Proceedings of the International Symposium on Evolving Intelligent Systems, Plamen Angelov, Dimitar Filev, and Nikola Kasabov (Eds.), at the AISB 2010 convention, 29 March – 1 April 2010, De Montfort University, Leicester, UK

A Structure Evolved Learning Method for Mamdani Fuzzy Systems

Di Wang, Xiao-Jun Zeng, and John A. Keane

***Abstract*—This paper proposes a Structure Evolved Learning Method (SELM) for Mamdani Fuzzy Systems. The mechanism of the algorithm is that it is an error-reducing driven learning method. Beginning with a simplest fuzzy system, the algorithm develops the system structure by adding more fuzzy terms and rules to reduce the model errors in a 'greedy' way. The main features of the proposed algorithm are that, firstly, it can automatically determines and controls the number and location of fuzzy terms needed by following the error-reducing driven evolving process to achieve the desired accuracy; secondly, it adds new fuzzy terms and rules by evenly distributing error to each sub-region aiming at an efficient set of fuzzy rules, thirdly, it uses triangular membership functions with the regular partitions and Mamdani fuzzy systems and leads to identified fuzzy system models with good transparency and interpretability. Two benchmark examples are given to illustrate the advantages of the proposed algorithm.**

Keywords- fuzzy systems, evolved learning, fuzzy systems identification.

I. INTRODUCTION

Since the invention of fuzzy sets and fuzzy systems by Zadeh four decades ago, fuzzy systems have been successfully applied in different areas. There are several fundamental reasons for the achievements of fuzzy systems [1]-[7]: firstly, fuzzy systems can be identified by combining human knowledge and learning from data; secondly, the transparency and interpretability of fuzzy rules and systems make them easy to understand, use and judge in applications, thirdly, fuzzy systems are universal approximators which can approximate any continuous nonlinear function to any degree of accuracy and therefore are applicable to be used to solve general complex and nonlinear problems.

In traditional fuzzy systems, fuzzy terms are defined by a global grid-based partition. There are three drawbacks by this global grid-based definition. Firstly, a global grid-based partition inherently suffers the exponential growth of fuzzy rules and parameters corresponding to the number of input variables; Secondly not all the fuzzy terms obtained by a global grid-based partition on each input variable are effective; Thirdly, this fixed definition of global grid-based partition makes the whole fuzzy systems unintelligent and inflexible.

Di Wang is with Manchester Business School, the University of Manchester, Booth Street West, Manchester, M15 6PB (e-mail: wangdi_2001@hotmail.com).

Xiao-Jun Zeng and John A. Keane are with the School of Computer Science, the University of Manchester, Manchester, M13 9 PL, UK (e-mail: x.zeng@manchester.ac.uk, John.Keane@manchester.ac.uk).

To overcome these weaknesses, various different methods have been developed. Adaptive fuzzy or neuro-fuzzy method has become one of the most important approaches in this aspect [8]-[37]. In particular, eTS, DENFIS, SOFNN, and SAFIS have been proposed by Angelov[10]-[11], ,Kasabov [12], Leng and al [13]-[14], and Rung and al [15] respectively. These methods have been proved having the capability to automatically identify the structure and parameters of complex nonlinear systems with good numerical accuracy.

Motivated by the weaknesses in traditional fuzzy systems, this paper proposes a Structure Evolved Learning Method for (SELM) fuzzy systems identification. SELM starts with a simplest fuzzy system with two fuzzy terms for each input variable and develops the structure of the fuzzy system by increasing fuzzy terms and rules on the selected input variable in the selected sub-region step by step. Different from the existing methods such as eTS, DENFIS, SOFNN, and SAFIS, the main features of SELM are as follows:

- SELM is an error-reducing driven method. That is, the structure of a fuzzy system evolves along with the aim of reducing the errors until the model error is within an accepted threshold. In other words, the method is focusing directly the goal of learning – to represent a given system with a desired accuracy. This leads to the mechanism of SELM in structure identification, that is, SELM automatically determines the number and location of fuzzy terms needed by following the error-reducing driven evolving process to achieve the desired accuracy.
- SELM is a greedy incremental process with the goal to achieve the most effective error-reducing. For this purpose, fuzzy terms and rules are added at point within selected region which evenly divides its local error along the input variable to get the best accuracy in one step. This step-by-step adding can be regarded as 'greedy construction' strategy. As all 'greedy' methods, SELM makes the locally optimum choice at each step of incremental construction with the hope of finding the global optimum [18]-[19].
- SELM uses triangular membership functions with regular partitions and Mamdani fuzzy systems in order to achieve the transparency and interpretability of the resulting fuzzy systems.

