A Fuzzy Clustering-Based Algorithm for Fuzzy Modeling

GEORGE E. TSEKOURAS*, CHRISTOS KALLONIATIS, EVAGELIA KAVAKLI, AND THOMAS MAVROFIDES
Laboratory of Image Processing and Multimedia Applications,
Department of Cultural Technology and Communication, University of the Aegean,
Faonos & Harilaou Trikoupi Str., 81100, Mytilene, Greece
*Tel: +301-2251-0-36631, Fax: +301-2251-0-36609

Abstract: Fuzzy rules have a simple structure within a multidimensional vector space and they are produced by dismembering this space into fuzzy subspaces. The most efficient way to produce fuzzy partitions in a vector space is the use of fuzzy clustering analysis. This paper proposes a fuzzy clustering-based algorithm, which generates fuzzy rules from a set of input-output data. 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 derived. The application of the algorithm to a test case, which has been considered as a benchmark in fuzzy modeling applications, shows that the produced models are of compact size, while their performances are very efficient.

Key-Words: Fuzzy clustering; Fuzzy modeling; System identification; Model parameter estimation; Fuzzy partition; Crisp partition.

1. Introduction

The basic issue in fuzzy modeling is the identification procedure that is employed. Fuzzy model identification consists of structure identification and parameter estimation. Structure identification is directly related to the determination of the appropriate number of rules [1,2]. On the other hand, parameter estimation concerns the calculation of the appropriate model parameter values that provide an accurate system description. Structure identification and parameter estimation are usually carried out via a training procedure. So far, a wide spectrum of methods has been proposed to train fuzzy systems. Many of these methods use heuristic approaches [3], self-learning and adaptive schemes [4,5], or gradient descent algorithms [6].

One of the most efficient fuzzy modeling procedures is the utilization of fuzzy clustering analysis. Fuzzy clustering provides a certain advantage over other approaches, since the partition of the input (or the product) space is obtained as a direct result [7]. The method developed in [8] use 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 [9] it is proposed an algorithm that yields clusters in the mapping space by incorporating the nature of the functional relationships into an objective function. In [10] the structure identification is obtained via hyper-ellipsoidal clustering with simultaneous use of human intuition, while in [11] the hyper-ellipsoidal subspaces have been replaced by spherical fuzzy areas where the membership functions determine the structure of the rules.

In this paper, a novel fuzzy clustering-based method is proposed for system 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 have to be satisfied.

2. The Proposed Algorithm

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

2.1 Partitioning the Input Space by Using Fuzzy Clustering Analysis

A major issue in fuzzy modeling is the reduction of the computational complexity, and since simplified fuzzy models use less parameters their usefulness is considerable. In our approach we adopt the simplified fuzzy model introduced in [3], which is described by the following fuzzy rules,

$$R^i : If \ x_1 \ is \ X_1^i \ and \ x_2 \ is \ X_2^i \ and \ ... \ and \ x_p \ is \ X_p^i \quad Then \ \ y \ is \ b^i \quad (1 \le i \le c) \tag{1}$$

where p is the number of inputs, c in the number of rules, X_j^i $(1 \le i \le c; 1 \le j \le p)$ are fuzzy sets, and b^i are real numbers. The above fuzzy model can approximate any nonlinear function to arbitrary accuracy on a compact set [3].

Based on fuzzy reasoning, it is evident that even when an input linguistic variable is not appearing in the premise part of one rule, a fuzzy set can be assigned to it with a firing degree of unity. This remark suggests a uniform structure of the premise part of the rule base, where all the input linguistic variables participate in all of the fuzzy rules. In addition to that, a more credible fuzzy rule base can be created by assuming that the output variable participates in each rule with a normal fuzzy set, meaning that there is at least one element belonging to the fuzzy set with membership degree of unity. By considering that c fuzzy rules are needed to describe a nonlinear system, the uniform structure of the premise part of the rule base enables us to partition the input space $\boldsymbol{X}$ into c fuzzy subspaces $\boldsymbol{X}^1, \boldsymbol{X}^2, \ldots, \boldsymbol{X}^c$. Each of these subspaces is assigned to only one fuzzy rule. Therefore, the fuzzy rule in (1) can be modified as,

