Fuzzy Modeling via Optimal Fuzzy Clustering

CHRISTOS KALLONIATIS, THOMAS MAVROFIDES, GEORGE E. TSEKOURAS*

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: This paper introduces a new method for fuzzy modeling based on set of input-output data pairs. The method consists of a sequence of steps aiming towards developing a Sugeno-type fuzzy model of optimal structure. In the first place, the algorithm uses the fuzzy *c*-means to classify all the input training data vectors into a predefined number of clusters. The centers of these clusters are further processed by using optimal fuzzy clustering, which is based on the weighted fuzzy *c*-means algorithm. The resulted optimal fuzzy partition defines the number of fuzzy rules and provides an initial estimation for the system parameters, which are further tuned using the gradient-descend algorithm. The proposed method is successfully applied to a time series prediction problem, where its performance is compared to the performances of other methods found in the literature.

Key-Words: Fuzzy modeling; Fuzzy clustering; Optimal fuzzy partition; System identification; Parameter estimation; Orthogonal least squares; Time series prediction.

1. Introduction

Fuzzy model identification consists of structure identification and parameter estimation [1,2]. Usually, the structure identification is concerned with the determination of the appropriate number of fuzzy rules. Fuzzy model identification is carried out via a training process. One of the most common approaches to train fuzzy systems is the use of fuzzy clustering analysis. Many authors [3,4] use clustering analysis to detect clusters in the product space and then induce fuzzy partitions in the input space by cluster projections. Kroll [5] employs clustering analysis to generate multidimensional reference fuzzy areas and then obtains a fuzzy model, with reduced number of parameters, by using a specific performance measure. Hirota et al [6] developed an algorithm that produces clusters in the mapping space, where the nature of the functional relationships is incorporated into a minimization procedure of an efficient objective function. Also, the use of hyper-ellipsoidal clusters [7] has been proven to be very efficient.

In most of the above clustering approaches, the algorithms are randomly initialized many times in order to obtain a desired local minimum. However, random initialization may give sub-optimal results. Therefore, the development of clustering algorithms that depend less on initialization is of main importance. To accomplish this, Wang [8] used nearest neighbor clustering. However, the final fuzzy partition that is produced by this approach may not correspond to the optimal fuzzy partition, which is directly related to the real data structure. One feasible way to accommodate nearest neighbor search is to perform cluster merging [9]. However, neither this approach will guarantee the optimal fuzzy partition. To solve this problem Linkens and Chen [10] developed a two-level algorithm, where in the first level a competitive scheme produces an initial number of clusters, while in the second level the centers of these clusters are further clustered using the fuzzy *c*-means method. The application of fuzzy *c*-means assumes that the cluster centers, taken from the first level, are of equal significance. However, this may not be true since two different initial clusters may contain different number of elements and therefore their variances are different, meaning that their significances with respect to the final partition are different. Thus, the optimal fuzzy partition of the original data set may still not be obtained.

In this paper we propose a two-level optimal fuzzy-clustering scheme to solve the aforementioned problems. In the first place, the algorithm utilizes the fuzzy *c*-means method to produce an initial fuzzy partition of the input space by dismembering it into a predefined number of clusters. Then, the resulted cluster centers are further clustered by means of optimal fuzzy clustering. This is achieved by using the weighted fuzzy *c*-means [11] since, as mentioned previously, the significance of each cluster center should depend on the variance of that cluster. Finally, in order to carry out the optimal fuzzy clustering we employ the Xie-Beni validity index.

