	Level 3	9		
	Total	27		
FSVRN	35		0.0258	0.0352
Jang's ANFIS	39		0.0275	0.0408

Table 2: Comparison of the TSK-type MLFSVRN models from their single-level counterpart FSVRN and Jang's ANFIS in Mackey-Glass chaotic prediction.

In order to evaluate the performance of TSK-type MLFSVRN models with the Root Mean Square Error (RMSE), the 200 points of the series from $t = 501 - 700$ and a comparatively larger one consisting of 700 points of the series from $t = 130 - 829$ are used as training data, and 500 points from $t = 830 - 1329$ are used as testing data. According to the variable selection method proposed in section 3, we obtained 3 levels and the number of fuzzy rules in each level of the incremental architecture is 9. For the purpose of comparison, the performance of the proposed TSK-type MLFSVRN models, their single level counterpart, i.e., FSVRN model and Jang's ANFIS model with 700 training points are listed in Table 2. From that, it can be seen that the proposed TSK-type MLFSVRN models uses much less fuzzy rules than other models. Furthermore, it is found that the MLFSVRN models perform best among the three models in terms of training and testing RMSE. Fig.5 shows that the TSK-type MLFSVRN prediction outputs are close the real outputs and achieves good generalization ability.

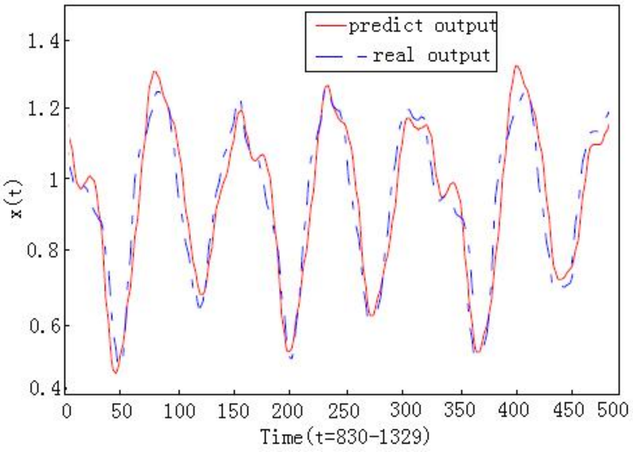

Figure 5: Test Result of the TSK-type MLFSVRN model.

The above simulations show that our proposed MLFSVRN models can get rid of the dimensionality problem fundamentally. TSK-type MLFSVRN models consume much less fuzzy rules compared with their single-level counterparts-FSVRN and Jang's ANFIS. TSK-type MSFNN models save both fuzzy rules and adjustable parameters significantly compared with Jang's ANFIS.

5 Conclusion

In this paper, a hierarchical TSK-type fuzzy system was proposed and its applications in system identification and time-series prediction were studied. In the proposed method, the major characteristic of such model is that the consequence of a rule will be used as a fact to another rule from which the number of fuzzy rules resulted will no longer be an exponential function of the number of input variables. The proposed MLFSVRN model is constructed with incremental architecture. First, some influential input variables are arranged to different reasoning levels by analyzing the influence degree of each input variable based on FCM clustering and fuzzy association rules. Then, each level reasoning module can be realized by FSVRN model. Its consequent parameters are learned by a linear SVR with a new TS-kernel. The major advantage of using MLFSVRN model other than a single-level fuzzy system is that the number of fuzzy rules and parameters involved in modelling process can be reduced significantly and the generalization ability can be improved when compared with those required by the single-level FSVRN systems and Jang's ANFIS systems. The effectiveness of the MLFSVRN model has been demonstrated through two problems. It can generally be concluded that the proposed method has higher performance in identification and time-series prediction in comparison with the other methods.

Funding

This paper is supported by the Fundamental Research Funds for the Central Universities (No.FRF-SD-12-009B) and the State Scholarship Fund.

References

[1] G. V. S. Raju, J. Zhou, and R. A. Kisner (1991), "Hierarchical fuzzy control", Int. J. Contr., vol. 54 no. 5, pp. 1201–1216.

[2] Cheong F (2007), "A hierarchical fuzzy system with high input dimensions for forecasting foreign exchange rates". IEEE Congress on Evolutionary Computation, CEC (Singapore), pp. 1642–1647.