This paper is organized as follows: Section II presents Structure Evolved Learning Method (SELM) for Fuzzy Systems. Section III presents the proposed SELM algorithm for Mamdani fuzzy system. Section IV gives two examples to illustrate the advantages of the proposed algorithm. Finally Section V summarizes the main conclusion.

II. Structure Evolved Learning Method for Fuzzy Systems

A. Structure evolution

In this section, we introduce the proposed scheme for evolved structure learning for fuzzy systems. We are using triangular membership functions with regular partitions and Mamdani fuzzy systems to achieve the better transparency and interpretability.

Initially we consider the problem definition domain as a whole region. We then split this whole region into two sub-regions to obtain even error distribution between these two sub-regions. Then we select one sub-region with maximal average error for further splitting. This splitting process continues until we obtain the satisfactory accuracy. In this incremental construction process, a sub-region having the maximal average local error is always selected for further splitting.

For the selected sub-region, we locate a proper data instance candidate along each attribute (the *i*th attribute, i.e. the input variable x_i) which evenly split the accumulated error for the selected sub-region. Then one attribute $x_s^{splitting}$ is selected for splitting if the best global accuracy is obtained comparing with splitting on other attributes。

The splitting point $x_s^{splitting}$ is identified as a new partition point on that input variable. The main idea behind this can be explained as follows: We partition the input variables unevenly step by step. The aim of this uneven partition is to distribute error more evenly to each sub-region in next step. This even error distribution policy helps to create the most efficient new fuzzy terms and rules.

We can visualize this uneven splitting process by Fig.1. Fig. 1 (a) shows the definition domain and Fig. 1 (b) shows the case to construct a fuzzy system with a global even grid-based partition. This even grid-based partition is predefined. Fig. 1 (c) and Fig. 1 (d) show the incremental splitting for SELM. If in Fig. 1 (c) the average error in Sub-region 1 is larger than that in Sub-region 2, then Sub-region 1 is selected for further splitting. It is spitted into Sub-region 1' and Sub-region 3 in Fig. 1 (d). Comparison with global even grid-based partition, we see that the proposed incremental construction scheme avoids the exponentially increasing of fuzzy terms/rules suffered by the global even grid-based partition as the number of sub-regions increases by only one each time.

Remark1: SELM is error-reducing driven and aims at reducing the errors by adding rules step by step. As other methods for system identification, SELM needs to balance accuracy and system complexity. Too few rules result in under fitting and inaccurate results whilst too many rules result in complicated system structure and over fitting (especially in presence of noise). To avoid over fitting, we need to set a proper expected accuracy as a stop critical for system splitting. A proper splitting should result in a system with similar training accuracy and testing accuracy.

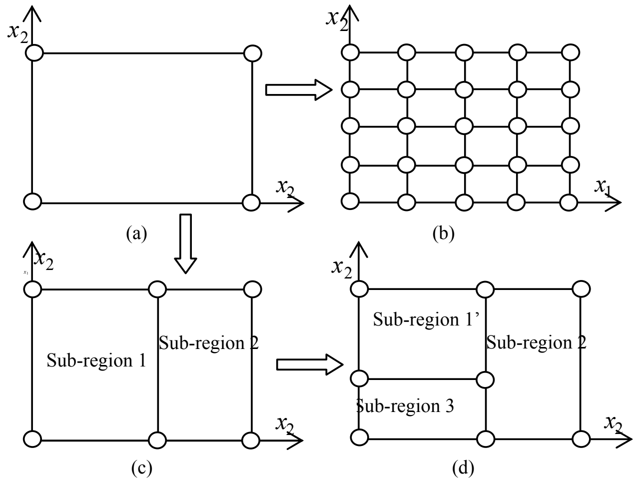

Fig.1. Sub-region creation for traditional fuzzy system and SELM

B. Formulations

In this section, to explain the formula for SELM, we consider a problem with *n* input variables.

