Lazy Learning for Nonparametric Locally Weighted Regression

Seok-Beom Roh¹ , Yong Soo Kim² , and Tae-Chon Ahn³

¹Department of Electrical Engineering, University of Suwon, Hwaseongi, Korea

²Department of Computer Engineering, Daejeon University, Dong-gu, Daejeon, Korea

³Department of Electronics Convergence Engineering, Wonkwang University, Iksan, Korea

Ijfis

Abstract

In this study, a newly designed local model called locally weighted regression model is proposed for the regression problem. This model predicts the output for a newly submitted data point. In general, the local regression model focuses on an area of the input space specified by a certain kernel function (Gaussian function, in particular). The local area is defined as a region enclosed by a neighborhood of the given query point. The weights assigned to the local area are determined by the related entries of the partition matrix originating from the fuzzy C-means method. The local regression model related to the local area is constructed using a weighted estimation technique. The model exploits the concept of the nearest neighbor, and constructs the weighted least square estimation once a new query is provided given. We validate the modeling ability of the overall model based on several numeric experiments.

Keywords: k -nearest neighbors, Locally weighted regression, Weighted least square estimation, Lazy learning, Fuzzy C-means clustering

1. Introduction

In the fields of machine learning and data analysis, the analysis of the relation between the input and output variables and the prediction of the output variables from newly given data have been some of the most relevant research topics [1]. In the studies dealing with the associated regression problem, two essential and conceptually diversified development strategies have been introduced: global and local learning techniques [2]. Global learning [3] assists in fitting a sophisticated model in terms of the structure to all the observation data patterns. Accordingly, the model that has been learned globally is valid throughout the input space. However, local learning [4, 5] focuses on and fits a relatively simple model to a subset of observations that are present only in a neighborhood of the query point [2]. In this scenario, the model identified by local learning pertains to some limited region of the entire input space, where the query point and its neighbors are positioned.

Local learning techniques, such as k -nearest neighbors (k -NN) classifiers [6], linear regression interpolation, and local interpolation [7], have been successfully applied to various learning tasks (refer to [8–13]).

A key issue in local learning techniques is the method to define a subset of all the data patterns to be used, to estimate the parameters of the local model at a related query point.

In this study, we are focused on developing a simple method to determine the neighbors of

Received: Mar. 16, 2020

Revised : May 12, 2020

Accepted: May 24, 2020

Correspondence to: Tae-Chon Ahn

(tcahn@wku.ac.kr)

©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

the given query point, without confining to any adaptive technique. Here, we develop a new nonparametric locally weighted regression (LWR), which is applicable within a region defined by the k -NN approach and in which the weight of each nearest neighbor is determined by fuzzy clustering. Typically, locally weighted learning (LWL) [14] is used to build a local model and estimate the coefficients.

The k -NN method, the first component of the proposed model, is the one of the most well-known non-parametric models. Local learning or instance-based learning (IBL) is applied here. In IBL, the learning phase is deferred until a new data pattern is provided. In the k -NN approach, the output value of a query point is estimated by taking the average of the target values corresponding to the k -NN of the query. LWR is the second functional component of the proposed model.

Although it is very difficult to determine an appropriate structure for a global model, it is unnecessary to do so for a local model in LWR. Therefore, LWR is useful in the modeling of complex non-linear systems [15]. The proposed model, referred to as non-parametric LWR is a regression model, is available within an area in which the k -NN of a query point is formed, and its coefficients are estimated using the weighted least square estimation (WLSE) with L_2 regularization. In this study, we apply the relative distance kernel of fuzzy clustering to complete the weight allocation. The distance kernel function is used to calculate the activation levels of the data patterns in fuzzy C-means (FCM) clustering [16].

This paper is organized as follows. First, in Section 2, we discuss the LWR based on the k -NNs exploited in the regression problem. The problem of local regression based on the k -NNs is further elaborated in Section 3. In Section 4, we present the experimental results. The conclusions are summarized in Section 5.

2. Locally Weighted Regression with k -NN Approach

The k -NN approach is a representative method of nonparametric modeling. k -NN learning is one of the lazy learning (i.e., IBL) techniques that defer the decision of the parameters of a model until a query point is provided newly. When a query instance is given, a set of similar patterns is determined among the already obtained training patterns set and is used to estimate the output of the new instance [17]. There are two well-known advantages of lazy learning: effective usage of local information available in the input space and short training time [1].

