A Fuzzy Clustering Algorithm for Generating Fuzzy Rules from Numerical Data

George E. Tsekouras
University of the Aegean,
Department of Cultural Technology and Communication
Faonos & Harilaou Trikoupi Str., 81100, Mytilene, Greece
Tel: +301-2251-0-36631, Fax: +301-2251-0-36609
gtsek@ct.aegean.gr

Gerasimos Pavlogeorgatos
University of the Aegean,
Department of Cultural Technology and Communication
Faonos & Harilaou Trikoupi Str., 81100, Mytilene, Greece
gpav@aegean.gr

Christos Kalloniatis
University of the Aegean,
Department of Cultural Technology and Communication
Faonos & Harilaou Trikoupi Str., 81100, Mytilene, Greece
ch.kalloniatis@ct.aegean.gr

Abstract - **This paper proposes a fuzzy clustering algorithm for fuzzy modeling. The algorithm is based on the assumption that, with an input fully matching with the premise part of a specific fuzzy rule, the corresponding output should completely participate in the consequent part. In order to accomplish this, certain conditions are satisfied. The application of the algorithm to two test cases, which have been considered as benchmarks in fuzzy modeling applications, showed that the produced models were of compact size, while the respective predictions were very accurate.**

I. INTRODUCTION

The most important task to accomplish a fuzzy model is to perform structure identification and parameter estimation. Assuming that the important linguistic variables have been chosen, structure identification is concerned with the determination of the number of rules [1,2]. On the other hand, parameter estimation refers to the calculation of the appropriate model parameter values, which provide an accurate system description. To perform structure identification and parameter estimation, many authors used heuristic processes [3] or adaptive schemes [4,5]. Sugeno and Yasukawa [2] used optimal fuzzy clustering to obtain clusters in the output space and then induced fuzzy clusters in the input space by cluster projections. Kim et al [6,7] used fuzzy clustering to produce ellipsoidal shaped clusters in the input-output space and then induced fuzzy rules by projecting these clusters on each dimension. According to Chen et al [8] the data space should be partitioned by considering local fields of the input data as well as linear relationships between input and output data. The method developed in [9] employed fuzzy clustering analysis to detect multidimensional reference fuzzy areas, where the number of rules is determined by reducing the model parameters, based on a system performance index. In [10], it is proposed an algorithm that yields clusters in the mapping space by incorporating the nature of the functional relationships into an objective function, while in [11] the structure identification is obtained via hyper-ellipsoidal clustering with simultaneous use of human intuition.
In this paper, a novel fuzzy clustering-based method is developed for fuzzy model identification. The proposed algorithm is based on decomposing the input space into a certain number of subspaces (clusters), each of which is assigned to a specific fuzzy rule. Then, the output space is relationally dismembered into the same number of clusters in such a way, that certain conditions are satisfied.

II. SUGENO-TYPE FUZZY MODEL

A major problem in fuzzy modeling is the reduction of the computational complexity, and since simplified fuzzy models utilize less parameters their usefulness is considerable. In [2], Sugeno and Yasukawa developed a simplified fuzzy model, which is described by the following fuzzy rules,

$$R^i : \textit{If} \;\; x_1 \; \textit{is} \; \Omega^i_1 \; \textit{and} \; x_2 \; \textit{is} \; \Omega^i_2 \; \textit{and} \; \ldots \; \textit{and} \; x_p \; \textit{is} \; \Omega^i_p$$
$$\textit{Then} \;\; y \; \textit{is} \; b^i \quad (1 \le i \le c) \tag{1}$$

where c is the total number of rules, p is the number of inputs, b_i are real numbers, and Ω^i_j are fuzzy sets, which have bell-typed shapes. Setting $x=[x_1, x_2, ..., x_p]^T$, the output of the model is determined as follows,

$$\tilde{y} = \frac{\sum_{i=1}^{c} w^i(x)\, b^i}{\sum_{i=1}^{c} w^i(x)} \tag{2}$$

$$\text{with} \;\; w^i(x) = \min_j \left\{\Omega^i_j(x_j)\right\}, \; 1 \le j \le p, \; 1 \le i \le c \tag{3}$$

From (1) we notice that the consequents are fuzzy singletons. The presence of fuzzy singletons in the consequent part of each rule requires a simpler identification procedure than other approaches [1].
To this end, we can expand the output of the model in (2), into the following fuzzy basis functions (FBFs) form,

$$\tilde{y} = \sum_{i=1}^{c} p^i(x)\, b^i \tag{4}$$

where the FBFs are given as:

$$p^i(x) = w^i(x) / \sum_{j=1}^{c} w^j(x) \qquad (1 \le i \le c) \tag{5}$$

III. THE PROPOSED METHOD

In this section the proposed algorithm is analyzed in details. The algorithm generates fuzzy rules based on a set of n input-output data pairs of the form $(x_k; y_k)$ $(1 \le k \le n)$. The basic design issues of the proposed method are described within the next subsections.