As described in Sub-section II.A, SELM starts with one whole region. The definition domain for an n-dimensional problem is $\left(\times_{i=1}^{n}[b_i, c_i]\right)$, where b_i and c_i are the minimal and maximal value for the *i*th attribute. Then the initial Mamdani fuzzy system has the rule base as follows:

IF x_1 is $A_{1,1}^{k_1}$, and x_2 is $A_{1,2}^{k_2}$, … and x_n is $A_{1,n}^{k_n}$, THEN y is $y^{k_1 k_2 \ldots k_n}$, (1)

where k_i is 0 or 1. If $k_i = 0$, then $A_{1,i}^{k_i} = A_{1,i}^{0} = b_i$; if $k_i = 1$, then $A_{1,i}^{k_i} = A_{1,i}^{1} = c_i$. $y^{k_1 k_2 \ldots k_n}$ is parameters to be identified.

The fuzzy system starts from the simplest rule base (1), and continues with sub-region splitting process on the selected domain. In the following discussion, *m* is used to indicate the index of the selected sub-region for further splitting. The sub-region having the maximal average local error is always selected for further splitting.

Assume that input vector **x** belongs to the *m*th sub-region, i.e., $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \times_{i=1}^{n}[b_{m,i}, c_{m,i}]$, where $b_{m,i}$ and $c_{m,i}$ are the minimal and maximal value for the *i*th attribute x_i for the *m*th sub-region. Then the fuzzy rule base for the *m*th sub-region is:

IF x_1 is $A_{m,1}^{k_1}$, and x_2 is $A_{m,2}^{k_2}$, … and x_n is $A_{m,n}^{k_n}$, THEN y is

$$y_m^{k_1 k_2 \dots k_n}, \tag{2}$$

where k_i is 0 or 1. If $k_i = 0$, then $A_{m,i}^{k_i} = A_{m,i}^0 = b_{m,i}$; if $k_i = 1$, then $A_{m,i}^{k_i} = A_{m,i}^1 = c_{m,i}$,

The regular triangular membership functions for that the *m*th sub-region (in Equation (2)) are defined as follow.

The membership function for fuzzy term $A_{m,i}^0(x_i)$ (when $k_i = 0$) is:

$$\mu_{m,i}^0(x_i) = \frac{c_{m,i} - x_i}{c_{m,i} - b_{m,i}}. \tag{3}$$

The membership function for fuzzy term $A_{m,i}^1$ ($x_{m,i}$) (when $k_i = 1$) is:

$$\mu_{m,i}^1(x_i) = \frac{x_i - b_{m,i}}{c_{m,i} - b_{m,i}}. \tag{4}$$

Then the output for the *m*th sub-region is:

$$o_m = \frac{\sum\limits_{k_1 k_2 \dots k_n} \left(A_m^{k_1 k_2 \dots k_n}(\mathbf{x}) \times y_m^{k_1 k_2 \dots k_n} \right)_{\mathbf{x} \in domain_m}}{\sum\limits_{k_1 k_2 \dots k_n} A_m^{k_1 k_2 \dots k_n}(\mathbf{x}) |_{\mathbf{x} \in domain_m}} \tag{5}$$

where

$$\sum_{k_1 k_2 \dots k_n} A_m^{k_1 k_2 \dots k_n}(\mathbf{x}) |_{\mathbf{x} \in domain_m} = 1 \tag{6}$$

$$A_m^{k_1 k_2 \dots k_n}(\mathbf{x}) = \mathop{\times}_{i=1}^{n} \left(\mu_{m,i}^{k_i}(x_i) \right). \tag{7}$$

Based on (6), Equation (5) can be written as:

$$o_m = \sum_{k_1 k_2 \dots k_n} A_m^{k_1 k_2 \dots k_n}(\mathbf{x}) \times y_m^{k_1 k_2 \dots k_n} |_{\mathbf{x} \in domain_m} \tag{8}$$

Through the definition of regular triangle membership function (3), (4) and (7), an input is uniquely located in one sub-region and activates only the membership functions at the edge of that sub- region. That is, the overall fuzzy system

$$o(X) = o_m(X) \qquad X \in domian_m \tag{9}$$

We can see from (8) the model output of an input vector $\mathbf{x}$ within the *m*th sub-region o_m is determined by both the consequent parameters $y_m^{k_1 k_2 \dots k_n}$ and the membership functions $A_m^{k_1 k_2 \dots k_n}(\mathbf{x})$ of the *m*th sub-region in which they are located.

