Robust Function Approximation Using Fuzzy Rules with Ellipsoidal Regions

Hiroyasu Kubota, Hisashi Tamaki, and Shigeo Abe
Graduate School of Science and Technology, Kobe University, Kobe, Japan
E-mail: abe@eedept.kobe-u.ac.jp

Abstract

This paper discusses robust function approximation when the Takagi-Sugeno type model is used for the consequent part of fuzzy rules. With this model, the parameters of the liner equation that defines the output value of the fuzzy rule are determined by the least-squares method. Therefore, if the training data include outliers, the method fails to determine the parameter values correctly. To overcome this problem we use the least median of squares method. Among the original training data set, we randomly select training data more than the number of parameters, and determine the parameter values using the least-squares method. We repeat this many times and determine the parameters with the smallest median of squared errors. We compare the proposed method with the least-squares method and the conventional least median of squares method using the data generated by the Mackey-Glass differential equation.

1 Introduction

Fuzzy systems are suited for nonlinear function approximation and many fuzzy systems with training capabilities have been developed [1]–[7]. In general, fuzzy systems have faster training capabilities than multilayer neural networks and have comparable generalization ability. But most fuzzy systems do not consider the situation when the training data include outliers and their performance is deteriorated.

In statistics, robust regression is an important research topic and several methods have been developed [8], but most of them are applicable to linear systems.

In this paper, we discuss robust function approximation with the Takagi-Sugeno type model for the consequent part of fuzzy rules. Since in the model the output is expressed by a linear combination of input variables, we determine the parameters by the least median of squares method (LMS) [8, 9]. Since inclusion of outliers is usually not known, approximation performance should not be deteriorated when there are no outliers. Thus, unlike the conventional LMS, we repeat determining the parameters with the least-squares method (LS) using randomly selected training data and then select the parameters with the smallest median of squared errors.

In the following we discuss robust fuzzy function approximation based on the fuzzy function approximator with ellipsoidal regions [4]. First we overview the approximator and then we discuss the robust approximation method for the approximator. Finally we compare the performance of the proposed method with our previous method and the conventional LMS.

2 Fuzzy Function Approximator with Ellipsoidal Regions

2.1 Fuzzy Rule Representation

We discuss function approximation using fuzzy rules with ellipsoidal regions as follows:

$$R_i : \text{If} \quad \mathbf{x} \quad \text{is} \quad \mathbf{c}_i \quad \text{then} \quad y = o_i \quad \text{for} \quad i = 1, \ldots, N, \tag{1}$$

where $\mathbf{x}$ is the m-dimensional input vector, $\mathbf{c}_i$ is the center of the ellipsoidal region for the ith fuzzy rule, y is the output for $\mathbf{x}$, o_i is the corresponding output, and N is the number of fuzzy rules. The degree of membership of the fuzzy rule R_i, $m_i(\mathbf{x})$, is given by

$$m_i(\mathbf{x}) = \exp(-d_i^2(\mathbf{x})), \tag{2}$$
$$d_i^2(\mathbf{x}) = (\mathbf{x} - \mathbf{c}_i)^t Q_i^{-1} (\mathbf{x} - \mathbf{c}_i), \tag{3}$$

where $d_i(\mathbf{x})$ is the weighted distance between $\mathbf{x}$ and $\mathbf{c}_i = (c_{i1}, \cdots, c_{im})^t$, Q_i is the $m \times m$ covariance matrix and given by either of the following ways:

1. a constant diagonal matrix with the same diagonal element σ^2; and
2. the diagonal matrix calculated using the set of data around the center $\mathbf{c}_i$, S_i,

$$Q_{i,jj} = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} (x_j - c_{ij})^2, \tag{4}$$

where $|S_i|$ is the number of data in S_i.

In the Takagi-Sugeno type model, instead of using constant o_i in (1), the linear combination of input variables is used:

$$o_i = p_{0i} + p_{1i} x_1 + \cdots + p_{mi} x_m \quad \text{for} \quad i = 1, \ldots, N, \tag{5}$$

where $p_{0i}, \ldots, p_{mi}$ are constants and are determined by the LS when robust approximation is not considered.