$$R^i : \textit{If} \;\; \boldsymbol{x} \;\; \textit{is} \;\; \boldsymbol{X}^i \;\; \textit{Then} \;\; y \;\; \textit{is} \;\; b^i \;\; (1 \le i \le c) \tag{2}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_p]^T$ and $\boldsymbol{X}^i \subset \boldsymbol{X}$ with $\boldsymbol{X}^i = \{X_1^i, X_2^i, \ldots, X_p^i\}$. Since our model is described by fuzzy rules of the form (2), we can produce a constrained fuzzy c-partition of the input space $\boldsymbol{X}$ by applying the well-known fuzzy c-means algorithm on the input training data set. The fuzzy c-means is based on the minimization of the following objective function [12],

$$J_m = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m \| \boldsymbol{x}_k - \boldsymbol{v}_i \|^2 \tag{3}$$

under the next equality constraint,

$$\sum_{i=1}^{c} u_{ik} = 1 , \qquad \forall\, k \tag{4}$$

where n is the number of training 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, $\boldsymbol{x}_k \in \Re^p$ are the input training data vectors, and $\boldsymbol{v}_i \in \Re^p$ 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 [12],

$$\boldsymbol{v}_i = \frac{\sum_{k=1}^{n} (u_{ik})^m \boldsymbol{x}_k}{\sum_{k=1}^{n} (u_{ik})^m} , \qquad 1 \le i \le c \tag{5}$$

and

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| \boldsymbol{x}_k - \boldsymbol{v}_i \|}{\| \boldsymbol{x}_k - \boldsymbol{v}_j \|} \right)^{\frac{2}{m-1}}} , \; 1 \le i \le c, \; 1 \le k \le n \tag{6}$$

The eqs (7) and (8) constitute an iterative optimization procedure.

By applying the above minimization procedure to the input training data vectors, these vectors are classified into c fuzzy clusters, where the i-th cluster corresponds to the i-th fuzzy subspace. Since a single fuzzy subspace corresponds to a specific fuzzy rule, the number of clusters coincides with the total number of fuzzy rules. Eventually, the membership degree of the training vector $\boldsymbol{x}_k$ to the i-th fuzzy subspace $\boldsymbol{X}^i$ is the membership degree u_{ik}. In the rest of the paper, the word «fuzzy cluster» will replace the word «fuzzy subspace», meaning that these two words are referred to the same concept.

2.2 Model Parameter Initialization

Based on the analysis presented in the previous section, the premise part of each rule consists of multidimensional fuzzy clusters, the membership functions of which are given in eq (6). The form of this equation indicates that the membership function is interpreted as the membership degree that is assigned to the input vector $\boldsymbol{x}_k$ by the center element $\boldsymbol{v}_i$ of the cluster $\boldsymbol{X}^i$. Thus, the width of the cluster $\boldsymbol{X}^i$ is not included in the membership function and therefore, it is not taken into account in the parameter estimation either. Another important issue is the presence of the parameter m. This parameter controls the fuzziness of the resulted partition and thus, it affects the overlapping degree between the multidimensional fuzzy clusters. More specifically, as this parameter increases, the overlapping degree also increases. This means that for a specific value of the parameter m the overlapping degree between the clusters is known, and therefore, the locations of the cluster centers indicate the distances between the clusters. Thus, the premise parameter identification only concerns the estimation of the appropriate cluster centers. To this end, the premise parameter estimation is based on iteratively applying the eqs (5) and (6) to the input training data, where the resulted cluster centers provide the fuzzy rule premise parameters and the respective membership degrees provide the firing degrees of the fuzzy rules. Therefore, the output of the fuzzy model can be calculated as,

$$\tilde{y}_k = \sum_{i=1}^{c} u_{ik}\, b^i \Big/ \sum_{i=1}^{c} u_{ik} \qquad (1 \le k \le n)$$

