A New Learning Algorithm for Neuro-Fuzzy Modeling Using Self-Constructed Clustering

Sung-Suk Kim*, Keun-Chang Kwak**, Sung-Soo Kim*, Jeong-Woong Ryu*

*Department of Electronic Engineering, Chungbuk National University
(Tel : +82-43-261-2419; E-mail: powerkim@chungbuk.ac.kr)
**PostDoc, University of Alberta, Canada
(E-mail: kwak@ece.ualberta.ca)

Abstract: In this paper, we proposed a learning algorithm for the neuro-fuzzy modeling using a learning rule to adapt clustering. The proposed algorithm includes the data partition, assigning the rule into the process of partition, and optimizing the parameters using predetermined threshold value in self-constructing algorithm. In order to improve the clustering, the learning method of neuro-fuzzy model is extended and the learning scheme has been modified such that the learning of overall model is extended based on the error-derivative learning. The effect of the proposed method is presented using simulation compare with previous ones.

Keywords: Clustering, Neuro-Fuzzy Modeling, TSK fuzzy model, Self-Constructed Clustering, and System Identification

1. INTRODUCTION

Basically, the modeling of neuro-fuzzy system are consisted of structure identification that determined the structure of model and parameters identification that optimized parameters.[1-3]. In the structure identification[2], the type of model and number of fuzzy rules are determined. and in the parameter identification, learning algorithm is optimized by adjusting the parameters of models. The choice the number and the type of the membership function is important factor that determines the parameters at identification of structure, such that the system contains a particular characteristics[4][5].

General method uses the process of partitioning the input space in grid type, then uses all surface of input dimension to connect relevant neuro-fuzzy rules. In this case[2], the grid partition method has an advantage that covered all input space, but in the case of increasing the dimension of input space and number of membership function, neuro-fuzzy rules are increased exponentially by connecting premise part and rule. These problems are weakened by employing the clustering method. The problem of grid partition is removed using clustering method which assigns the neuro-fuzzy rules[2][4][5] to the space with data not to the empty space and using a new approach that counts only the number of associated number of function without counting on the dimension of input.

The clustering methods determined the number of clusters without information on the shape of density function. The second method estimates the parameters involved in the shape of density function with the knowledge on the number of clusters. In the first case, the data density is generally used estimate the number of clusters using density function, such that the process of optimization may missed. Also, when the number of clusters is determined, the algorithm performs well when correctly choose the number of clusters and the number of patterns. However, when the number of clusters fails to represent the whole data, the algorithm’s performance cannot be guaranteed.

In the modeling of neuro-fuzzy system, clustering methods are used for the initial parameters estimation of the model[4][6]. In this case, clustering algorithm and neuro-fuzzy modeling is used in sequence.

The input space partition optimization by clustering becomes ineffective when the parameters involved in the neuro-fuzzy modeling are changed. Basic learning method of general neuro-fuzzy modeling is based on updating the membership function of premise part using the error-derivative of model output.

In this paper, we propose a new learning algorithm that simultaneously inference a number of cluster and optimizes these parameters using the membership function of Takagi-Sugeno-Kang fuzzy model in the process of learning. Also, we extend the supervised learning to the unsupervised learning in learning by the clustering using feedback of error in every step of the overall model, we extend the clustering based learning from the error–derivative based ones, such that self-construct the neuro-fuzzy rules optimize these parameters. Finally, we applied the proposed method for benchmark problems and obtained better results than previous ones.

2. PROPOSED CLUSTERING ALGORITHM

2.1 Self-Constructing Clustering

Basic concepts of clustering are grouping data that includes the same kind clusters with high similarity and exclusive others[2][7][8]. For this purpose, clustering algorithm uses a function of similarity measure. Proposed method is use generalized functions[2][4] as following

\[ r_j = \exp \left( -\frac{1}{2} (x_i - c_j)^T \Sigma_j^{-1} (x_i - c_j) \right) \]  

where \( c_j \) and \( \Sigma_j \) are mean and covariance of \( j \) th cluster, number of data is represented by \( i = 1, 2, ..., N \) and number of clustering is represented by \( j = 1, 2, ..., c \). The proposed
function has the identity when data is the same, and decrease exponentially when away from the mean. The data groups which are not affected by the proposed method still effective measuring the similarities between groups through cluster parameter estimation. So, we use the constrained magnitude of similarity using the predetermined threshold value $\zeta_i$.