Table 1. Basic notations

$(\mathbf{x}, y)$	Query pattern (test pattern) and its target ($\mathbf{x} \in \mathbb{R}^m, y \in \mathbb{R}$)
$\hat{y}$	Predicted target of a query
$(\mathbf{x}_i, y_i)$	Reference pattern (i.e., training pattern) and its target ($\mathbf{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R}, i = 1, \dots, n^*$)
$\mathbf{w}$	Weight matrix (a diagonal matrix with nonzero weights positioned on its diagonal)
$K(\mathbf{a}, \mathbf{b})$	Kernel function ($\mathbf{a}, \mathbf{b}$: location vectors)

* is $i = 1, 2, \dots, n$, where n denotes the number of data patterns.

The key assumption behind the k -NN approach is that the data patterns close to a newly provided query point may have similar target values used in the prediction task. The neighborhood of a query point, $\mathbf{x}$, is defined by the k -NN method based on some distance function. The Euclidean distance or its generalized weighted version is commonly used as a distance function.

In the detailed description of the algorithm, we use the notations listed in Table 1.

The set of indices of the selected neighborhoods for a query is defined as follows:

$$L = \{j \mid \mathbf{x}_j \text{ is one of the } k \text{ nearest instances to query } \mathbf{x} (j = 1, \dots, n)\}. \quad (1)$$

Notation $L[j]$ is used to represent the j th element of set L .

After deciding the neighborhood of a query point $\mathbf{x}$, we estimate the target value at $\mathbf{x}$, which is commonly encountered in classification problems, as follows:

$$\hat{y} = \underset{c}{\operatorname{argmax}} \sum_{i=1}^k g(y_{L[i]}, c), \quad (2)$$

$$\text{where } g(y_{L[i]}, c) = \begin{cases} 1, & \text{if } y_{L[i]} = c, \\ 0, & \text{if } y_{L[i]} \neq c. \end{cases}$$

In case of function approximation, the k -NN method estimates the target value using a weighted average computed within some varying neighborhood [18] as Eq. (3).

$$\hat{y} = \frac{\sum_{i=1}^k u_i \cdot y_{L[i]}}{\sum_{i=1}^k u_i}, \quad (3)$$

where $u_i = K(\mathbf{x}_{L[i]}, \mathbf{x})$.

It is known that the output determined by locally weighted

averages could be significantly biased at the boundaries of the local region [19]. To overcome this limitation of the conventional k -NN learning, we apply the LWR. Although a linear model has two representative benefits—simplicity and easiness of use—it has a critical drawback: a relatively high bias. Specifically, if the target function is highly complex that a linear function cannot well approximate it, then a linear regression may present a poor modeling performance. Local linear regression was introduced from the viewpoint that any complex function can be well approximated by a simple linear function over an appropriate subspace in the input space [20].

The LWR is derived from the standard linear regression. The importance of a relevant instance is increased and that of an irrelevant one is decreased by weighting the data [19].

After determining the neighbors of a given query point, $\mathbf{x}$, the weight of each neighbor is calculated [21]. This value indicates the importance of a particular neighbor to the training of the local regression. The weight function is determined based on the similarity between the given query point, $\mathbf{x}$, and its neighbors.

We define the similarity (i.e., the kernel function) between two points, yielding the weight value of each neighbor as follows. The similarity is derived from the partition matrix defined in [23].

The kernel function is generally defined as any explicit functional form (e.g., Gaussian and ellipsoidal). However, the kernel function used in this study is a result of fuzzy clustering called as the fuzzy activation function. The kernel function called as the relative distance function is expressed as

$$K(\mathbf{x}_l, \mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x}_l = \mathbf{x}, \\ \frac{1}{\sum_{j=1}^k \left(\frac{\|\mathbf{x}_l - \mathbf{x}\|}{\|\mathbf{x}_{L[j]} - \mathbf{x}\|} \right)^{\frac{2}{p-1}}}, & \text{if } \mathbf{x}_l \neq \mathbf{x}, \end{cases} \quad (4)$$

where $l \in L$, $\mathbf{x}_l$ is one of the nearest neighbors, and $p \in (1, \infty)$ denotes the fuzzification coefficient used in the FCM.