A. Estimation of the Model Premise Parameters

The estimation of the fuzzy set parameters in the premise part of each rule is based on producing a constrained fuzzy c-partition of the input space X by applying the well-known fuzzy c-means algorithm [12] on the input training data set. The fuzzy c-means minimizes the next objective function,

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m \| x_k - v_i \|^2 \tag{6}$$

under the following equality constraint,

$$\sum_{i=1}^{c} u_{ik} = 1 , \qquad \forall\ k \tag{7}$$

where n is the number of input data vectors, c is the number of clusters, u_{ik} is the membership degree of the k-th training vector to the i-th cluster, $m \in (1, \infty)$ is a factor to adjust the membership degree weighting effect, $x_k \in \Re^p$ are the input training data vectors, and $v_i \in \Re^p$ are the cluster centers. The cluster centers and the respective membership degrees that solve the above constrained optimization problem are respectively given by the following equations:

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} , \qquad 1 \le i \le c \tag{8}$$

and

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| x_k - v_i \|}{\| x_k - v_j \|} \right)^{\frac{2}{m-1}}} , \ 1 \le i \le c,\ 1 \le k \le n \tag{9}$$

Equations (8) and (9) constitute an iterative optimization procedure that produces a constrained fuzzy c-partition in the input space by dismembering it into c fuzzy clusters $X^1, X^2, \ldots, X^c$, where the i-th fuzzy cluster corresponds to the premise part of the i-th fuzzy rule. The membership function of each cluster is given in (9). In this equation the parameter m controls the fuzziness of the resulted fuzzy c-partition, and affects the overlapping degree between the multidimensional fuzzy clusters. More specifically, as this parameter increases, the overlapping degree also increases. On the other hand, as m approximates unity the resulted partition is a nearly crisp partition. As mentioned in section II, the fuzzy model in (1) uses bell-typed fuzzy sets in the premise part of each fuzzy rule, which are described as follows,

$$\Omega^i_j(x_{kj}) = \exp\left\{ -\left(\frac{x_{kj} - v^i_j}{\sigma^i_j} \right)^2 \right\} \tag{10}$$

The fuzzy set centers v^i_j $(1 \le j \le p,\ 1 \le i \le c)$, in the above equation, are obtained by projecting the final cluster centers v_i $(i=1, 2, \ldots, c)$ on each axe. On the other hand, to calculate the standard deviations (σ^i_j) we use the following fuzzy covariance matrix,

$$F_i = \frac{\sum_{k=1}^{n} (u_{ik})^m (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^{n} (u_{ik})^m} \tag{11}$$

Then, the standard deviation for each fuzzy set is given as follows,

$$\sigma^i_j = [Diag(F_i)]^{1/2} \ (1 \le i \le c,\ 1 \le j \le p) \tag{12}$$

B. Estimation of the Model Consequent Parameters

With the fuzzy c-partition of the input space introduced, the output space should be partitioned in a similar way. According to Chen et al [8] and Bezdek [12], this partition should be based on the following conditions:

<u>Condition 1</u>: If in the i-th fuzzy rule the vector x_k is the center element of the cluster X^i then the output y_k should satisfy the rule's consequence by concluding a truth degree equal to unity.

<u>Condition 2</u>: If in the i-th fuzzy rule the vector x_k is not the center element of the cluster X^i then the output y_k should satisfy the rule's consequence by concluding a truth degree less than unity.