Taking into account the eq (4), the above equation is modified as follows,

$$\tilde{y}_k = \sum_{i=1}^{c} u_{ik}\, b^i \quad (1 \le k \le n) \tag{7}$$

With the fuzzy c-partition of the input space introduced, the output space should be partitioned in a similar way. Moreover, this partition should be based on the following conditions [11, 12],

<u>Condition 1</u>: If in the i-th fuzzy rule the vector $\boldsymbol{x}_k$ is the center element of the cluster $\boldsymbol{X}^i$ 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 $\boldsymbol{x}_k$ is not the center element of the cluster $\boldsymbol{X}^i$ 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. One feasible way to satisfy these two conditions is to perform clustering analysis in the product space (i.e. the input-output space) and then induce fuzzy sets by projecting the resulted clusters on each dimension. Such kinds of approaches are investigated in [7,8,9]. However, the main drawback of these approaches is that the consequent parameters are not calculated by the use of an optimizing criterion. In order to solve this problem, we introduce the following condition,

<u>Condition 3</u>: The consequent parameters should be estimated by minimizing the sum of the square errors (SSE) criterion.

The above condition has to be satisfied together with the conditions 1 and 2. The SSE criterion is given as,

$$J_1 = \sum_{k=1}^{n} (y_k - \tilde{y}_k)^2 \tag{8}$$

With the premise parameters known, the respective consequent parameters can be obtained by minimizing the J_1 over the n input-output data pairs. Using eq. (7), eq. (8) gives that,

$$J_1 = \sum_{k=1}^{n} \Big(y_k - \sum_{i=1}^{c} u_{ik}\, b^i\Big)^2 \tag{9}$$

One feasible way to minimize J_1 is to employ the well-known least squares algorithm. However, the utilization of this algorithm does not guarantee that the conditions 1 and 2 will be satisfied. Therefore, we introduce the following procedure.

Theorem 1

If $m \to 1^+$ then the objective function J_1, given in eq. (9) can be calculated as,

$$J_1 = \sum_{k=1}^{n}\sum_{i=1}^{c} (u_{ik})^2\, (y_k - b^i)^2 \tag{10}$$

Proof

For $1 \le i \le c$ and $1 \le k \le n$, from eq. (6) we obtain that,

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

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

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

$$u_{ik} = \begin{cases} 1, & if\ \ \boldsymbol{x}_k \in \boldsymbol{X}^i \\ 0, & otherwise \end{cases} \tag{11}$$

where $\boldsymbol{X} = \boldsymbol{X}^1 \cup \boldsymbol{X}^2 \cup \ldots \cup \boldsymbol{X}^c$ is a crisp partition of $\boldsymbol{X}$.

From eq. (11) it follows that, there are k_1 input data vectors that belong to the cluster $\boldsymbol{X}^1$, k_2 data vectors that belong to the cluster $\boldsymbol{X}^2$, …., and k_c data vectors that belong to the cluster $\boldsymbol{X}^c$, such that,

$$k_1 + k_2 + \ldots + k_c = n \tag{12}$$

Therefore, the following relation holds,

$$\begin{aligned} \Big(y_k - \sum_{i=1}^{c} u_{ik}\, b^i\Big)^2 &= (y_k - u_{l_i k}\, b^{l_i})^2 \\ &= (u_{l_i k})^2\, (y_k - b^{l_i})^2 \end{aligned} \tag{13}$$

where the index l_i corresponds to the crisp cluster $\boldsymbol{X}^{l_i}$ at which the $\boldsymbol{x}_k$ belongs to. Based on eqs (11), (12), and (13) the objective function in (9) can be modified as follows,

$$J_1 = \sum_{k=1}^{k_1} (u_{1k})^2 \, (y_k - b^1)^2 + \sum_{k=1}^{k_2} (u_{2k})^2 \, (y_k - b^2)^2 + ... + \sum_{k=1}^{k_c} (u_{ck})^2 \, (y_k - b^c)^2$$

which means that,

$$J_1 = \sum_{i=1}^{c} \sum_{k=1}^{k_i} (u_{ik})^2 \, (y_k - b^i)^2 \qquad (14)$$