C. *Fuzzy term/rule creation and membership function updating*

In each sub-region splitting step, one new fuzzy term (membership function) on the selected splitting attribute is created and the two membership functions on the selected splitting attribute on the edge of the old selected sub-region are updated. We now explain how to create new fuzzy terms and rules and how to update related membership functions.

For a selected sub-region (with index of *m*), we assume *s* is the index for the selected attribute for splitting. This attribute selection criterion is described in subsection II.A.

If we represent the splitting point by $x_s^{splitting}$, then one new fuzzy term $x_s^{splitting}$ is added and its associated membership function is created as follows:

$$\mu_s^{splitting}(x_s) = \begin{cases} \dfrac{x_s - b_{m,s}}{x_i^{splitting} - a_{m,s}}, & if \quad x_s \in [b_{m,s}, x_s^{splitting}) \\ \dfrac{c_{m,s} - x_s}{c_{m,s} - x_i^{splitting}}, & if \quad x_s \in [x_s^{splitting}, c_{m,s}) \end{cases} . \tag{10}$$

After creating a new membership function, the two membership functions on selected splitting attribute x_s on the edge of the (old) selected (the *m*th) sub-region (represented by Equation (3) and (4)) are updated by (11) and (12).

$$\mu_{m,s}^0(x_s) = \frac{x_s^{splitting} - x_s}{x_s^{splitting} - b_{m,s}}. \tag{11}$$

$$\mu_{m,s}^1(x_s) = \frac{x_s - x_s^{splitting}}{c_{m,s} - x_s^{splitting}}. \tag{12}$$

Along with the creation of one fuzzy term (membership function) , 2^{n-1} new fuzzy rules are created as:

IF x_1 is $A_{m,1}^{k_1}$, and x_2 is $A_{m,2}^{k_2}$, … , and x_i is $x_i^{splitting}$, …, and x_n is $A_{m,n}^{k_n}$, THEN Y is". $y^{k_1 \dots k_{s-1} k_{s+1} \dots k_n}$ (13)

where k_i is 0 or 1 and $i \neq s$.

We now visualize this new fuzzy term (membership function)/rule creation process and the membership function updating process by a two-dimensional example in Fig.2: The white points are fuzzy rules; the black points are data points for splitting. $b_{m,i}$ and $c_{m,i}$ are the minimal and maximal value for the *i*th attribute for the *m*th sub-region. In Fig. 2 (a), assuming sub-region$_2$ is the selected sub-region with maximal average local error, $x_1^{splitting}$ is the selected point for splitting, where $\mathbf{x}^{splitting} = (x_1^{splitting}, x_2^{splitting})$ is a data instance in the dataset. Sub-region$_2$ is then split into Sub-region'$_2$ and Sub-region$_3$ by $x_1^{splitting}$. Two new fuzzy rules are created at follows as shown in Fig. 2 (b).

IF x_1 is $x_1^{splitting}$, x_2 is $b_{2,2}$, THEN Y is $y_{splitting}^0$ (14)

IF x_1 is $x_1^{splitting}$, x_2 is $c_{2,2}$, THEN Y is $y_{splitting}^1$ (15)

The membership functions for Sub-region$_2$ are as shown as in Fig. 2 (c) (before splitting) and Fig. 2 (d) (after splitting). In fig. 2 (d), the middle triangle is the membership function for the new created fuzzy term. The two half triangles on the edges are membership functions after updating.

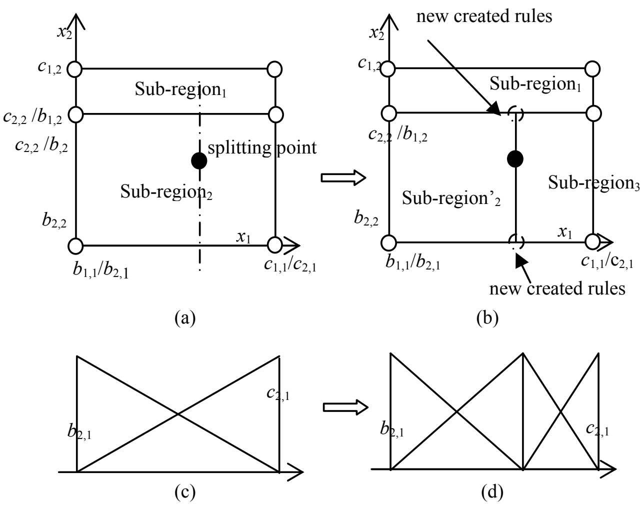

Fig.2. Creation of new fuzzy rules, new membership functions, and updating of existing fuzzy rules and membership functions.