$$r_{ij} = \begin{cases} r_{ij}, & \text{if } r_{ij} > \zeta_i \\ 0, & \text{otherwise} \end{cases}$$

(2)

In this case, for varying the $\zeta_i$, the results of clustering changes such that when $\zeta_i$ is large, it eliminates information on the current clusters and when $\zeta_i$ is small, $\zeta_i$ has no effect in the algorithm. So, in order to solve this problem, we modified the function of similarity measure from the equation (1) to equation (3) such as

$$r_{ij} = \exp \left( -\frac{1}{2} (x_i - c_j) (\zeta_i \cdot \Sigma_j)^{-1} (x_i - c_j) \right)$$

(3)

Adding the condition of similarity constrained and the function of similarity measure in inverse type to the system, the clustering results are stabilized when sudden (very large or small) change of threshold occur. In the Fig. 1, the range of similarity measure is showed in changing the predetermined threshold $\zeta_i$.

![Fig. 1 Range of similarity by threshold](image)

The mean of cluster is inferred by similarity as eq(4)

$$c_j = \frac{\sum_{i=1}^{N} r_{ij} x_i}{\sum_{i=1}^{N} r_{ij}}$$

(4)

After inferencing the mean of cluster, these covariance $\Sigma_j$ is inferred using Maximum Likelihood Estimation (MLE)[9-11], with the prior probability as follows.

$$\Pr(x_i) = \frac{1}{N} \sum_{i=1}^{N} u_{ij}$$

(5)

where $u_{ij}$ is a element of partition matrix $U$ in $j$ th cluster after the normalization step followed by the product step with prior probability in equation (6) and (7).

$$p_{ij} = \frac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} \times \exp \left( -\frac{1}{2} (x_i - c_j)^T \Sigma_j^{-1} (x_i - c_j) \right)$$

(6)

$$u_{ij} = \frac{p_{ij} \Pr(x_i)}{\sum_{i=1}^{N} p_{ij} \Pr(x_i)}$$

(7)

The initial number of cluster in proposed method is equal to number of data patterns. In the processing a algorithm, cluster parameters are converge with high densities of data space. And we introduce modified Subtractive clustering[2][12] in learning step to solve that perform the high converging speed, difficult of converging in the case of large data set. In the algorithm, modified subtractive algorithm make a meaningful cluster using cumulative densities of inferred cluster only the meaningful cluster is represented in equation (8).

$$c_{i,\text{new}} = c_i, \text{ if } D_j = \max \left( \sum_{i=1}^{N} u_{ij} \right)$$

(8)

Using the means that infers clusters, we eliminate the similarities in algorithm are calculate as shown in equation (9) and then neighbor clusters are eliminated.

$$s_{i,\text{new}} = \exp \left( -\frac{1}{2} (c_i - c_{i,\text{new}})^T \Sigma_i^{-1} (c_i - c_{i,\text{new}}) \right)$$

(9)

Protecting the eliminate with unnecessary clusters, we introduce convergence limit value $\epsilon$ as equation (2), algorithm can re-calculate as equation (10).

$$s_{i,\text{new}} = \begin{cases} 0, & \text{if } s_{i,\text{new}} < \epsilon \\ s_{i,\text{new}}, & \text{otherwise} \end{cases}$$

(10)

Figure 2 show the result that effects of eliminate at limit $\epsilon$ to 0.9

![Fig. 2 Effect of $\epsilon$](image)

After this step, centers of new meaningful clusters are inferred using equation (11).

$$c_{i,\text{new}} = c_i, \text{ if } D_j = \max \left( D_{\text{max}} - \sum_{i=1}^{N} s_{i,\text{new}} \right)$$

(11)

Algorithm from equation (8) to (11) repeats until termination conditions are satisfied. After the steps, the clustering algorithm learns one epoch then check the termination conditions.
3. LEARNING METHOD AND NEURO-FUZZY MODEL

3.1 TSK fuzzy model

We use Tagaki-Sugeno-Kang(TSK) fuzzy model which have the properties that have linguistic inputs and polynomial output show its model from layer 1 to layer 5 in figure 3 in fig 3.

\[ R^i: \text{IF } x \text{ is } A_i \text{ and } y \text{ is } B_2 \]
\[ \text{THEN } f_i = p_i x + q_i y + r_i \]

Figure 4 shows concept of input-output relation of TSK fuzzy models.