The i-th crisp cluster $\boldsymbol{X}^i$ includes k_i training vectors and therefore the rest (n-k_i) training data vectors are assigned by $\boldsymbol{X}^i$ membership degrees equal to zero. Therefore, the following relation holds,

$$\sum_{k=1}^{k_i} (u_{ik})^2 \, (y_k - b^i)^2 =$$

$$= \sum_{k=1}^{k_i} (u_{ik})^2 \, (y_k - b^i)^2 + \sum_{k=1}^{n-k_i} (u_{ik})^2 \, (y_k - b^i)^2$$

$$= \sum_{k=1}^{n} (u_{ik})^2 \, (y_k - b^i)^2 \qquad (15)$$

Replacing eq. (15) into eq. (14) we can derive the eq. (10). This completes the proof of theorem 1.

□

The next theorem provides the values of the consequent parameters that minimize the objective function in (10).

Theorem 2

For $1 \le i \le c$; If the values of the membership degrees u_{ik} $(1 \le k \le n)$ are fixed, then the values of the consequent parameters b^i that minimize the objective function J_1, given in eq. (10), are calculated as,

$$b^i = \frac{\sum_{k=1}^{n} (u_{ik})^2 \, y_k}{\sum_{k=1}^{n} (u_{ik})^2} \qquad (16)$$

Proof

Setting the partial derivative $\partial J_1 / \partial b^i$ equal to zero, and solving with respect to b^i, we can easily derive the eq. (16). This completes the proof of theorem 2.

□

Summarizing, the premise parameters are calculated by the eq (5) and the consequent parameters by the eq. (16).

2.3 Fine Tuning of the Model Parameters

In this section the model parameters, obtained in the previous step, are further tuned by using a gradient descent approach. The objective function that is used for this purpose is given as,

$$J_2 = \frac{1}{2n} \sum_{k=1}^{n} (y_k - \tilde{y}_k)^2$$

By substituting eq. (7) into the above function we obtain that,

$$J_2 = \frac{1}{2n} \sum_{k=1}^{n} (y_k - \sum_{i=1}^{c} u_{ik} \, b^i)^2 \qquad (17)$$

In order to minimize J_2 the premise parameters have to be adjusted as follows,

$$\Delta \boldsymbol{v}_i = \frac{\beta_1}{n} \sum_{k=1}^{n} \left[\left(y_k - \tilde{y}_k \right)^2 b^i \, \frac{\partial u_{ik}}{\partial \boldsymbol{v}_i} \right] \qquad (18)$$

where, based on (6), the partial derivative is given as,

$$\frac{\partial u_{ik}}{\partial \boldsymbol{v}_i} = \frac{2}{(m-1)(\boldsymbol{x}_k - \boldsymbol{v}_i)} \, \frac{\sum_{\substack{j=1 \\ j \ne i}}^{c} \left(\frac{\| \boldsymbol{x}_k - \boldsymbol{v}_i \|}{\| \boldsymbol{x}_k - \boldsymbol{v}_j \|} \right)^{\frac{2}{m-1}}}{\left[\sum_{j=1}^{c} \left(\frac{\| \boldsymbol{x}_k - \boldsymbol{v}_i \|}{\| \boldsymbol{x}_k - \boldsymbol{v}_j \|} \right)^{\frac{2}{m-1}} \right]^2} \qquad (19)$$

Relationally, the learning rule for the consequent parameters is as follows,

$$\Delta b^i = \frac{\beta_2}{n} \sum_{k=1}^{n} \left[(y_k - \tilde{y}_k) \, u_{ik} \right] \qquad (20)$$

In the above equations, the parameters β_1 and β_2 are the gradient descent learning parameters.

2.4 The Identification Algorithm

Based on the previous analysis, the proposed fuzzy modeling algorithm is now given as follows.

The Proposed Algorithm

Suppose we are given n input-output data pairs of the form $(\boldsymbol{x}_k; y_k)$ $(1 \le k \le n)$. Initially select a small value for the parameter m, which is close to unity. Set the number of rules c=2, and select a value for the terminal condition parameters ε_1 and ε_2.

Step 1). Randomly, initialize the premise parameters $\boldsymbol{v}_i$ $(1 \le i \le c)$ and the consequent parameters b^i $(1 \le i \le c)$.