The fuzzification coefficient, p , determines the fuzziness of the partition produced in the clustering. When the fuzzification coefficient approaches 1, the resulting partition asymptotically approaches a Boolean (0 – 1) partition of the data. However, the fuzziness of the partition increases with the increase in the values of the fuzzification coefficient.

The output value of the query point, $\mathbf{x}$, is estimated by the weighted average of the already given output values, y_i , of the

neighbors of $\mathbf{x}$.

$$\begin{aligned} \hat{y} &= \sum_{i=1}^k K(\mathbf{x}_{L[i]}, \mathbf{x}) \cdot y_{L[i]} \\ &= \sum_{i=1}^k \left(\frac{y_{L[i]} \left(\frac{\|\mathbf{x}_{L[i]} - \mathbf{x}\|}{\|\mathbf{x}_{L[j]} - \mathbf{x}\|} \right)^{\frac{2}{p-1}}}{\sum_{j=1}^k \left(\frac{\|\mathbf{x}_{L[i]} - \mathbf{x}\|}{\|\mathbf{x}_{L[j]} - \mathbf{x}\|} \right)^{\frac{2}{p-1}}} \right), \end{aligned} \quad (5)$$

where $\hat{y}$ denotes the estimated output value for the newly given point, $\mathbf{x}$.

3. Lazy Learner for Local Regression based on Neighborhood Information

Here, we present the development of a methodology to design the local regression defined within an area in which several neighbors of a given query point are positioned.

The local models being used in this study involve a constant and a linear regression. A linear regression might exhibit a high modeling bias when dealing with nonlinear systems. A local linear regression focuses on the operating region only (i.e., the local area enclosed by the nearest neighbors); therefore, it can compensate for the high bias of the overall global linear regression.

First, when a query point is provided, the local area should be determined by the nearest neighbors. The “ k ” nearest neighbors of the given query point are determined in terms of the Euclidean distance or its variant.

The obtained nearest neighbors determine the local area where the local model becomes available (suitable). If a new query point is provided, then in return, a new local area is defined.

After determining the local area, the local model has to be identified. To estimate its parameters, the least square estimation (LSE) is used. In this case, WLSE is more aligned with the k -NN rationale than the typical LSE.

To implement the local model within an area defined by the nearest neighbors of the query point, we substitute the kernel function (i.e., Gaussian function) with the similarity function defined in Eq. (14).

When a new data point is newly provided, the already implemented local model related to the old query point is discarded and a new one is designed based on the information associated with the new given query point. Specifically, whenever a query point is provided newly, the coefficients of the local regression

are re-estimated.

As noted, the local models could be either constant or linear. Both these options were subjects of this study.

3.1 Constants

A constant local model is defined as follows:

$$f(\mathbf{x}) = a, \quad (6)$$

where $\mathbf{x}$ is a vector of the input variables.

We modify the local objective function to be minimized,

$$\begin{aligned} J(\mathbf{x}) &= \sum_{i=1}^k (y_{L[i]} - f(\mathbf{x}_{L[i]}))^2 \cdot (K(\mathbf{x}_{L[i]}, \mathbf{x}))^s + \lambda a^2 \\ &= \sum_{i=1}^k (y_{L[i]} - a)^2 \cdot \left(\frac{1}{\sum_{j=1}^k \left(\frac{\|\mathbf{x}_{L[i]} - \mathbf{x}\|}{\|\mathbf{x}_{L[j]} - \mathbf{x}\|} \right)^{\frac{2}{p-1}}} \right)^s + \lambda a^2, \end{aligned} \quad (7)$$

where s denotes the learning effect coefficient (or the so-called linguistic modifier) of the kernel function, $K(\mathbf{x}_{L[i]}, \mathbf{x})$, defined in Eq. (4).

This learning effect coefficient “ s ” controls the shape of the weight function. In Figure 1, we illustrate the variation in the shape of the weight function as a function of the values of “ s .” As shown the increase in n , s reduces the width of the kernel function, whereas the decrease in n increases the width.

Figure 1. Variation in shape of weight function.