D. Parameter initialization and optimization for new created fuzzy rules

The initial values of newly created parameters $y^{k_1k_2\dots k_{s-1}k_{s+1}\dots k_n}$ are set as follows.

$$y^{k_1k_2\dots k_{s-1}k_{s+1}\dots k_n} = \sum_{k_1'k_2'\dots k_n'} \left(\prod_{i=1}^{n} \mu_{m,i}^{k_i'}(x_i)\right)\times y_m^{k_1'k_2'\dots k_n'} \qquad (16)$$

where $x_i = b_{m,i}$, when $k_i = 0$; $x_i = c_{m,i}$, when $k_i = 1$, and $x_s = x_s^{splitting}$.

From equation (1) and (2), we have: $\mu_{m,i}^{k_i'}(x_i) = 0$, when $k_i \neq k_i'$; $\mu_{m,i}^{k_i'}(x_i) = 1$, when $k_i = k_i'$. If $b_s^{splitting} \le x_s^{splitting} \le c_s^{splitting}$, when $i = s$: $\mu_{m,s}^{k_s'}(x_s) = \dfrac{c_{m,s} - x_s^{splitting}}{c_{m,s} - b_{m,s}}$ if $k_s = k_s' = 0$; $\mu_{m,s}^{k_s'}(x_s) = \dfrac{x_s^{splitting} - c_{m,s}}{b_{m,s} - c_{m,s}}$ if $k_s = k_s' = 1$.

Based on (16), this implies

$$y^{k_1k_2\dots k_{s-1}k_{s+1}\dots k_n} = \left(\frac{c_{m,s} - x_s^{splitting}}{c_{m,s} - b_{m,s}}\right)\times y_m^{k_1k_2\dots k_n}\,\Big|_{k_s=0} + \left(\frac{x_s^{splitting} - b_{m,s}}{c_{m,s} - b_{m,s}}\right)\times y_m^{k_1k_2\dots k_n}\,\Big|_{k_s=1} \qquad (17)$$

By doing this, we obtain a good starting point for the following training.

III. ALGORITHM

We now generalize the algorithm for SELM.

1. The fuzzy system starts from one (sub-) region, the definition domain $\left(\times_{i=1}^{n}[b_i, c_i]\right)$. The fuzzy terms and rule base are initialized by the minimal value b_i and maximal value c_i (i = 1…n) for each attribute. The detailed formulas for these initial fuzzy rules are described as (1)
2. Compute the average local error, $error_j^{average}$ for each sub-region j and select one sub-region: the mth sub-region with maximal average local error: $m = \arg\min_{j=1}^{M}(error_j^{average})$, where M is the total number of current sub-regions, and $error_j^{average}$ is defined as:

$$error_j^{average} = \left(\frac{1}{N_j}\times\sum_{x^k \in Sub-region_j}(o^k - y^k)^2\right)^{\frac{1}{2}}, \qquad (18)$$

assuming that N_j data instances locate in the jth sub-region, o^k and y^k are the model output and actual output of $\boldsymbol{x}^k$.
3. Within the selected (the mth) sub-region, along each attribute i (i = 1…n) we find a data instance with $x_i^{splitting}$ which evenly separates the accumulated error for Sub-region m. Try to split Sub-region m into two sub-regions at each $x_i^{splitting}$ along the ith attribute (i = 1…n). The attribute $x_s^{splitting}$ with index s is selected if splitting sub-region m at $x_s^{splitting}$ results in the best global accuracy: $s = \arg\min_{i=1}^{n}(error_i^{global})$, where $error_i^{global}$ is the global error after splitting Sub-region m at $x_i^{splitting}$.
4. Split Sub-region m into two sub-regions and increase the total number of sub-regions by one: $M = M + 1$.
5. Based on the splitting in Step 4, create new fuzzy terms according to Equation (10), update the existing fuzzy terms according to Equation (11) and (12) and create new fuzzy rules are as (13). The values for parameters in the new created fuzzy rules are initialized using Equation (17).
6. Re-training parameters for global optimization after splitting.
7. If M is larger than the maximal limit of sub-regions or the global accuracy is better than the requirement,

then stop. Else continue the evloving process with Step 2.