2. Sugeno-Type Fuzzy Model

In [2], Sugeno and Yasukawa developed a fuzzy model, which is described by the following fuzzy rules,

$$R^i : If \;\; x_1 \;\; is \;\; \Omega_1^i \;\; and \;\; x_2 \;\; is \;\; \Omega_2^i \;\; and \; ... \; and \;\; x_p \;\; is \;\; \Omega_p^i$$
$$Then \;\; y \;\; is \;\; B^i \qquad (1 \le i \le c) \tag{1}$$

where c is the total number of rules, p is the number of inputs, and B^i and Ω_j^i are fuzzy sets. In this paper, the fuzzy sets Ω_j^i $(1 \le i \le c;\, 1 \le j \le p)$ are of bell-typed shapes. Setting $\boldsymbol{x} = [x_1, x_2, ..., x_p]^T$, the output of the model is determined as follows,

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

where $\omega^i(\boldsymbol{x}) = \min_j \left\{ \Omega_j^i(x_j) \right\},\; 1 \le j \le p,\; 1 \le i \le c$ (3)

and $b^i = \int B^i(y)\, y\, dy \Big/ \int B^i(y)\, dy,\;\; 1 \le i \le c$ (4)

From eq. (4) we notice that the inference mechanism "understands" the consequents as fuzzy singletons. The presence of fuzzy singletons in the consequent part of each rule requires a simpler identification procedure than other approaches [1].

3. The Proposed Method

This section describes the proposed algorithm. The algorithm consists of 4 steps, which are briefly described next.

In the first step, the algorithm applies the fuzzy *c*-means to the available input training data vectors, by employing a significantly large number of clusters.

In the second step, the determination of the appropriate number of fuzzy rules takes place. More specifically, the cluster centers obtained in the previous step are considered as a new data set, which is further clustered by means of optimal fuzzy clustering. To accomplish this, the algorithm uses the weighted fuzzy *c*-means [11], where the weight of significance for each cluster center is taken into account. Then, the optimal fuzzy clustering is carried out by using the Xie-Beni validity index. The implementation of the optimal fuzzy clustering requires a large number of clusters in the first step, since otherwise the resulted optimal fuzzy partition will not be a credible one.

In the third step, the fuzzy covariance matrix and the final cluster centers are used to initialize the fuzzy set parameters in the premise part of each fuzzy rule. Moreover, an initial estimation of the consequent parameters is obtained by a using the orthogonal least squares algorithm [12].

Finally, in the last step, the premise and consequent model parameters are fine tuned by employing a gradient descent based approach.

The four steps of the algorithm are analyzed in details within the following subsections.

3.1 Initialization of the Input Space Fuzzy Partition (Step 1)

One of the most widely used fuzzy clustering algorithms is the well-known fuzzy *c*-means method. Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\}$ be a set of N unlabeled feature data vectors with $\boldsymbol{x}_k \in \Re^p$ $(1 \le k \le N)$, and n $(2 \le n \le N)$ be the number of fuzzy clusters defined in *X*. Given that the membership degree of the data vector $\boldsymbol{x}_k$ to the *i*-th cluster $(1 \le i \le n)$ is denoted as $\mu_{ik} = \mu_i(\boldsymbol{x}_k)$, the objective of the fuzzy *c*-means is to minimize the following function,

$$J_m = \sum_{k=1}^{N} \sum_{i=1}^{n} (\mu_{ik})^m \, \| \boldsymbol{x}_k - \boldsymbol{\upsilon}_i \|^2 \tag{5}$$

under the next equality constraint,

$$\sum_{i=1}^{n} \mu_{ik} = 1 \,, \qquad \forall \; k \tag{6}$$

where $m \in (1, \infty)$ is a factor to adjust the membership degree weighting effect, and $\boldsymbol{\upsilon}_i \in \Re^p$ the center of the *i*-th cluster. The cluster centers and the respective membership functions that solve the above constrained optimization problem are given by the following equations [11],

$$\boldsymbol{\upsilon}_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m \, \boldsymbol{x}_k}{\sum_{k=1}^{N} (\mu_{ik})^m} \;, \qquad 1 \le i \le n \tag{7}$$

and

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| \boldsymbol{x}_k - \boldsymbol{\upsilon}_i \|}{\| \boldsymbol{x}_k - \boldsymbol{\upsilon}_j \|} \right)^{\frac{2}{m-1}}} \qquad 1 \le k \le N,\; 1 \le i \le n \tag{8}$$

Equations (7) and (8) constitute an iterative optimization procedure.