There are two extreme cases: s approaches 0 and infinity. In the first case (i.e., $s \rightarrow 0$), the kernel function can cover the entire universe of the discourse, whereas when $s \rightarrow \infty$, it can be considered as a singleton [24].

The optimized coefficients of the local model (6) can be calculated as follows:

$$a = (\mathbf{1}^T \mathbf{W}^s \mathbf{1} + \lambda)^{-1} (\mathbf{1}^T \mathbf{W}^s \mathbf{y}), \quad (8)$$

$$\text{where } \mathbf{W}^s = \begin{bmatrix} K(\mathbf{x}_{L[1]}, \mathbf{x})^s & 0 & \cdots & 0 \\ 0 & K(\mathbf{x}_{L[2]}, \mathbf{x})^s & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K(\mathbf{x}_{L[k]}, \mathbf{x})^s \end{bmatrix},$$

$\mathbf{1} = [1 \ 1 \ \cdots \ 1]^T$, $\mathbf{y} = [y_{L[1]} \ y_{L[2]} \ \cdots \ y_{L[k]}]^T$, and $L[j]$ denotes the j th element of the set of L nearest neighbors.

In contrast with the global learning, in Eq. (8), only the nearest neighbors are used to estimate the values of this coefficient.

For the constant local model, the WLSE is used to estimate this value, and the shape parameter is equal to 1; the resulting local model is equivalent to the radial basis function neural network (RBFNN), where the radial basis functions are defined by FCM.

The optimized coefficient has the form

$$\begin{aligned} \hat{y} = a &= (\mathbf{1}^T \mathbf{W}^s \mathbf{1} + \lambda)^{-s} (\mathbf{1}^T \mathbf{W}^s \mathbf{y}) \\ &= \frac{\sum_{i=1}^k K(\mathbf{x}_{L[i]}, \mathbf{x}) \cdot y_{L[i]}}{\sum_{i=1}^k K(\mathbf{x}_{L[i]}, \mathbf{x}) + \lambda} \\ &= \frac{1}{1 + \lambda} \cdot \sum_{i=1}^k K(\mathbf{x}_{L[i]}, \mathbf{x}) \cdot y_{L[i]} \\ &= \frac{1}{1 + \lambda} \cdot \sum_{i=1}^k \frac{y_{L[i]}}{\sum_{j=1}^k \left(\frac{\|\mathbf{x}_{L[i]} - \mathbf{x}\|}{\|\mathbf{x}_{L[j]} - \mathbf{x}\|} \right)^{\frac{2}{p-1}}}. \end{aligned} \quad (9)$$

3.2 Linear Regression

The local model with a linear regression is defined as follows:

$$f(\mathbf{x}) = a_0 + \sum_{i=1}^m a_i x_i = \mathbf{a}^T \tilde{\mathbf{x}}. \quad (10)$$

We modify the local objective function to be minimized, which is now expressed as

$$\begin{aligned}
J(\mathbf{x}_q) &= \sum_{i \in L} (y_i - f(\mathbf{x}_i))^2 \cdot (K(\mathbf{x}_i, \mathbf{x}_q))^s + \lambda \sum_{j=0}^m a_j^2 \\
&= \sum_{i \in L} ((y_i - \tilde{\mathbf{x}}_i \mathbf{a}_q)^2 \cdot \left(\frac{1}{\sum_{j \in L} \left(\frac{\|\mathbf{x}_i - \mathbf{x}_q\|}{\|\mathbf{x}_j - \mathbf{x}_q\|} \right)^{\frac{2}{p-1}}} \right)^s \\
&\quad + \lambda \mathbf{a}_q^T \mathbf{a}_q \\
&= (\mathbf{y} - \mathbf{X} \mathbf{a}_q)^T \mathbf{W}_q^s (\mathbf{y} - \mathbf{X} \mathbf{a}_q) + \lambda \mathbf{a}_q^T \mathbf{a}_q, \quad (11)
\end{aligned}$$

where $\tilde{\mathbf{x}}_i = [1 \ \mathbf{x}_i] = [1 \ x_{i1} \ x_{i2} \ \cdots \ x_{im}]$, $\mathbf{a}_q = [a_{q0} \ a_{q1} \ \cdots \ a_{qm}]^T$,