Remark2: By using the above process to create new fuzzy terms and rules, the maximal number of fuzzy rules is $2^n + (MaxReg - 1) \times 2^{n-1} = (MaxReg + 1) \times 2^{n-1}$, and the maximal total number of fuzzy terms is $2^n + (MaxReg - 1)$, where *MaxReg* is the maximal number of sub-regions and n is the input dimension. With such a maximal allowed number of sub-regions *MaxReg*, the overall complexity of the final fuzzy model and the number of fuzzy terms allowed are controlled.

IV. SIMULATION

In this section, two benchmark system identification examples are given to illustrate the advantages of the proposed algorithm.

Simulation 1: Consider the following widely used two-dimensional non-linear function approximation problem [1], [26]-[28];

$$y = f(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1.5})^2 \quad x_1 \in [1,5],\ x_2 \in [1,5].$$

In this simulation, 50 training samples are randomly generated on the definition domain for both training and test. The performance comparison by different methods in MSE is reported in Table 1. The number of parameters in the third column is the number of parameters used in the consequent part. We can see from Table 1 that SELM obtains better accuracy than other training algorithms when using the same number of parameters.

TABLE 1. PERFORMANCE COMPARISON IN MSE WITH OTHER ALGORITHMS FOR SIMULATION 1

Method	Number of rules	Number of parameters	MSE
Sugeno and Yasukawa [28]	6	22	0.0790
Emami and Turksen [1]	8	24	0.0040
Tsekouras [27]	8	24	0.0042
Kim et all [26]	3	9	0.0550
SELM	8	8	0.0410
	10	10	0.0018
	12	12	0.0012
	14	14	0.0009

Simulation 2: In this simulation, we consider the non-linear dynamic system identification problem used by SAFIS [15], MRAN [27], RANEKF [28], Simp_eTS [11], eTS [10] and HA[29] as follows:

$$y(t+1) = \frac{y(t) \times y(t-1) \times [y(t) - 0.5]}{1 + y(t)^2 + y(t-1)^2} + u(t)$$

where $u(t) = \sin\left(\frac{2\pi t}{25}\right)$, $y(0) = 0$, and $y(1) = 0$.

5000 data were created for training, and 300 data were created for testing. The performance comparison of the proposed ICLA with SAFIS, MRAN, RANEKF, Simp_eTS, eTS and HA in RMSE is shown in Table 2. From Table 2, we see that the proposed SELM uses fewer fuzzy rules but achieves better accuracy than all the other methods. In additions, SELM products a more transparent fuzzy system due to using regular triangular membership functions.

TABLE 2 PERFORMANCE COMPARISON IN RSME WITH OTHER ALGORITHMS FOR SIMULATION 1

Method	Number of rules/neurons	RMSE for training	RMSE for testing
SAFIS [17]	17	0.0539	0.0221
MRAN [29]	22	0.0371	0.0271
RANEKF [30]	35	0.0273	0.0297
Simp_eTS [11]	22	0.0528	0.0225
eTS [10]	49	0.0292	0.0212
HA [31]	28	0.0182	0.0244
SELM	8	0.0318	0.0317
SELM	12	0.0039	0.0038
SELM	16	0.0013	6.7663E-4
SELM	20	0.0010	7.79243E-4

V. CONCLUSIONS

This paper proposes a Structure Evolved Learning Method (SELM) for the identification of Fuzzy Systems. Comparing with traditional learning methods of fuzzy systems, the motivation of SELM is to overcome the following main weaknesses:

- The even global grid-based partitions of fuzzy terms leads to the ineffectiveness in the sense that the rules and parameters increases exponentially with the number of input variables;
- The fixed definition of global grid-based partitions leads to the resulting fuzzy systems unintelligent and inflexible.

There are two key ideas behind the proposed SELM:

- The first is that the fuzzy partition of input spaces should follow the different smoothness of the approximated nonlinear function in the different regions. The more sparse (or coarse) partitions for the regions where the approximated function is smoother and the denser (or finer) partitions for the regions where the approximated function is less smooth.
- The second is that the incremental and evolving learning process is to follow the fastest routine to achieve the approximation error reduction.