The output of fuzzy rules $R_i (i = 1, \ldots, N)$, $\widehat{y}(\mathbf{x})$, for the input $\mathbf{x}$ can be synthesized by the center-of-gravity method:

$$\widehat{y}(\mathbf{x}) = \frac{\displaystyle\sum_{i=1}^{N} o_i m_i(\mathbf{x})}{\displaystyle\sum_{i=1}^{N} m_i(\mathbf{x})}, \tag{6}$$

where o_i are calculated by (5).

2.2 Fuzzy Rule Generation

We generate fuzzy rules dynamically until the approximation error meets the required error limit. For each rule we determine the subset of the training data dynamically that should be used to determine the parameters p_{ji}.

The center of the fuzzy rule needs to be at the center of a cluster where data gather. In [7], for each training datum, the potential is calculated and the training datum with the maximum potential is selected as the cluster center. To avoid consuming the calculation time in clustering, here we use the simplified version of the method discussed in [7]. Namely, to generate the first fuzzy rule, for each training input, we count the number of the training inputs that are within the distance $R(> 0)$ from the training input in consideration. Then we select the training input that has the maximum number of the training inputs within the specified distance as the center of the first fuzzy rule and determine the parameters $p_{01}, p_{11}, \ldots, p_{m1}$ using the data within the distance R from the center.

The ith ($i > 1$) fuzzy rule is generated as follows: We delete the training inputs that are within the specified distance from the center of the kth ($k = 1, \ldots, i-1$) fuzzy rule. For each of the remaining training inputs, we count the number of the training inputs that are within the distance R from the training input in consideration. Then we select the training input that has the maximum number of training inputs within the specified distance as the center of the ith fuzzy rule. We set the set S_i with the training data that are within the distance R from the center. And determine the parameters $p_{0i}, p_{1i}, \ldots, p_{mi}$ using the data in $S_1 \cup \cdots \cup S_i$. By deleting the training inputs that are within the distance R from the already selected centers, we can avoid selecting the training inputs that are near some of the centers.

3 Robust Parameter Estimation

In Takagi-Sugeno type model, the parameters $p_{0i}, \ldots, p_{mi}$ are determined by the LS. Thus if the training data include outliers, approximation performance is worsened. Therefore, to realize robust function approximation, we apply the LMS, instead of the LS, to the fuzzy function approximator with ellipsoidal regions.

3.1 Robust Estimation by the Least Median of Squares Method

When we generate the i fuzzy rule, the parameters $p_{0k}, \ldots, p_{mk} (k = 1, \ldots, i-1)$ have already been determined. Thus from (6) the parameters $p_{0i}, \ldots, p_{mi}$ satisfy

$$\widehat{y} = a_0(\mathbf{x}) p_{0i} + a_1(\mathbf{x}) p_{1i} + \cdots + a_m(\mathbf{x}) p_{mi} = \mathbf{a}(\mathbf{x})^t \mathbf{p}_i \qquad \text{for} \quad \mathbf{x} \in S_1 \cup \cdots \cup S_i, \tag{7}$$

where $a_i(\mathbf{x})$ are determined by (5) and (6), and $\widehat{y}$ is the estimate of the training output y. Define the residual as $r(\mathbf{p}_i) = y_i - \widehat{y}_i = y - \mathbf{a}(\mathbf{x})^t \mathbf{p}_i$ and the set of the data for determining $\mathbf{p}_i$ as $Z_i = \{(\mathbf{x}, y) | \mathbf{x} \in S_1 \cup \cdots \cup S_i\}$. Then the estimate $\widehat{\mathbf{p}}_i(Z_i)$ of $\mathbf{p}_i$ by the LMS is given by

$$\widehat{\mathbf{p}}_i(Z_i) = \arg \min_{\mathbf{p}_i} (r^2(\mathbf{p}_i))_{h:|Z_i|} = \arg \min_{\mathbf{p}_i} |r(\mathbf{p}_i)|_{h:|Z_i|}, \tag{8}$$

where $|Z_i|$ is the number of the input-output pairs in Z_i and for $|Z_i|$ outputs of a real valued function $r(x)$, $r(x)_{h:|Z_i|}$ stands for the h-th element of the ordered values. In this study we take $h = [n/2] + 1$.