$$\begin{aligned}
\mathbf{X} &= \begin{bmatrix} \tilde{\mathbf{x}}_{L\{1\}} \\ \tilde{\mathbf{x}}_{L\{2\}} \\ \vdots \\ \tilde{\mathbf{x}}_{L\{k\}} \end{bmatrix} = \begin{bmatrix} 1 & x_{L\{1\}1} & \cdots & x_{L\{1\}m} \\ 1 & x_{L\{2\}1} & \cdots & x_{L\{2\}m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{L\{k\}1} & \cdots & x_{L\{k\}m} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_{L\{1\}} \\ y_{L\{2\}} \\ \vdots \\ y_{L\{k\}} \end{bmatrix}, \\
\mathbf{W}_q^s &= \begin{bmatrix} K(\mathbf{x}_{L\{1\}}, \mathbf{x}_q)^s & 0 & \cdots & 0 \\ 0 & K(\mathbf{x}_{L\{2\}}, \mathbf{x}_q)^s & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K(\mathbf{x}_{L\{k\}}, \mathbf{x}_q)^s \end{bmatrix}.
\end{aligned}$$

The optimal coefficients of the local model (10) that minimize this local objective function are calculated in the form

$$\mathbf{a}_q = (\mathbf{X}^T \mathbf{W}_q^s \mathbf{X} + \lambda \mathbf{I})^{-1} (\mathbf{X}^T \mathbf{W}_q^s \mathbf{y}), \quad (12)$$

where $\mathbf{I} \in \mathbb{R}^{(m+1) \times (m+1)}$ denotes the identity matrix.

The pseudo code of the proposed model is as follows.

Step 1. The k nearest neighbors of the query point are determined.

Step 2. The weight values are calculated by Eq. (4).

Step 3. The coefficient of a local model is calculated by Eq. (8) or (12) (when a constant model is used, Eq. (8) is used to estimate the coefficient, and in the case of a linear function, we use Eq. (12) to estimate the coefficients).

Step 4. The final output of the local model is calculated by Eq. (10).

4. Experimental Studies

We conducted several experiments to validate the generalization performance of the proposed model and compared its behavior with those of other models already introduced in the literature. Synthetic low-dimensional data are dealt with in the first subsection. In the second subsection, some selected machine learning datasets from a machine learning repository (<http://www.ics.uci.edu/~mllearn/MLSummary.html>) are used to validate the proposed model. We use a 10-fold cross-validation technique to determine the important structural parameters. First, all the data patterns are divided into 10 folds, and each fold is then selected as a test dataset to validate the generalization ability in consecutive order. The remaining folds are used in the training phase to estimate the coefficients of the local model with several structural parameters.

We summarize the results of the experiments in terms of the mean and standard deviation of the performance index.

The important structural parameters are summarized in Table 2. In this study, the number of nearest neighbors (k), value of the fuzzification coefficient (p), and shape parameter (s) are determined by conducting some cross-validation experiments.

The performance index of a model is defined as the root mean square error (RMSE) as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2}. \quad (13)$$

4.1 Low-Dimensional Synthetic Data

A polynomial with two input variables is defined as follows:

$$f(x, y) = \frac{(x-2)(2x-1)}{(1+x^2)} \frac{(y-2)(2y-1)}{(1+y^2)},$$

Table 2. Structural parameters of the proposed model

Parameter	Value
Polynomial order (O)	0 (constant) or 1 (linear)
Number of nearest neighbors (k)	
$O = 0$	2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
$O = 1$	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Fuzzification coefficient (p)	1.2, 1.4, $\dots$, 3.8, 4.0
Shape parameter (s)	0.2, 0.4, $\dots$, 1.8, 2.0