The overall output of model are inferred by weighted average using each rule bases as (13).

\[ f = \sum_{i=1}^{r} \mu_i f_i \]
\[ \mu_i = \exp \left( -\frac{1}{2} \frac{(x-c_i)^2}{\sigma_i^2} \right) \]

Cluster parameters in (14) through self-constructing clustering are directly used in neuro-fuzzy model as the same clustering parameters and membership function of model. Also overall inferred output are summation in (13). In detail, each output \( \overline{\mu}_i f_i \) is considered as the independent outputs of submodel and performance of clustering. The errors of each fuzzy submodel are show as (15).

\[ e_{i,\text{neuro fuzzy}} = y - \overline{\mu}_i f_i \]

Using (15), we construct the error weighting function \( e_{ij} \) as (16).

\[ e_{ij} = \exp \left( -\frac{1}{2} \frac{(e_{i,\text{neuro fuzzy}} - \mu_{ij})^2}{\sigma_{ij}^2} \right) \]

The similarity between clusters are improved by altering from (3) to (17).

The overall model, self-constructing clustering are used to estimate the number of cluster and optimize the parameters in learning step. The inferenced parameters directly used to the neuro-fuzzy model and to learn the consequent parameters by least square method. In the continuous learning process, proposed model is constructed of the structure identification and the parameter identification simultaneously. We extended the proposed model by clustering based learning instead general error derivative based learning as shown in Figure 5.

4. SIMULATION AND RESULTS

4.1 Example 1: Box-Jenkins Gas Furnace

Box and Jenkins gas furnace data[2] is conventional by nonlinear time series experiments at sampling with each 9 sec and composed input-output data pairs with 296 data sets. The input is flow of methane and output is rate of \( CO_2 \) and given data set is composed as

\[ y(t-1), y(t-2), y(t-3), y(t-4), u(t-1), u(t-2), u(t-3), u(t-4), u(t-5), u(t-6) \] at same times.
Using the Jang's input selection method, we choose the training and checking data \( y(k + 1) = f(y(k), u(k - 3)) \). Also we select the training data set from odd number of data pairs and use checking data set of the whole data.

Using the input-output relation, we constructed the TSK neuro-fuzzy model as (18).

\[
\text{if} \ u(k - 3) \text{is } A_i \text{ and } y(k) \text{ is } B_j \text{ then } y(k + 1) = \sum_{i=1}^{I} \sum_{j=1}^{J} \beta_{ij} u(k - 3) + \sum_{i=1}^{I} \sum_{j=1}^{J} \gamma_{ij} y(k) + \sum_{i=1}^{I} \sum_{j=1}^{J} \delta_{ij} \]

(18)

(1) Clustering Results

Firstly we set predetermined threshold \( \zeta \) to 0.1 and consequent similarity \( t \) to 0.2 and learning 50th epoch. In this learning, learned model has 14 clusters and the Root Mean Square Error of training(learning) step has 0.1037 and error of checking step has 0.1697. Fig. 7 shows the results of clustering after training step in premise part.

(2) Results of clustering based neuro-fuzzy modeling

We show the membership functions after the learning in figure 8 and figure 9.

Also the experiment results of the changes of a predetermined threshold \( \zeta \) or consequent similarity \( t \), show the comparison with previous ones presented in table 1. Compares the results with previous one using RMSE. As shown in table 1, the proposed system reduces not only RMSE with training data, but also RMSE with checking data.

<table>
<thead>
<tr>
<th>Error(RMSE)</th>
<th>rules</th>
<th>Training error</th>
<th>Checking error</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedrycz[13]</td>
<td>81</td>
<td>0.320</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Xu[14]</td>
<td>25</td>
<td>0.328</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Sugeno[15]</td>
<td>2</td>
<td>0.359</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Abonyi[16]</td>
<td>16</td>
<td>0.154</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Babuska[17]</td>
<td>23</td>
<td>0.124</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Jang[6]</td>
<td>4</td>
<td>0.135</td>
<td>0.530</td>
<td></td>
</tr>
<tr>
<td>Oh[18]</td>
<td>4</td>
<td>0.026</td>
<td>0.272</td>
<td></td>
</tr>
<tr>
<td>Proposed</td>
<td>( \zeta = 0.1, t = 0.4 )</td>
<td>9</td>
<td>0.1282</td>
<td>0.1863</td>
</tr>
<tr>
<td>Proposed</td>
<td>( \zeta = 0.2, t = 0.4 )</td>
<td>5</td>
<td>0.1267</td>
<td>0.1523</td>
</tr>
<tr>
<td>Proposed</td>
<td>( \zeta = 0.1, t = 0.2 )</td>
<td>14</td>
<td>0.1037</td>
<td>0.1697</td>
</tr>
</tbody>
</table>