In general to minimize the objective function given by (8) is difficult. Therefore, in [8, 9], the following resampling algorithm is used.

1. Randomly select $m + 1$ input-output pairs from Z_i and solve a set of linear equations $y = a(\mathbf{x})^t \mathbf{p}_i$ for $\mathbf{p}_i$.

2. Using $\mathbf{p}_i$, for $|Z_i|$ training data, calculate the squared errors between the estimated outputs and training outputs, and calculate and memorize the median.

3. Repeat 1 and 2 many times while keeping $\mathbf{p}_i$ with the smallest median of squared errors.

Since in the above algorithm, $\mathbf{p}_i$ is determined by solving a linear equation, $\mathbf{p}_i$ may be specialized to the selected input-output pairs and approximation performance may not be good when there are no outliers. Thus we determine $\mathbf{p}_i$ by the least-squares method using the subset of Z_i as follows.

1. Randomly select an $l (l > m+1)$-element subset Z from Z_i and determine $\mathbf{p}_i$ by minimizing the squared error $\sum_{(\mathbf{x},y)\in Z}(y - a(\mathbf{x})^t \mathbf{p}_i)^2$.

2. Using $\mathbf{p}_i$, for $|Z_i|$ training data, calculate the squared errors between the estimated outputs and the training outputs, and calculate and memorize the median.

3. Repeat 1 and 2 many times while keeping $\mathbf{p}_i$ with the smallest median of squared errors.

4 Performance Evaluation

4.1 Mackey-Glass Differential Equation

Using the Mackey-Glass differential equation we evaluate the proposed method, the conventional LMS, and the LS. The equation generates time series data with a chaotic behavior and is given by

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t), \tag{9}$$

where t and τ denote time and time delay. By integrating (9), we obtain the time series data $\{x(0), x(1), \ldots\}$. Using x prior to time t, we predict x after t. Setting $\tau = 17$, and using four inputs $x(t-18), x(t-12), x(t-6), x(t)$, we estimate $x(t+6)$. The first 500 data from the time series data $x(118), \ldots, x(1117)$ were used to generate fuzzy rules (namely, $n = 500$) and the remaining 500 data were used to test approximation performance.

4.2 Including Outliers in the Data

We generated the following two training data sets that include outliers:

- Data set I : We randomly selected 15 values from the first 100 input-output pairs and 5 values from the last 100 input-output variables and replaced them with 10.0.

- Data set II: We randomly selected 25 values from the first 100 input-output pairs and multiplied them with 10.

For the conventional LMS, the LS, and the proposed method a constant covariance matrix with $\sigma = 0.03$ was used and R was set to 0.05. For the proposed method we set $l = 20$, namely four times the number of input and output variables. We repeated resampling 1000 times for the proposed method and 3000 times for the conventional LMS and we used the NRMSE (normalized root-mean-square error) and median for measuring the performance.

Tables 1 and 2 show the results for data sets I and II, respectively. Both the proposed method and the conventional LMS are superior to the LS in both cases, and the proposed method is better than the conventional LMS in the medians of the squared errors for the training and test data.

Figure 1 (a) shows the predicted test data by the LS for the data set II. Because of the outliers, the prediction is erroneous. Figure 1 (b) shows the predicted test data by the proposed method. The prediction is not affected by outliers.

4.3 Without Outliers

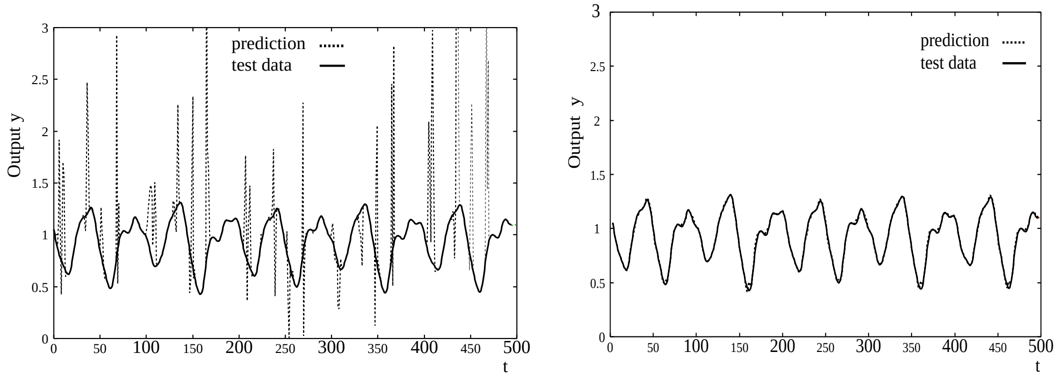
With the same evaluation conditions, we evaluate approximation performance when there are no outliers in the training data. Table 3 shows the results. The approximation performance by the proposed method is a