3.2 Optimization of the Input Space Fuzzy Partition (Step 2)

This step performs the structure identification of the fuzzy model, where the appropriate number of rules is determined. More specifically, the cluster centers $\boldsymbol{\upsilon}_k \ (1 \le k \le n)$, generated in the previous step, will be further clustered by using optimal fuzzy clustering. Since these data have been determined by applying the fuzzy c-means, each of them should weight differently with respect to the final fuzzy partition. The reason is that one of these data, say $\boldsymbol{\upsilon}_{k_0}$, may correspond to a cluster that contains many more elements from the original data set than the rest of the clusters, and therefore its contribution to the final partition should be greater. In order to solve this problem, we use the weighted fuzzy c-means method [11]. In our case, the objective function for the weighted fuzzy c-means is given as,

$$J_a = \sum_{k=1}^{n}\sum_{i=1}^{c} w_k \, (u_{ik})^a \, \| \boldsymbol{\upsilon}_k - \mathbf{v}_i \|^2 \qquad (9)$$

where n is the number of the cluster centers generated in the previous step, c is the number of clusters that constitute the final fuzzy partition (i.e. the number of rules), $a \in (1, \infty)$ is a parameter to adjust the membership degree weighting effect, w_k is the weight of significance that is assigned to the data vector $\boldsymbol{\upsilon}_k$, $\mathbf{v}_i \ (1 \le i \le c)$ are the cluster centers of the final fuzzy partition, and u_{ik} is the membership degree of $\boldsymbol{\upsilon}_k$ to the $\mathbf{v}_i$. Since the k-th vector $\boldsymbol{\upsilon}_k$ corresponds to a cluster of the original data set, we introduce the following procedure to determine the respective weight,

$$\rho_k = \sum_{j=1}^{N} \mu_{kj} \quad (1 \le k \le n) \qquad (10)$$

where μ_{kj} is given in eq. (8). Then, the weight w_k is defined as follows,

$$w_k = \frac{\rho_k}{\sum_{i=1}^{n} \rho_i} \qquad (11)$$

The optimization problem is to minimize the objective function in (9) under the following equality constraint,

$$\sum_{i=1}^{c} u_{ik} = 1 \quad \forall k \qquad (12)$$

The final cluster centers and the respective membership functions that solve the above constrained optimization problem are given by the following equations [11],

$$\mathbf{v}_i = \frac{\sum_{k=1}^{n} w_k \, (u_{ik})^a \, \boldsymbol{\upsilon}_k}{\sum_{k=1}^{n} w_k \, (u_{ik})^a} \qquad (1 \le i \le c) \qquad (13)$$

and

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| \boldsymbol{\upsilon}_k - \mathbf{v}_i \|}{\| \boldsymbol{\upsilon}_k - \mathbf{v}_j \|} \right)^{\frac{2}{a-1}}} \qquad (1 \le i \le c, 1 \le k \le n) \qquad (14)$$

Equations (13) and (14) constitute an iterative optimization procedure, which is known as the weighted fuzzy c-means algorithm [11].

Optimal fuzzy clustering concerns the estimation of the number of clusters that yields a fuzzy partition, in which data belonging to the same cluster are as similar as possible. Despite the fact that both the fuzzy c-means and the weighted fuzzy c-means are able to detect similarities within a data set, they cannot perform optimal fuzzy clustering because they require an a-priori knowledge of the number of clusters. Therefore, a convenient way to exhibit optimal fuzzy clustering is to develop a reliable index (i.e. function) to accompany the objective function in (9), in order to obtain the best possible clustering results. In this section, we use the well-known Xie-Beni index [13], which for the weighted fuzzy c-means is calculated as follows,

$$S_{XB} = \frac{\sum_{k=1}^{n}\sum_{i=1}^{c} w_k \, (u_{ik})^a \, \| \boldsymbol{\upsilon}_k - \mathbf{v}_i \|^2}{n \min_{i \ne j} \{ \| \mathbf{v}_i - \mathbf{v}_j \|^2 \}} \qquad (15)$$

The optimum number of clusters using the above index is the one that corresponds to its lowest value.