SELM starts with the simplest fuzzy system, and develops its structure by adding fuzzy terms and rules based on the appropriately selected sub-region, input variable and point for splitting. SELM increases the number of sub-regions and fuzzy rules along the direction of the fastest approximation error reduction until it meets some pre-defined accuracy requirement. It can be proved that, by adding new fuzzy rules, the new fuzzy system can always achieve better accuracy than the old one. The reason is that we make use of a 'greedy' approach to find the sub-region, the splitting point and the attribute to create new fuzzy rules to obtain the least error in the next step by sub-region splitting. To verify the effectiveness and efficiency of SELM, two benchmark simulation examples have been carried out and compared with the various evolving learning methods such as eTS, DENFIS, and SAFIS. These simulation results show that SELM often obtains better accuracy by using the same or less number of fuzzy rules.

REFERENCES

[1] M. R. Emami, B.Turksen and A.A.Goldenberg, “Development of a systematic methodology of fuzzy modeling”, IEEE Trans. Fuzzy Syst., Vol.6, No.6, pp.341-361, 1998.

[2] Witold Pedrycz, “Logic-Based Fuzzy Neurocomputing with Unineurons”, IEEE Transactions on Fuzzy Systems, Vol. 14, No. 6, pp. 860-873, 2006.

[3] T. Takagi and M. Sugeno, “Fuzzy Identification of Systems and its Applications to Modeling and Control”, IEEE Transactions on Systems, Man and Cybernetics, Vol. 15, No. 1, pp. 116-132, 1985.

[4] L.-X. Wang, “Fuzzy Systems are Universal Approximators”, Proceeding 1st IEEE Conference on Fuzzy Systems, pp. 1163--1169, San Diego, CA, 1992

[5] L.-X. Wang and J. M. Mendel, “Fuzzy basis functions, universal approximation, and orthogonal least-squares learning,” IEEE Trans.Neural Networks, vol. 3, pp. 807-814, 1992.

[6] X.-J. Zeng and M.G. Singh, ``Approximation theory of fuzzy systems---SISO case," IEEE Trans. Fuzzy Systems, Vol. 2, pp. 162-176, 1994.

[7] X.-J. Zeng and M.G. Singh, ``Approximation theory of fuzzy systems---MIMO case," IEEE Trans. Fuzzy Systems, Vol. 3, pp. 219-235, 1995.

[8] P. Angelov, Evolving Rule-based Models: A Tool for Design of Flexible Adaptive Systems, Springer-Verlag, Heidelberg, New York, 2002.

[9] P. Angelov, D. Filev, N. Kasabov, and O. Cordon (Eds.), Evolving Fuzzy Systems, Proc. of the 2006 International Symposium on Evolving Fuzzy Systems EFS'06, IEEE Press, 2006.

[10] P.P. Angelov and D.P. Filev, “An approach to online identification of Takagi–Sugeno fuzzy models,” IEEE Trans. Systems Man Cybernet.—Part B, Vol. 34, pp. 484–498, 2004.

[11] P. Angelov, D. Filev, “Simpl_eTS: a simplified method for learning evolving Takagi–Sugeno fuzzy models,” in: The 14th IEEE Internat. Conf. on Fuzzy Systems, pp. 1068–1073, 2005.

[12] N. K. Kasabov, “DENFIS: Dynamic Evolving Neural-Fuzzy Inference System and its application for time-series prediction”, ”, IEEE Transactions On Fuzzy Systems. Vol. 10, pp.144-153, 2002.

[13] G. Leng, T. M. McGinnity and G. Prasad, "An Approach for On-line Extraction of Fuzzy Rules using a Sefl-organising Fuzzy Neural Network", Fuzzy Sets and Systems, Vol. 150, No. 2, pp. 211-243, 2005.

[14] G. Leng, G. Prasad, and T. M. McGinnity, (Dec 2004) "An on-line algorithm for creating self-organising fuzzy neural networks", Neural Networks, Vol. 17, No. 10, pp. 1477-1493, 2004.

[15] H. J. Rung, N. Sundararajan, G.B.Huang and P.Saratchandran, “Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction,” Fuzzy System and Set, Vol. 157, pp. 1260-1275, 2006.

[16] E. Jack., “Matroid and the greedy algorithm”, Math. Programming Vol. 1, pp. 127–113, 1971.

[17] K. Bernard, L. Lázsló, “Mathematical structures underlying greedy algorithms”, in Gecseg, F. (Ed.), Fundamentals of Computation Theory: Proceedings of the 1981 International FCT-Conference, Lecture Notes in Comput. Sci. Vol. 117, pp. 205–209, Berlin: Springer-Verlag, 1981.

[18] L.-X. Wang, A Course in Fuzzy Systems and Control. Upper Saddle River, NJ: Prentice-Hall, 1997.