Table 1: Approximation performance for data set I

Method	Training Data		Test Data	
	NRMSE	Median	NRMSE	Median
LS	0.903	0.00552	2.42	0.00590
LMS	1.00	0.00460	0.0624	0.00495
Proposed	1.00	0.00148	0.0665	0.00214

Table 2: Approximation performance for data set II

Methods	Training Data		Test Data	
	NRMSE	Median	NRMSE	Median
LS	0.909	0.00525	1.91	0.00475
LMS	0.994	0.00478	0.0640	0.00475
Proposed	0.993	0.00195	0.0505	0.00233



(a) The LS method

(b) The proposed method

Figure 1: Comparison between the proposed method and the LS for the Mackey-Glass data

Table 3: Approximation performance without outliers

Method	Training Data		Test Data	
	NRMSE	Median	NRMSE	Median
LS	0.0200	0.00174	0.0207	0.00164
LMS	0.0563	0.00393	0.0548	0.00429
Proposed	0.0338	0.00144	0.0322	0.00205

little worse than that of the LS in both NRMSE and the median of squared errors for the test data. However approximation performance by the proposed method is superior to that of the conventional LMS.

For the data set I, the LS method took 11 seconds, the proposed method took 489 seconds and the conventional LMS took 1016 seconds using SUN Ultra Sparc-IIi 335MHz. Both the proposed method and the conventional LMS took more computation time than the LS because of resampling.

5 Conclusions

In this paper we proposed robust function approximation by extending the conventional least median of squares method. Namely, to make approximation performance comparable with the LS when no outliers exist, we combined the LS with resampling. We evaluated the proposed method for the time series data generated by the Mackey-Glass differential equation and showed that approximation performance of our method was better than that of the conventional LMS and robuster than that of the LS when outliers were included and comparable when no outliers were present.

References

[1] S.Abe, *Neural Networks and Fuzzy Systems: Theory and Applications*, Kluwer Academic, Boston, 1996.

[2] L. X. Wang and J. M. Mendel, ”Generating Fuzzy Rules by Learning from Examples,” *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 22. No. 6, pp. 1414–1427, 1992.

[3] R. Thawonmas, S. Abe, ”Function Approximation Based on Fuzzy Rules Extracted from Partitioned Numerical Data,” *IEEE Trans. Systems, Man, and Cybernetics–Part B*, Vol. 29, No. 4, 1999.

[4] S. Abe, ”Fuzzy Function Approximators with Ellipsoidal Regions,” *IEEE Trans. Systems, Man, and Cybernetics–Part B*, Vol. 29, No. 5, pp. 654–661, 1999.

[5] R. Katayama, ”Performance Evaluation of Self Generating Radial Basis Function for Function Approximation,” in *Proc. International Joint Conference on Neural Networks*, Vol. 1, pp. 471–474, Nagoya, Japan, October 1993.

[6] T. Takagi and M. Sugeno, ”Fuzzy Identification of Systems and Its Applications to Modeling and Control,” *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 15, No. 1, pp. 116–132, 1985.

[7] S. L. Chiu, ”Fuzzy Model Identification Based on Cluster Estimation,” *Journal of Intelligent and Fuzzy Systems*, Vol. 2, pp. 267–278, 1994.

[8] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Detection*,” John Wiley, New York, 1987.

[9] P. J. Rousseeuw and M. Hubert, ”Recent developments in Progress,” Institute of Mathematical Statistics Lecture Notes-Monograph Series, Vol. 31, pp. 201–214, Hayward, California, 1997.