Step 2). For $k = 1, 2, ..., n$ and $i = 1, 2, ..., c$; Use the eq (6) to calculate the membership degrees u_{ik}.

Step 3). For i=1, 2, …, c; Update the premise parameters $\boldsymbol{v}_i$ using the eq. (5).

Step 4). For i=1, 2, …, c; Calculate the consequent parameters using the eq. (16).

Step 5). Calculate the distance $\| \boldsymbol{b} - \boldsymbol{b}_{\mathrm{p}} \|$ where $\boldsymbol{b} = [b^1, b^2, ..., b^c]^T$ and $\boldsymbol{b}_{\mathrm{p}}$ the previous state of $\boldsymbol{b}$.

Step 6). If $\| \boldsymbol{b} - \boldsymbol{b}_{\mathrm{p}} \| \le \varepsilon_1$ then go to step 7; else go to step 2.

Step 7). Employ the gradient descent approach to minimize J_2, where the model parameter learning rules are given by the eqs (18) and (20).

Step 8). Calculate the performance index of the model: $PI = \sum_{k=1}^{n} (y_k - \tilde{y}_k)^2 \Big/ n$. If $PI \le \varepsilon_2$ then stop; Else set c= c+1 and go to step 1.

□

The final result of the above iterative optimization is that, with an input fully matching with one of rules' premise part, the corresponding output satisfies the consequence completely, meaning that the truth degree of each fuzzy rule is equal to unity. Thus, the eq. (7) can be used for inference of the output from a specific input data vector.

3. Simulation Study

In this subsection the proposed algorithm is applied to the well-known Box and Jenkins data set [2], 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. The proposed method was used to design a fuzzy model for this process

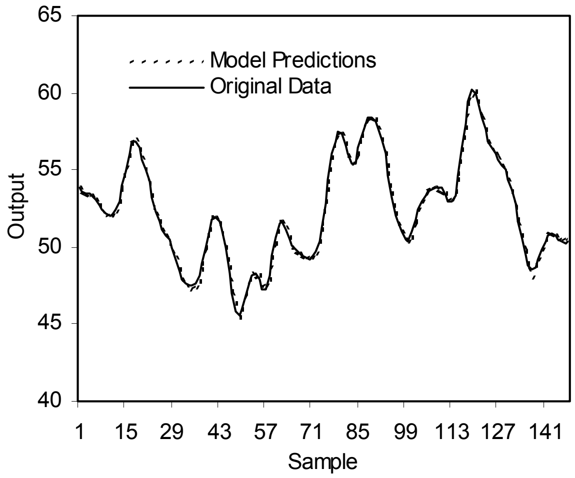

Figure 1: Original and predicted values for the training data set of the Box and Jenkins system (Case 1).

with 6 inputs: $x(k)$, $x(k$-1), $x(k$-2), $y(k$-1), $y(k$-2), $y(k$-3) and one output: $y(k)$. In order to compare our method with other approaches, we performed two experimental cases namely, case 1 and case 2.

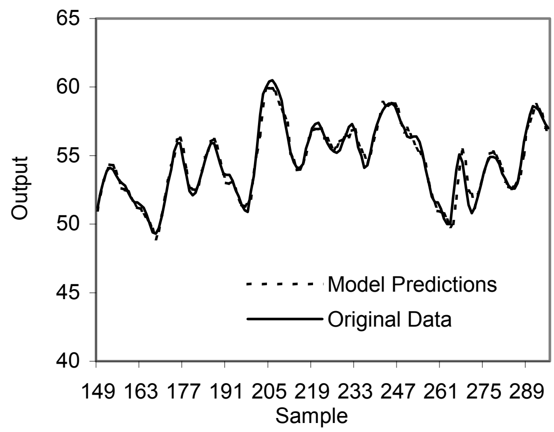

Figure 2: Original and predicted values for the test data set of the Box and Jenkins system (Case 1).