[19] Y. S. Ding, H. Ying, and S. H. Shao, “Necessary Conditions on Minimal System Configuration for General MISO Mamdani Fuzzy Systems as Universal Approximators”, IEEE Trans. Syst. Man and Cybern. - B, Vol. 30, No. 6, pp. 857-864, 2000.

[20] W. Pedrycz, “Linguistic models as a framework of user-centric system modeling”, IEEE Trans. Syst. Man and Cybern. - A, Vol. 36, No. 4, pp. 727-745, 2006.

[21] E. Kim, M.Park, S.Ji and M.Park, “A new approach to fuzzy modeling”, IEEE Trans. Fuzzy Syst., Vol. 5, No.3, pp.328-337, 1997.

[22] G. E. Tsekouras, “On the use of the weighted fuzzy C-means in fuzzy modeling”, Int. J Advances in Eng. Software, Vol. 36, pp. 287-300, 2005.

[23] M. Sugeno and T. Yasukawa, “A fuzzy-logic based approach to qualitative modeling”, IEEE Trans. Fuzzy Syst., Vol. 1, No.1, pp.7-31, 1993.

[24] http://lib.stat.cmu.edu/datasets/.

[25] S. Wu and M. J. Er, “Dynamic fuzzy neural networks-a novel approach to function approximation”, IEEE Trans. Syst. Man and Cybern. - B, Vol. 30, pp. 358–364, 2000.

[26] S. Wu, M. J. Er, and Y. Gao, “A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks”, IEEE Trans. Fuzzy Syst., Vol. 9, 578–594, 2001.

[27] L.Yingwei, N. Sundararajan, and P. Saratchandran, “A sequential learning scheme for function approximation using minimal radial basis function (RBF) neural networks”, Neural Comput. Vol. 9, pp. 461–478, 1997.

[28] Kadirkamanathan and M. Niranjan, “A function estimation approach to sequential learning with neural networks”, Neural Comput. Vol. 5, pp. 954–975, 1993.

[29] L. Wang and J. Yen, “Extracting fuzzy rules for system modeling using a hybrid of genetic algorithm and Kalman filter”, Fuzzy Sets and Systems, Vol. 101, pp. 353–362, 1999.

[30] B. Fritze, “A growing neural gas network learns topologies,” Advances in Neural Information Processing Systems, Vol. 7, pp. 1-8, 1995.

[31] J. Platt, “A resource allocating network for function interpolation,” Neural Comutation, Vol. 3, pp. 213-225, 1991.

[32] D. Deng and N. Kasabov, “Evolving self-organizing maps for online learning, data analysis and modeling”, in Proceeding IJCNN 2000, Neural Networks Neural Computing, New Changes Perspectives New Millenium, Vol. VI, S. –I. C.L.Giles, M.Gori and V.Piuri, Eds. New York, pp.3-8, 2000.

[33] N. Kasabov, “Evolving fuzzy neural networks – algorithms, applications and biological motivation”, in Methodologies for conception design and application of soft computing, T. Yamakawa, and G. Matsumoto, Eds, Singapore: World Scientific, pp.217-274, 1998.

[34] S.J. Mousavi, K. Ponnambalam and F. Karray, “Inferring operating rules for reservoir operations using fuzzy regression and ANFIS,” Fuzzy Systems and Sets, Vol. 158, No. 10, pp. 1064-1062, 2007.

[35] H. Qin and S. X. Yang, “Adaptive neuro-fuzzy inference systems based approach to nonlinear noise cancellation for images”, Fuzzy Systems and Sets, Vol. 158, No. 10, pp. 1036-1063, 2007.

[36] Faa-Jeng Lin,and Po-Hung Shen, “Adaptive Fuzzy-Neural-Network Control DSP-Based Permanent Magnet Linear Synchronous Motor Servo Drive”, IEEE Transactions On Fuzzy Systems, Vol. 14, No. 4, pp. 481-495, 2006.

[37] H. Ishibuchi, K. Nozaki and H. Tanaka, “Efficient fuzzy partition of pattern space for classification problems”, Fuzzy Sets and Systems, Vol. 59, pp. 295-304, .1993.

[38] Gang Feng, “A Survey on Analysis and Design of Model-Based Fuzzy Control Systems”, IEEE Transactions On Fuzzy Systems, Vol. 14, No. 5, pp. 676-697, 2006.