4.2 Example 2 : Iris flower Classification

Fisher's iris classification problem is conventional benchmark problem in pattern recognitions. Data sets composed four inputs and one output for three patterns and is set as the 150th data. The each inputs describes four characteristics as follows.

1 : sepal length (SL)
2 : sepal width (SW)
3 : petal length (PL)
4 : petal width (PW)

Also output is consisted of three classes as follow.
iris setosa (class 1)
iris veresicolor (class 2)
iris virginica (class 3)

Each class have 50 data sets and first class linearly derives other two class easily, but the second class and last class can not be derived linearly. Input-output relations represented in (19).

\[
y(\text{class}) = f(SL, SW, PL, PW)
\] (19)

(1) Results of clustering
First, we set predetermined threshold \( \zeta \) to 0.9 and consequent similarity \( t \) to 0.2 and learning 100th epoch. The results yield 5 clusters after learning and figure 12 display with data sets.

![Fig. 12 Clustering result in iris data](image)

(2) Results of clustering based on neuro-fuzzy modeling
Fig 13 and 14 show the membership function of SW and PW in the after learned model.

![Fig. 13 Membership function of SW in iris data](image)

![Fig. 14 Membership function of PW in iris data](image)

Also, the results in table 2 describe the case that the predetermined threshold \( \zeta \) and the consequent similarity \( t \) are varied.

<table>
<thead>
<tr>
<th>Error(MSE) Method</th>
<th>Number of rules</th>
<th>Recognition rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>T. P. Hong, C. Y. Lee[19]</td>
<td>95.570</td>
<td></td>
</tr>
<tr>
<td>T. P. Hong, J. B. Chen[20]</td>
<td>95.570</td>
<td></td>
</tr>
<tr>
<td>C. H. Chang, S. M. Chen[21]</td>
<td>96.070</td>
<td></td>
</tr>
<tr>
<td>T. P. Wu, S. M. Chen[22]</td>
<td>96.210</td>
<td></td>
</tr>
<tr>
<td>F. M. Tsai, S. M. Chen[23]</td>
<td>95.833</td>
<td></td>
</tr>
<tr>
<td>T. P. Hong, J. B. Chen[25]</td>
<td>97.333</td>
<td></td>
</tr>
<tr>
<td>Proposed ( \zeta = 0.05, t = 0.5 )</td>
<td>8</td>
<td>97.333</td>
</tr>
<tr>
<td>Proposed ( \zeta = 0.05, t = 0.2 )</td>
<td>6</td>
<td>97.333</td>
</tr>
<tr>
<td>Proposed ( \zeta = 0.09, t = 0.2 )</td>
<td>5</td>
<td>98.67</td>
</tr>
</tbody>
</table>

In the iris classification, the optimized model has no error in classifying the training data but has some error in the checking process, some papers have reported both classified with no error.

5. CONCLUSION

In this paper, we proposed a new leaning scheme of neuro-fuzzy model with self-constructing clustering method. The proposed model extends the learning method from error derivative based to clustering based, and automatically identifies the structures and parameters of neuro-fuzzy model simultaneously in modeling process and shows through simulation the benefits comparing with previous ones.

1. Using the self-constructed clustering, proposed algorithm perform detecting the number of clusters and optimizing its parameters simultaneously.

2. In the overall model, the proposed a clustering based learning by the structure identification and the parameter identification using clustering method.
3. We extended the learning method of clustering from unsupervised learning to supervised learning using input-output relation as a neuro-fuzzy model.

We would like to extend proposed method in a way to reduce computational loach such that we modify the structure of neuro-fuzzy model for improving performance of overall model and so on.

REFERENCES