[3] Aja-Ferna´ndez S, Alberola-Lo´pez C (2008), "Matriz modeling of hierarchical fuzzy systems", IEEE Trans Fuzzy Syst, vol. 16, no. 3, pp. 585–599.

[4] Zeng X, Goulermas J, Liatsis P, Wang D, Keane J (2008), "Hierarchical fuzzy systems for function approximation on discrete input spaces with application", IEEE Trans Fuzzy Syst, vol. 16 no. 5, pp. 1197–1215.

[5] Benftez A, Casillas J (2009), "Genetic learning of serial hierarchical fuzzy systems for large-scale problems". Proceedings of Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference (IFSA-EUSFLAT, Lisbon), pp. 1751–1756.

[6] Zajaczkowski J, Verma B (2008), "Selection and impact of different topologies in multilayered

hierarchical fuzzy systems", Appl Intell, vol. 36, no. 3, pp. 564–584.

[7] Salgado P (2008), "Rule generation for hierarchical collaborative fuzzy system", Appl Math Modell Sci Direct, vol. 32, no. 7), pp. 1159–1178.

[8] Joo M, Sudkamp T (2009), "A method of converting a fuzzy system to a two-layered hierarchical fuzzy system and its run-time efficiency", IEEE Trans Fuzzy Sys, vol. 17, no. 1, pp. 93–103.

[9] Jelleli T, Alimi A (2010), "Automatic design of a least complicated hierarchical fuzzy system". IEEE International Conference on Fuzzy Systems (FUZZ), pp. 1–7.

[10] S. Mitra and Y. Hayashi (2000), "Neuro-fuzzy rule generation: Survey in soft computing framework", IEEE Trans. Neural Net, vol. 11, no. 3, pp. 748–768.

[11] M. F. Azeem, M. Hanmandlu, N. Ahmad (2003), "Structure Identification of Generalized Adaptive Neuro-Fuzzy Inference Systems", IEEE Trans. On Fuzzy Systems, vol. 11, no. 5, pp. 666-681.

[12] L. X. Wang (1999), "Analysis and design of hierarchical fuzzy systems", IEEE Transactions on Fuzzy systems, vol. 7, no. 5, pp. 617-624.

[13] F. L Chung and J. C. Duan (2000), "On multistage fuzzy neural network modelling", IEEE Transactions on fuzzy systems, vol. 8, no. 2, pp. 125-142.

[14] M. C. Mackey and L. Glass (1977), "Oscillation and chaos in physiological control systems", Sci., vol. 197, pp. 287–289.

[15] J. Bezdek (1981), Pattern Recognition With Fuzzy Objective Function Algorithms.New York: Plenum.

[16] W. Pedrycz (2002), "Collaborative Fuzzy Clustering", Pattern Recognition Letters, vol. 23, no. 14, pp. 1675-1686.

[17] Agrawal R, Imielinski T, Swami A (1993), "Mining association rules between sets of items in large databases". In Proceedings of the ACM SIGMOD conference on management of data, pp. 207–216.

[18] Kantardzic M,John Wiley and Sons (2003), Data mining -Concepts, models, methods, and algorithms.

[19] Y. C. Lee, T. P. Hong and W. Y. Lin (2004), "Mining fuzzy association rules with multiple minimum supports using maximum constraints". The Eighth International Conference on Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, pp. 1283-1290.

[20] B. Schölkopf, A.J. Smola (2002), Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge MA. : MIT Press.

[21] Jang, J.-S.R (1993). "ANFIS: adaptive-network-based fuzzy inference system". IEEE Transactions on systems, Man and Cybernetics, vol. 23, no. 3, pp. 665-685.

[22] Tachibana K, Furuhashi T (2002). "A structure identification method of submodels for hierarchical fuzzy modeling using the multiple objective genetic algorithm", Int J Intell Syst, vol. 17, no. 5, pp. 495–513.

[23] C. T. Lin, S. F. Liang, C. M. Yeh, and K. W. Fan (2005), "Fuzzy neural network design using support vector regression for function approximation with outliers", IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2763-2768.

[24] J. M. Leski (2005), "TSK-fuzzy modeling based on -insensitive learning", IEEE Trans, Fuzzy Systems, vol. 13, no. 2, pp. 181-193.