In case 1 we used the first 148 input-output data as training data to build the fuzzy model and the last 148 as test data to validate its performance. The terminal conditions were selected as ε_1=10^{-4} and ε_2=10^{-2}, and the learning rates for the gradient descent method were: β_1 = β_2= 0.55. The final number of rules was equal to c=3.The predicted and the original output values for the training data are given in Fig.1, where the corresponding Mean Square Error (MSE) was equal to 0.045. Fig. 2 shows the predicted and the actual values for the validation data for which, the MSE was equal to 0.251. The MSEs, which were obtained for the same case study by the method developed in [14] were 0.071 for that training data, and 0.261 for the test data, meaning that our model performs better than this method.

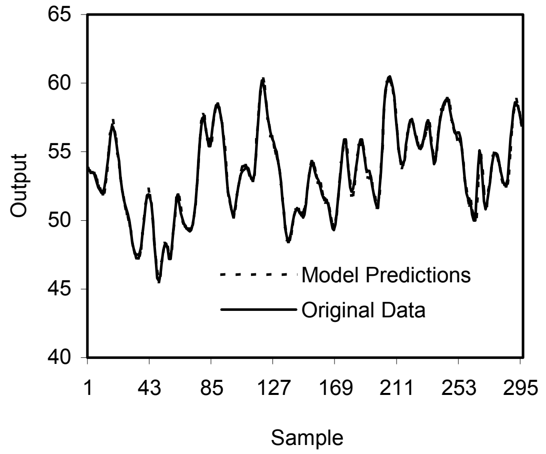

Figure 3: Original and predicted values for the Box and Jenkins system (Case 2).

In case 2 we used all the data set to build the fuzzy model and to validate its performance. The terminal conditions were selected as $\varepsilon_1=10^{-4}$ and $\varepsilon_2=10^{-2}$, and the learning rates for the gradient descent method were: $\beta_1 = \beta_2= 0.3$. The final number of rules was equal to $c=4$. Fig.3 depicts the predicted and the actual values, where the MSE was equal to 0.1398. Table 4 compares the performance of the produced fuzzy model to other models that can be found in the literature. From this table we can easily notice that our model achieves the best performance.

Table 4: Comparison results for the Box-Jenkins example (Case 2)

Model	Number of rules	MSE
Box and Jenkins[13]	---	0.2020
Chen et al. [11]	3	0.2678
Sugenoand Yasukawa [3]	6	0.1900
Xu and Lu [5]	25	0.3280
Gomez-Skarmeta et al. [8]	2	0.1570
Kroll [8]	2	0.1495
Our Model	4	0.1398

4. Conclusions

In this paper we have proposed a novel method to train 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 a test case shows that the algorithm is able to achieve a very efficient performance, while keeping the size of the model within reasonable and acceptable levels.

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 (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 (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 (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 (4), 1987, pp. 683-689.

[6] 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.

[7] A.G. Skarmeta, M. Delgado, and M. Vila, About the use of fuzzy clustering techniques for fuzzy model identification, *Fuzzy Sets and Systems*, Vol. 106, 1999, pp 179-188.

[8] A. Kroll, Identification of functional fuzzy models using multidimensional reference fuzzy sets, *Fuzzy Sets and Systems*, Vol. 80, 1996, pp 149-158.

[9] K. Hirota, and W. Pedrycz, Directional fuzzy clustering and its application to fuzzy modeling, *Fuzzy Sets and Systems*, Vol. 80, 1996, pp 315-326.

[10] Y. Nakamori, and M. Ryoke, Identification of fuzzy prediction models through hyper-ellipsoidal clustering, *IEEE Trans. Syst. Man and Cybern.*, Vol. 24(8), 1994, pp.1153-1173.

[11] J. Chen, Y. Xi and Z. Zhang, A clustering algorithm for fuzzy model identification, *Fuzzy Sets and Systems*, Vol. 98, 1998, pp 319-329.

[12] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. Y., 1981.

[13] G.E.Box, and G.M. Jenkins, *Time series Analysis, forecasting and control*, San Francisco, CA: Holden Day, 1970.

[14] Y. Lin, and G.A. Cunningham, A new approach to fuzzy-neural modeling, *IEEE Trans. Fuzzy Systems*, Vol. 3, 1995, pp 190-197.