3.3 Model Parameter Initialization (Step 3)

As mentioned in section 2, the fuzzy model in (1) uses bell-typed fuzzy sets in the premise part of each fuzzy rule, which are described as follows,

$$\Omega_j^i(x_{kj}) = \exp\left\{ -\left(\frac{x_{kj} - v_j^i}{\sigma_j^i} \right)^2 \right\} \qquad (16)$$

The fuzzy set centers v_j^i $(1 \le j \le p, 1 \le i \le c)$ are obtained by projecting the final cluster centers $\boldsymbol{v}_i$ (i=1, 2, …, c) on each axe. In order to calculate the standard deviations (σ_j^i), we use the fuzzy covariance matrix,

$$\boldsymbol{F}_i = \frac{\sum_{k=1}^{n} w_k \, (u_{ik})^{\alpha} \, (\boldsymbol{v}^k - \boldsymbol{v}_i) \, (\boldsymbol{v}^k - \boldsymbol{v}_i)^T}{\sum_{k=1}^{n} w_k \, (u_{ik})^{\alpha}} \tag{17}$$

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

$$\sigma_j^i = [Diag(\boldsymbol{F}_i)]^{1/2} \;\; (1 \le i \le c, \, 1 \le j \le p) \tag{18}$$

After the premise parameters of relation (1) have been initialized, we can expand the output of the model, given in eq. (2), into the following fuzzy basis functions (FBFs) form,

$$\tilde{y}_k = \sum_{i=1}^{c} p^i(\boldsymbol{x}_k) \, b^i \tag{19}$$

where $$p^i(\boldsymbol{x}_k) = \omega^i(\boldsymbol{x}_k) \Big/ \sum_{i=1}^{c} \omega^i(\boldsymbol{x}_k) \tag{20}$$

For the N input-output data pairs of the form ($\boldsymbol{x}_k$; y_k) (k=1, 2, …, N), the consequent parameters are obtained via an optimization procedure. In order to do this, we employ the following regression model,

$$\boldsymbol{y} = \boldsymbol{p}\boldsymbol{b} + \boldsymbol{e} \tag{21}$$

with,
$\boldsymbol{y} = [y_1 \;\; y_2 \;\; .... \;\; y_N]^T$, $\boldsymbol{b} = [b^1 \;\; b^2 \;\; .... \;\; b^c]^T$, $\boldsymbol{e} = [e_1 \;\; e_2 \;\; .... \;\; e_N]^T$, and $\boldsymbol{p} = [\boldsymbol{p}_1 \;\; \boldsymbol{p}_2 \;\; .... \;\; \boldsymbol{p}_c]$, with $\boldsymbol{p}_i = [p^i(\boldsymbol{x}_1) \;\; p^i(\boldsymbol{x}_2) \;\; .... \;\; p^i(\boldsymbol{x}_N)]^T$. In the above relations $\boldsymbol{e}$ is the error vector. In order to solve (21) we use the orthogonal least squares (OLS) approach proposed in [12], which is described by the next algorithm,

Orthogonal Least Squares

Step1) For i=1 set: $\boldsymbol{\beta}_1 = \boldsymbol{p}_1$ and

$$g_1 = (\boldsymbol{\beta}_1)^T \boldsymbol{y} \Big/ \left((\boldsymbol{\beta}_1)^T \boldsymbol{\beta}_1 \right)$$

Step2) For $2 \le i \le c$ set: $\gamma_{ji} = (\boldsymbol{\beta}_j)^T \boldsymbol{p}_j \Big/ \left((\boldsymbol{\beta}_j)^T \boldsymbol{\beta}_j \right)$ with $1 \le j < i$ and also,

$$\boldsymbol{\beta}_i = \boldsymbol{p}_i \quad \text{and} \quad g_i = (\boldsymbol{\beta}_i)^T \boldsymbol{y} \Big/ \left((\boldsymbol{\beta}_i)^T \boldsymbol{\beta}_i \right)$$