The above conditions are referred to the matching degree between the premise and the consequent part of each fuzzy rule. In order to satisfy both conditions we proceed with the following analysis. The sum of the square errors (SSE) criterion is given as,

$$J_1 = \sum_{k=1}^{n} (y_k - \tilde{y}_k)^2 \tag{13}$$

Since the premise parameters are known, we can use (4) to modify (13) as follows,

$$J_1 = \sum_{k=1}^{n} (y_k - \sum_{i=1}^{c} p^i(\boldsymbol{x}_k)\, b^i)^2 \quad (14)$$

The objective is to minimize J_1 over the input-output training data set. One feasible way to accomplish this is to employ the least squares algorithm or any variant of it. However, in this case there is no guarantee that the conditions 1 and 2 will be satisfied. Therefore, we introduce the following theorem:

Theorem 1

If $m \to 1^+$ then the objective function J_1, given in (14) can be calculated as,

$$J_1 = \sum_{k=1}^{n} \sum_{i=1}^{c} (p^i(\boldsymbol{x}_k))^2 \, (y_k - b^i)^2 \quad (15)$$

Proof

For $1 \le i \le c$ and $1 \le k \le n$, from (9) we obtain that,

$$\lim_{m \to 1^+} \left\{ u_{ik} = \left[\sum_{j=1}^{c} \left(\frac{\| \boldsymbol{x}_k - \boldsymbol{v}_i \|}{\| \boldsymbol{x}_k - \boldsymbol{v}_j \|} \right)^{2/(m-1)} \right]^{-1} \right\} =$$

$$= \begin{cases} 1, & \text{if } \| \boldsymbol{x}_k - \boldsymbol{v}_i \| < \| \boldsymbol{x}_k - \boldsymbol{v}_j \| \; \forall i \ne j \\ 0, & \text{otherwise} \end{cases}$$

Thus, as $m \to 1^+$ the membership degrees in the input space are given as follows,

$$u_{ik} = \begin{cases} 1, & \text{if } \boldsymbol{x}_k \in \boldsymbol{X}^i \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

where $\boldsymbol{X} = \boldsymbol{X}^1 \cup \boldsymbol{X}^2 \cup ... \cup \boldsymbol{X}^c$ is a crisp partition of $\boldsymbol{X}$. But whenever u_{ik} is unity the vector $\boldsymbol{x}_k$ coincides with the center of the i-th cluster. Therefore, based on (3) and (5), the FBF that corresponds to that cluster is also equal to unity and all the rest FBFs are zero. Hence, as $m \to 1^+$ it holds that,

$$p^i(\boldsymbol{x}_k) = \begin{cases} 1, & \text{if } \boldsymbol{x}_k \in \boldsymbol{X}^i \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

Thus,

$$(y_k - \sum_{i=1}^{c} p^i(\boldsymbol{x}_k)\, b^i)^2 = (y_k - p^{l_i}(\boldsymbol{x}_k) b^{l_i})^2$$
$$= (p^{l_i}(\boldsymbol{x}_k))^2 \, (y_k - b^{l_i})^2 \quad (18)$$

where the index l_i corresponds to the crisp cluster $\boldsymbol{X}^{l_i}$ at which the $\boldsymbol{x}_k$ belongs to. Based on (17), and replacing (18) into the objective function in (14), we can easily derive (15). This completes the proof of theorem 1.

□

The next theorem provides the values of the consequent parameters that minimize the objective function in (15).

Theorem 2

For $1 \le i \le c$; If the values of the FBFs $p^i(\boldsymbol{x}_k)$ $(1 \le k \le n)$ are fixed, then the values of the consequent parameters b^i that minimize the objective function J_1, given in (16), are calculated as,

$$b^i = \frac{\sum_{k=1}^{n} (p^i(\boldsymbol{x}_k))^2 \, y_k}{\sum_{k=1}^{n} (p^i(\boldsymbol{x}_k))^2} \quad (19)$$

Proof

Setting the partial derivative $\partial J_1 / \partial b^i$ equal to zero, and solving with respect to b^i, we obtain (19). This completes the proof of theorem 2.

□

Summarizing, the key point to simultaneously satisfy the conditions 1 and 2 is to select a value for the parameter m close to unity. In this case, whenever the current training vector fully matches the premise part of a specific rule the firing degree of this rule, given in (3), equals unity. Correspondingly, based on (5) the respective FBF also equals unity. Thus, from (19) the respective consequent completely participates to this rule, meaning that the condition 1 is satisfied. Following the same analysis, we can easily figure out that the condition 2 is also satisfied.