Table 3. Performance index (RMSE) of the proposed model versus the polynomial order of the basic model and the size of the neighborhood

k	p	s	$O = 0$	k	p	s	$O = 1$
2	3.4	0.2	0.5503±0.0166	-	-	-	-
3	1.4	0.6	0.5163±0.0146	-	-	-	-
4	1.8	1.8	0.4436±0.0245	-	-	-	-
5	1.4	1.2	0.4505±0.0143	-	-	-	-
6	1.2	1.2	0.4533±0.0208	-	-	-	-
7	1.4	1.6	0.4580±0.0182	-	-	-	-
8	1.2	1	0.4573±0.0152	-	-	-	-
9	1.2	1.2	0.4585±0.0182	-	-	-	-
10	1.4	1.8	0.4464±0.0179	10	1.2	2	0.4282±0.0172
20	1.4	2	0.4624±0.0228	20	1.2	1.6	0.4344±0.0223
30	1.4	2	0.4596±0.0147	30	1.2	1.6	0.4342±0.0264
40	1.2	1	0.4604±0.0118	40	1.2	1.6	0.4390±0.0134
50	1.2	1.8	0.4616±0.0169	50	1.2	1.6	0.4323±0.0169
60	1.2	1	0.4660±0.0237	60	1.2	1.6	0.4288±0.0145
70	1.4	1.8	0.4617±0.0222	70	1.2	1.4	0.4367±0.0131
80	1.2	1	0.4556±0.0162	80	1.2	1.4	0.4251±0.0131
90	1.2	1.2	0.4559±0.0182	90	1.2	1.6	0.4310±0.0195
100	1.2	2	0.4602±0.0207	100	1.2	1.2	0.4291±0.0136

Values are presented as mean±standard deviation.

$$x \in [-5, 5], y \in [-5, 5]. \quad (14)$$

The input variables, x and y , were generated from the evenly distributed over the input range $x, y \in [-5, 5]$. Table 3 summarizes the modeling performance (RMSE) of the proposed model according to the appropriate structural parameters of k , p , s , and O .

We note that the number of nearest neighbors that defines the local area is irrelevant to the performance of both the constant local model and the linear local model.

Let us consider the relevance of some other design parameters, such as the fuzzification coefficient (p) and the shape parameter (s), to the performance of the proposed models.

Figure 2 shows the performance index of the proposed models (constant local model and linear local model) according to the fuzzification coefficient (p) and the shape parameter (s).

As shown in Figure 2(a) and 2(b), the increase in the values of the shape parameter (s) and the decrease in the values of the fuzzification coefficient improve the performance of the proposed models under the condition that the number of nearest neighbors is fixed. When $p = 2$, as shown in Figure 2(c) and 2(d), a low value of the shape parameter (s) and an increase in

the number of nearest neighbors lead to the worsening of the performance of the model. When the shape parameter is fixed ($s = 1$), as illustrated in Figure 2(e) and 2(f), we note that large values of the fuzzification coefficient (p) and numerous nearest neighbors worsen the performance.

Table 4 summarizes and compares the performance of the proposed model with the already studied models, i.e., linear regression, second-order polynomial, fuzzy k -NN smoothing approach, and local regression learning using a typical LSE.

4.2 Machine Learning Dataset

The experiments conducted with several datasets obtained from a machine learning repository are discussed in this subsection. To validate the modeling performance of the proposed model, we performed experiments with nine machine learning datasets, whose detailed information is specified in Table 5.

Table 6 summarizes the comparison results of the proposed model and the already studied models that were implemented in WEKA [25] in terms of the performance index. In Table 6, EPI denotes the performance index of the test dataset.

From Table 6, we can see that the proposed model with a

Figure 2. Performance index (RMSE) of the proposed local models according to the fuzzification coefficient (p), shape parameter (s), and number of nearest neighbors (k): constant model with $k = 100$ (a), $p = 2$ (c), and $s = 1$ (e); linear local model with $k = 100$ (b), $p = 2$ (d), and $s = 1$ (f).

Table 4. Results of the comparative analysis

Model	Testing data
Linear regression	2.9309±0.038
Second order polynomial	2.6674±0.0087
Fuzzy k -NN smoothing approach	$k = 5, p = 1.4$ 0.4567 ±0.0234
Local regression with LSE	
Constant local model	$k = 4$ 0.5189±0.0334
Linear local model	$k = 10$ 0.7871±0.0155
Proposed model	
Constant local model	$k = 4, p = 1.8, s = 1.8$ 0.4436 ±0.0245
Linear local model	$k = 20, p = 1.2, s = 0.8$ 0.4251 ±0.0131

Values are presented as mean±standard deviation. Local regression with LSE is a type of local model that is available within an area defined by the nearest neighbors.