Step3) Solve the triangular system: $\boldsymbol{A} \; \boldsymbol{b} = \boldsymbol{g}$ where,

$$\boldsymbol{A} = \begin{bmatrix} 1 & \gamma_{12} & \gamma_{13} & \cdots\cdots & . & \gamma_{1c} \\ 0 & 1 & \gamma_{23} & \cdots\cdots & . & \gamma_{2c} \\ 0 & 0 & 0 & \cdots\cdots & . & \gamma_{3c} \\ . & . & . & \cdots\cdots & . & . \\ . & . & . & \cdots\cdots & . & . \\ . & . & . & \cdots\cdots & . & . \\ 0 & 0 & 0 & \cdots\cdots & 1 & \gamma_{(c-1)c} \\ 0 & 0 & 0 & \cdots\cdots & 0 & 1 \end{bmatrix}$$

and $\boldsymbol{g} = [g_1 \;\; g_2 \;\; .... \;\; g_c]^T$

3.4 Parameter Tuning (Step 4)

In this step the system parameters obtained in the previous section are further tuned using the well-known gradient descent algorithm. The objective function that is used for this purpose is,

$$J_1 = \frac{1}{2N} \sum_{k=1}^{N} (y_k - \tilde{y}_k)^2 \tag{22}$$

where $\tilde{y}_k$ is given in eq. (19). By applying the gradient descent method to minimize J_1, the premise parameters of the fuzzy model can be precisely adjusted by using the following learning rules,

$$\Delta v_j^i = \frac{\beta_1}{N} \sum_{k=1}^{N} \Bigg[(y_k - \tilde{y}_k) \, (b^i - \tilde{y}_k) \cdot \frac{p^i(\boldsymbol{x}_k)}{\omega^i(\boldsymbol{x}_k)} \frac{\partial[\omega^i(\boldsymbol{x}_k)]}{\partial v_j^i} \Bigg] \tag{23}$$

and

$$\Delta \sigma_j^i = \frac{\beta_2}{N} \sum_{k=1}^{N} \Bigg[(y_k - \tilde{y}_k)(b^i - \tilde{y}_k) \cdot \frac{p^i(\boldsymbol{x}_k)}{\omega^i(\boldsymbol{x}_k)} \frac{\partial[\omega^i(\boldsymbol{x}_k)]}{\partial \sigma_j^i} \Bigg] \tag{24}$$

The partial derivatives in the above equations can be easily calculated by using the eqs (3) and (16). Relationally, for the consequent parameters the learning formula should be,

$$\Delta b^i = \frac{\beta_3}{N} \sum_{k=1}^{N} \left[(y_k - \tilde{y}_k) \, p^i(\boldsymbol{x}_k) \right] \qquad \forall i \tag{25}$$

In the above equations the parameters β_1, β_2 and β_3 are the learning rates for the gradient descent.

4. Simulation Experiments

In this subsection we use the proposed modeling scheme to predict the Mackey-Glass time series, which is generated by the following time-delay differential equation,

$$\frac{dx(t)}{dt} = \frac{0.2\, x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \qquad (26)$$

When the parameter τ is large enough, the system appears a chaotic behavior. In our simulations we set τ=17 and generated a sample of 1000 points, which are drawn in Fig. 1. The first 500 points were used to build the fuzzy model, and the last 500 points as test data to validate its performance. The proposed algorithm was applied to build a model with 4 inputs: $x(k-18)$, $x(k-12)$, $x(k-6)$ and $x(k)$, while the output was the point $x(k+6)$. For the step 1, we used n=40, m=2, and the terminal condition for the fuzzy c-means was ε_1=0.00001.

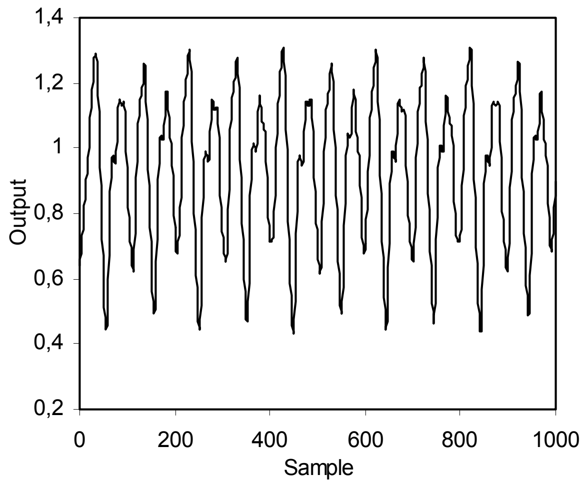

Figure 1: A set of 1000 data points of the Mackey-Glass time-series system.