C. Fine Tuning of the Model Parameters

In this section the model parameters, obtained in the previous sections, are further tuned by using the back-propagation algorithm developed in [13]. The task is to minimize the following error function,

$$e = \frac{1}{2} (\tilde{y}_k - y_k)^2 \quad (20)$$

To train v_j^i and σ_j^i we use the following learning rules,

$$v_j^i(t+1) = v_j^i(t) - \beta \left. \frac{\partial e}{\partial v_j^i} \right|_t \quad (21)$$

$$\sigma_j^i(t+1) = \sigma_j^i(t) - \beta \left. \frac{\partial e}{\partial \sigma_j^i} \right|_t \quad (22)$$

The partial derivative in (21) is calculated as,

$$\frac{\partial e}{\partial v_j^i} = \left(\tilde{y}_k - y_k\right)\left(b^i - \tilde{y}_k\right)\frac{p^i(x_k)}{w^i(x_k)}\frac{\partial [w^i(x_k)]}{\partial v_j^i} \quad (23)$$

Substituting (23) into (21), the learning rule for the v_j^i is given as follows,

$$v_j^i(t+1) = v_j^i(t) - \beta\left(\tilde{y}_k - y_k\right)\left(b^i - \tilde{y}_k\right)\frac{p^i(x_k)}{w^i(x_k)}\frac{\partial [w^i(x_k)]}{\partial v_j^i}$$

Following the same analysis, for the standard deviations we obtain the following learning rule:

$$\sigma_j^i(t+1) = \sigma_j^i(t) - \beta\left(\tilde{y}_k - y_k\right)\left(b^i - \tilde{y}_k\right)\frac{p^i(x_k)}{w^i(x_k)}\frac{\partial [w^i(x_k)]}{\partial \sigma_j^i}$$

Based on (3) and (10), the partial derivatives in the above equations can be easily derived. Relationally, for the consequent parameters the learning rule should be:

$$b^i(t+1) = b^i(t) - \beta\left(\tilde{y}_k - y_k\right)\ p^i(x_k)$$

The parameter β is the learning rate of the back-propagation training process.

D. The Proposed Algorithm

Based on the previous analysis, the proposed fuzzy modeling algorithm is now given as follows:
Suppose we are given n input-output data pairs of the form $(x_k; y_k)$ $(1 \le k \le n)$. Initially, select a value for the parameter m, which is close to unity. Set the number of rules c=2, and select a value for the parameters ε and β.

Step 1) Randomly, initialize the premise parameters v_i $(1 \le i \le c)$ and the respective consequent parameters b^i $(1 \le i \le c)$.

Step 2) For $k = 1, 2, ..., n$ and $i = 1, 2, ..., c$; Use (9) to calculate the membership degrees u_{ik}.

Step 3) For i=1, 2, …, c; Update the premise parameters v_i using (8).

Step 4) For $i = 1, 2, ..., c$ and $j = 1, 2, ..., p$; Determine the v_i^j by projecting the cluster centers v_i on each axe, and the standard deviations by using (12).

Step 5) For $k = 1, 2, ..., n$ and $i = 1, 2, ..., c$; Employ equation (5) to determine the FBFs.

Step 6) For i=1, 2, …, c; Calculate the consequent parameters using the (19).

Step 7) Calculate the distance $\| b - b_p \|$ where $b = [b^1, b^2, ..., b^c]^T$ and b_p the previous state of b. If $\| b - b_p \| \le \varepsilon$ then go to step 8; else go to step 2.

Step 8) Employ the back-propagation approach to fine the model parameters.

Step 9) If the performance of the model is OK then stop; Else set $c = c+1$ and go to step 1.

The above algorithm obtains a fuzzy partition of the input space, the properties of which are mapped into the output space. In this way, a uniform correspondence between these two spaces is created; where the fitting capabilities of the model are decided by the consequent parameters, with respect to the output space (step 7). Finally, it has to be noticed that the determination of the final number of clusters (i.e. fuzzy rules) is based on system's performance index.

IV. SIMULATION STUDY

A. Static Function Approximation

In this example we used the proposed algorithm for function approximation. The function is described by the following equation,

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \le x_1, x_2 \le 5 \quad (24)$$

The data set was taken from [2], and consists of fifty input-output data pairs, which were used to build a fuzzy model that can approximate the function. The design parameters of the algorithm were: m=1.05, $\varepsilon = 10^{-4}$ and β =5.15.