Table 5. Specification of the machine learning datasets

Dataset	No. of data patterns	No. of features
Airfoil	1,503	5
Autompg	392	7
Boston housing	506	13
Concrete	1,030	8
CPU	209	6
MIS	390	10
NOx	260	5
Wine quality (red)	1,599	11
Yacht	308	6

linear function is an appropriate model whose average rank is 2.67. The proposed model with a linear function has the best results in six of the nine datasets among all the models in terms of the performance index (i.e., RMSE).

5. Conclusions

In this paper, we introduce a new local model based on the LWR. The proposed model can be considered as the expanded version of the k -NN approach, which is a representative of non-parametric models and the WLSE. When a query point is provided, it invokes lazy learning. First, the local area is defined in the input space, in which the k -NNs of the given query point in terms of the distance are enclosed. Second, the weights of the nearest neighbors are calculated using some predefined similarity function. Finally, the coefficients of the local regression are determined by the WLSE. As a new query is provided, the

already constructed local regression is not available to the query. Therefore, a new local regression, which is available within the new local area surrounded by the new neighborhood of the newly given query point, is invoked. Based on the experimental results summarized in Table 6, local models reconstructed whenever new queries are arrived exhibit better performance than the global models. Therefore, the proposed local model can be considered as an effective approach to deal with complex system modeling. From the experimental results, it is validated that the local regression, which is reconstructed whenever a new query is given, performs better than the typical global models.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

Acknowledgements

This paper was supported by Wonkwang University in 2019.

References

- [1] H. A. Guvenir and I. Uysal, "Regression on feature projections," *Knowledge-Based Systems*, vol. 13, no. 4, pp. 207-214, 2000. [https://doi.org/10.1016/S0950-7051\(00\)00060-5](https://doi.org/10.1016/S0950-7051(00)00060-5)
- [2] G. Bontempi, H. Bersini, and M. Birattari, "The local paradigm for modeling and control: from neuro-fuzzy to lazy learning," *Fuzzy Sets and Systems*, vol. 121, no. 1,

Table 6. Results of the comparative analysis

Data	Proposed model		Multi-layer perceptron [25]	k-NN [25]		Linear regression [25]	Additive regression [25]	SVR [25]	
	$O = 0$	$O = 1$		$k = 3$	$k = 10$			Polynomial kernel	RBF kernel
Airfoil									
EPI	1.91	1.26	4.4	2.59	4.25	4.81	4.93	4.88	4.83
Rank	2	1	5	3	4	6	9	8	7
Autompg									
EPI	2.75	2.51	3.19	2.88	2.99	3.37	3.53	3.44	3.55
Rank	2	1	5	3	4	6	8	7	9
Boston housing									
EPI	3.47	3.09	4.32	4.41	5.22	4.8	4.79	4.95	5.49
Rank	2	1	3	4	8	6	5	7	9
Concrete									
EPI	12.73	11.84	7.91	8.9	9.48	10.47	8.17	10.9	10.8
Rank	9	8	1	3	4	5	2	7	6
CPU performance									
EPI	55.05	45.74	57.25	63.65	74.98	62.79	63.57	64.45	81.02
Rank	2	1	3	6	8	4	5	7	9
MIS									
EPI	0.97	0.96	1.26	1.04	1.07	1.05	1.16	1.08	1.19
Rank	2	1	9	3	5	4	7	6	8
NOx									
EPI	2.65	0.47	1.19	3.45	3.49	4.33	6.47	4.76	9.42
Rank	3	1	2	4	5	6	8	7	9
Wine quality									
EPI	1.08	1.03	0.73	0.69	0.67	0.65	0.67	0.66	0.66
Rank	9	8	7	6	4.5	1	4.5	2.5	2.5
Yacht									
EPI	4.91	2.05	1.12	8.59	9.42	8.91	2.79	10.66	11.94
Rank	4	2	1	5	7	6	3	8	9
Average rank	3.89	2.67	4	4.11	5.5	4.89	5.72	6.61	7.61

pp. 59-72, 2001. [https://doi.org/10.1016/S0165-0114\(99\)00172-4](https://doi.org/10.1016/S0165-0114(99)00172-4)

- [3] J. Yen, L. Wang, and C. W. Gillespie, "Improving the interpretability of TSK fuzzy models by combining global learning and local learning," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 4, pp. 530-537, 1998. <https://doi.org/10.1109/91.728447>

- [4] D. W. Aha, "Feature weighting for lazy learning algo-

rithms," in *Feature Extraction, Construction and Selection*. Boston, MA: Springer, 1998, pp. 13-32.