For the step 2 we used a=2, and the terminal condition for the weighted fuzzy c-means was set equal to ε_2=0.00001. Different initializations were used, which gave similar results. In order to apply the validity index, the parameter c was set to take values within the interval [2, $c_{\max}$], where $c_{\max} \le 2\sqrt{n}$. Thus, for n=40 the $c_{\max}$ was equal to $c_{\max}$=12. Fig. 2 depicts the results obtained by applying the Xie-Beni validity index, showing that the optimal number of clusters was c_{opt}=9. Therefore, the final number of rules was equal to c=9. The learning rates for the gradient descent were selected as β_1=β_2=β_3=0.55. Fig. 3 depicts the model outputs and the actual outputs of the system for the test data, where the Root Mean Square Error (RMSE) is equal to 0.00624.

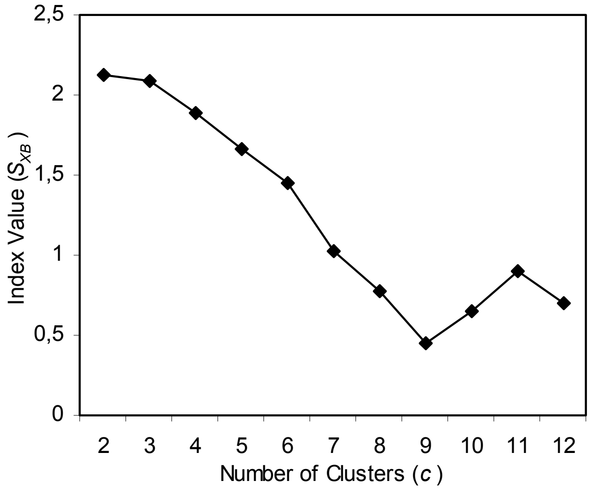

Figure 2: Values of the proposed validity index for different numbers of clusters for the Mackey-Glass system.

Finally, table 6 compares the RMSE obtained by the proposed algorithm to other results found in the literature. From this table we can easily notice that our model gives the best results and therefore, it can be used as a reliable tool for time series prediction.

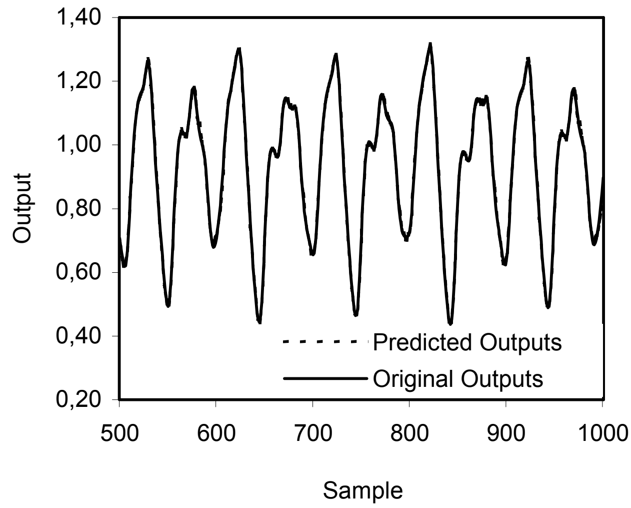

Figure 3: Original and predicted values for the test data set of the Mackey-Glass system.

5. Discussion and Conclusions

In this paper we have proposed a novel method for fuzzy modeling, which is based on optimal fuzzy clustering. The algorithm starts by producing a number of fuzzy clusters within the multidimensional input space, so that all the training input data vectors belong to at least one cluster. The number of clusters is then reduced to an optimum, which is determined using the weighted fuzzy c-means. The optimum number of clusters corresponds to the total number of rules, since each fuzzy cluster defines a single fuzzy rule.