TABLE I
COMPARISON RESULTS FOR THE STATIC FUNCTION EXAMPLE

Model	No of rules	No of parameters	MSE
Sugeno-Yasukawa [2]	6	65	0.0790
Nozaki et al. [3]	25	125	0.0075
Kim et al. [6]	3	21	0.0197
Kim et al. [7]	3	21	0.0090
This model	6	30	0.0055

The implementation of the proposed algorithm gave a fuzzy model that consists of c=6 fuzzy rules. The resulted mean square error (MSE) was equal to 0.0055. In Table I the performance of this model is compared to the respective performances of other methods that can be found in the literature. From this table we can easily see that our model achieves the best performance with a relatively small number of parameters. Finally, the rule base of this model is shown in Table II.

B. Box-Jenkins System

In this subsection the proposed algorithm is applied to the well-known Box and Jenkins data set [14], which consists of 296 input-output measurements of a gas-furnace process, obtained using a sampling ratio of 9 s. At each sampling time k, the input $x(k)$ of this process is the gas flow rate and the output $y(k)$ is the output CO_2 concentration.

TABLE II
RULE BASE FOR THE STATIC FUNCTION EXAMPLE

i	A_1^i		A_2^i		b^i
	v_1^i	σ_1^i	v_2^i	σ_2^i	
1	2.741014	1.111055	4.430420	1.024052	0. 649984
2	4.033471	0.950046	2.631367	1.184880	1.858344
3	6.567444	27.04950	4.764024	2.902487	2.743225
4	4.867095	5.376120	-0.309028	1.133425	7.022224
5	4.849250	1.599936	4.408581	1.616570	-0.159688
6	0.606286	0. 490201	0.350242	14.77199	7.271129

The proposed method was used to design a fuzzy model for this process with 6 inputs: $x(k)$, $x(k\text{-}1)$, $x(k\text{-}2)$, $y(k\text{-}1)$, $y(k\text{-}2)$, $y(k\text{-}3)$ and one output: $y(k)$. In order to compare our method with other approaches, we used the first 148 data to build the model and the last 148 data as test data to validate its performance.

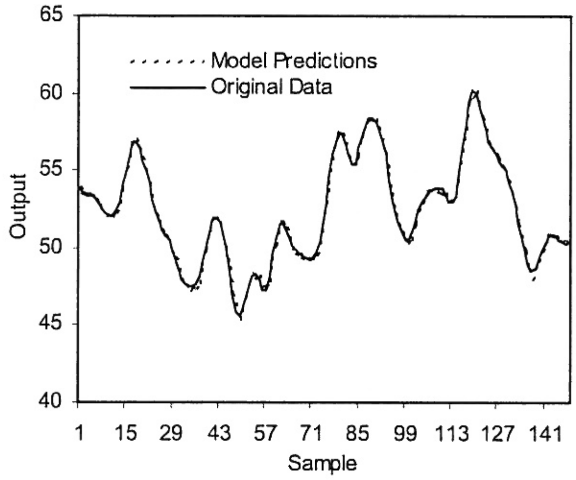

Fig. 1. Original and predicted values for the training data set of the Box and Jenkins system

The design parameters of the algorithm were: m=1.05, $\varepsilon = 10^{-4}$ and β =2.55. The final number of rules was c=7. The predicted and the original output values for the training data are given in Fig. 1, where the corresponding MSE is equal to 0.022. Fig. 2 shows the predicted and the actual values for the validation data for which, the MSE was equal to 0.236.

TABLE III
COMPARISON RESULTS FOR THE BOX-JENKINS EXAMPLE

Model	Number of rules	MSE Training data	MSE Test data
Lin-Cunnigham [15]	4	0.071	0.261
Kim et al. [16]	2	0.034	0.244
Our model	7	0.022	0.236

Finally, Table III compares the performances obtained here for the training and the test data to the respective performances of other methods that can be found in the literature. From this table we can see that our method obtains the best results in both the training and test data.