Table 1: Comparison results for the test data of the Mackey-Glass system.

Model	RMSE
Crowder [14]	0.02
De Souza [15]	0.0065
Kim [16]	0.0264
Lenng et al. [17]	0.0215
Wang and Mendel [18]	0.01
Our model	0.0062

The parameters of the premise membership functions are calculated by projecting the clusters on each axe, while the respective consequent parameters are obtained by applying the orthogonal least squares algorithm on the input-output training data set. Finally, the fuzzy model parameters are fine tuned using the gradient descent approach.

The novelty of the contribution lies on the fact that the above method extends the fuzzy clustering-based fuzzy modeling approaches by employing the weighted fuzzy *c*-means. As it was shown from the experiments, the main advantage of using weight factors for the clusters, which are produced in the first step, is that the whole approach becomes significantly less sensitive to initialization. Therefore, compared to other approaches, the final fuzzy partition produced by our model corresponds to a local minimum that is closer to the global minimum.

The main characteristics of the method are summarized as follows: Firstly, the method yields the optimal fuzzy partition, which is directly related to the real data structure. Secondly, the whole approach is one pass through the training data set and thus, it needs less computational demands than other methods do. Thirdly, the resulted fuzzy model utilizes a small number of rules, while at the same time the prediction performance is very accurate.

In view of the above, we can conclude that the proposed algorithm is a very attractive method that can be efficiently implemented to a wide area of practical applications such as: modeling of chemical processes, function approximation, and time series prediction.

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, and T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, *IEEE Trans. Fuzzy Systems*, Vol 1 (1), 1993, pp 7-31.

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

[4] J. A. Dickerson, and B. Kosko, Fuzzy function approximation with ellipsoidal rules, *IEEE Trans Syst. Man and Cyber.*, Vol. 26 (4), 1996, pp.542-560.

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

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

[7] 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.

[8] L.X. Wang, *Adaptive fuzzy systems and control*, Prentice Hall, 1994.

[9] U. Kaymak, and M. Setnes, Fuzzy clustering with volume prototypes and adaptive cluster merging, *IEEE Trans. Fuzzy Systems*, Vol. 10 (6), 2002, pp. 705-712.

[10] D. A. Linkens, and M. Y. Chen, Input selection and partition validation for fuzzy modeling using neural network, *Fuzzy Sets and Systems*, Vol. 107, 1999, pp. 299-308.

[11] J. C Bezdek, *Pattern recognition with fuzzy objective function algorithms*, Plenum, N.Y., 1981.

[12] L. X. Wang, and J. M. Mendel, Fuzzy basis functions, universal approximation, and orthogonal least squares learning, *IEEE Trans. Neural Networks*, Vol. 3 (5), 1992, pp. 807-814.

[13] L. X. Xie, and G. Beni, A validity measure for fuzzy clustering, *IEEE Trans. PAMI*, Vol. 13, 1991, pp. 841-847.

[14] R. S. Crowder, *Predicting the Mackey-Glass time series with cascade-correlation learning*, in: D. Touretzky, G. Hinton, and T. Senjowski (Eds) Proceedings of the 1990 Connectionist Models, Summer School, Carnegie Mellon University, 1990, pp 117-123.

[15] F. J. de Souza, M. M. R. Vellasco, and M. A. C. Pacheco, Hierarchical neuro-fuzzy quadtree models, *Fuzzy Sets and Systems*, Vol. 130, 2002, pp. 189-205.

[16] D. Kim, Improving the fuzzy system performance by fuzzy system ensemble, *Fuzzy Sets and Systems*, Vol. 98, 1998, pp. 43-56.

[17] C. S. Lenng, and L. W. Chan, Dual extended Kalman filtering in recurrent neural networks, *Neural Networks*, Vol. 16, 2003, pp. 223-239.

[18] L. X. Wang, and J. M. Mendel, Generating fuzzy rules from numerical data with applications, *IEEE Trans Systems Man Cybern.*, Vol. 22 (6), 1992, pp. 1414-1427.