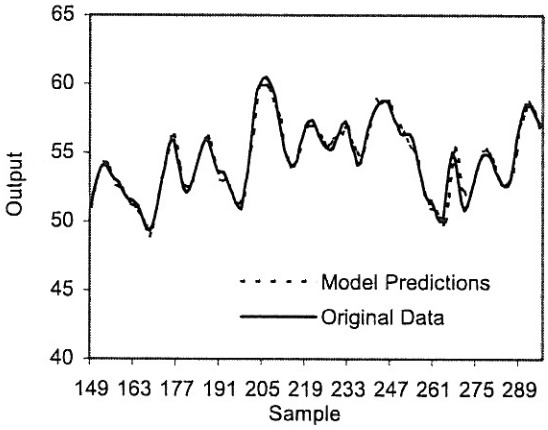

Fig. 2. Original and predicted values for the test data set of the Box and Jenkins system

V. CONCLUSIONS

In this paper we have proposed a novel method for building fuzzy models. The method is developed so that emphasis is given on both the accuracy and the size of the produced model. In order to achieve these targets, the method follows a number of steps, which are independent each other, so that the result of each step becomes the input of the next step. The basic design issue of the algorithm is that both the premise and the consequent parts appear an equal contribution to the firing degree of each rule. In order to accomplish this, certain conditions are taken into account. The application of the algorithm to two test cases shows that the algorithm is able to achieve a very efficient performance, while maintaining the size of the model within reasonable and acceptable levels.

VI. ACKNOWLEDGMENT

George E. Tsekouras gratefully acknowledges the financial support received by the State Scholarships Foundation of Greece.

VII. REFERENCES

[1] T. Takagi, and M. Sugeno, "Fuzzy identification of systems and its application to modeling and control", *IEEE Trans. Systems Man Cybern.*, Vol. 15, no. 1, 1985, pp. 116-132.

[2] M. Sugeno, M., and Yasukawa, T., "A fuzzy-logic-based approach to qualitative modeling", *IEEE Trans.Fuzzy Syst.*, Vol. 1, no. 2, 1993, pp 7-31.

[3] K. Nozaki, H. Ishibuchi, and H. Tanaka, "A simple but powerful method for generating fuzzy rules from numerical data", *Fuzzy Sets and Systems*, Vol. 86, 1997, pp 251-270.
[4] J.S.R Jang, "ANFIS: Adaptive-Network-based Fuzzy Inference Systems", *IEEE Trans Systems Man and Cybern.*, Vol. 23, no. 3, 1993, pp. 665-685.
[5] C.W. Xu, and Y.Z. Lu, "Fuzzy Model Identification and Self-Learning for Dynamic Systems", *IEEE Trans. Syst. Man & Cyber.*, Vol. 17, no. 4, 1987, pp. 683-689.
[6] Kim, E., Park, M., Ji, S., and Park, M., "A new approach to fuzzy modeling", *IEEE Trans Fuzzy Systems*, Vol. 5, no. 3, 1997, pp. 328-337.
[7] Kim, E., Park, M., Kim, S., and Park, M., "A transformed input-domain approach to fuzzy modeling", *IEEE Trans Fuzzy Systems*, Vol. 6, no. 4, 1998, pp. 596-604.
[8] J. Chen, Y. Xi and Z. Zhang, "A clustering algorithm for fuzzy model identification", *Fuzzy Sets and Systems*, Vol. 98, 1998, pp 319-329.
[9] A. Kroll, "Identification of functional fuzzy models using multidimensional reference fuzzy sets", *Fuzzy Sets and Systems*, Vol. 80, 1996, pp 149-158.
[10] K. Hirota, and W. Pedrycz, "Directional fuzzy clustering and its application to fuzzy modeling", *Fuzzy Sets and Systems*, Vol. 80, 1996, pp 315-326.
[11] Y. Nakamori, and M. Ryoke, "Identification of fuzzy prediction models through hyper-ellipsoidal clustering", *IEEE Trans. Syst. Man and Cybern.*, Vol. 24, no. 8, 1994, pp.1153-1173.
[12] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. Y., 1981.
[13] L.X. Wang, J.M. Mendel, "Back-propagation fuzzy systems as nonlinear dynamic system identifiers", in *Proc. IEEE International Conf. Fuzzy Systems*, San Diego, 1992 pp. 1409-1418.
[14] G.E.Box, and G.M. Jenkins, *Time series Analysis, forecasting and control*, San Francisco, CA: Holden Day, 1970.
[15] Y. Lin, and G.A. Cunningham, "A new approach to fuzzy-neural modeling", *IEEE Trans. Fuzzy Systems*, Vol. 3, 1995, pp 190-197.
[16] E. Kim, H. Lee, M. Park, and M. Park, "A simply identified Sugeno-type fuzzy model via double clustering", *Information Sciences,* Vol. 110, 1998, pp. 25-39.