- [5] F. Dornaika and A. Bosaghzadeh, "Exponential local discriminant embedding and its application to face recognition," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 921-934, 2013. <https://doi.org/10.1109/TSMCB.2012.2218234>

- [6] M. Sarkar, "Fuzzy-rough nearest neighbor algorithms in

- classification,” *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2134-2152, 2007. <https://doi.org/10.1016/j.fss.2007.04.023>
- [7] N. Wang, W. X. Zhang, and C. L. Mei, “Fuzzy nonparametric regression based on local linear smoothing technique,” *Information Sciences*, vol. 177, no. 18, pp. 3882-3900, 2007. <https://doi.org/10.1016/j.ins.2007.03.002>
- [8] P. Angelov, E. Lughofer, and X. Zhou, “Evolving fuzzy classifiers using different model architectures,” *Fuzzy Sets and Systems*, vol. 159, no. 23, pp. 3160-3182, 2008. <https://doi.org/10.1016/j.fss.2008.06.019>
- [9] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21-27, 1967. <https://doi.org/10.1109/TIT.1967.1053964>
- [10] M. R. Gupta, R. M. Gray, and R. A. Olshen, “Nonparametric supervised learning by linear interpolation with maximum entropy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 766-781, 2006. <https://doi.org/10.1109/TPAMI.2006.101>
- [11] S. Park, N. Song, W. Yu, and D. Kim, “PSR: PSO-based signomial regression model,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 19, no. 4, pp. 307-314, 2019. <http://doi.org/10.5391/IJFIS.2019.19.4.307>
- [12] S. I. Cho, T. Negishi, M. Tsuchiya, M. Yasuda, and M. Yokoyama, “Estimation system of blood pressure variation with photoplethysmography signals using multiple regression analysis and neural network,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 18, no. 4, pp. 229-236, 2018. <https://doi.org/10.5391/IJFIS.2018.18.4.229>
- [13] H. S. Yoon and S. H. Choi, “The impact on life satisfaction of nursing students using the fuzzy regression model,” *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 19, no. 2, pp. 59-66, 2019. <https://doi.org/10.5391/IJFIS.2019.19.2.59>
- [14] C. G. Atkeson, A. W. Moore, and S. Schaal, “Locally weighted learning,” in *Lazy Learning*. Dordrecht, Netherlands: Springer, 1997, pp. 11-73. https://doi.org/10.1007/978-94-017-2053-3_2
- [15] H. Leung, Y. Huang, and C. Cao, “Locally weighted regression for desulphurisation intelligent decision system modeling,” *Simulation Modelling Practice and Theory*, vol. 12, no. 6, pp. 413-423, 2004. <https://doi.org/10.1016/j.simpat.2004.06.002>
- [16] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York, NY: Plenum Press, 1981.
- [17] J. M. Valls, I. M. Galvan, and P. Isasi, “Learning radial basis neural networks in a lazy way: a comparative study,” *Neurocomputing*, vol. 71, no. 13-15, pp. 2529-2537, 2008. <https://doi.org/10.1016/j.neucom.2007.10.030>
- [18] P. Kang and S. Cho, “Locally linear reconstruction for instance-based learning,” *Pattern Recognition*, vol. 41, no. 11, pp. 3507-3518, 2008. <https://doi.org/10.1016/j.patcog.2008.04.009>
- [19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer, 2001. <https://doi.org/10.1007/978-0-387-21606-5>
- [20] M. R. Gupta, E. K. Garcia, and E. Chin, “Adaptive local linear regression with application to printer color management,” *IEEE Transactions on Image Processing*, vol. 17, no. 6, pp. 936-945, 2008. <https://doi.org/10.1109/TIP.2008.922429>
- [21] G. Wen, L. Jiang, and J. Wen, “Using locally estimated geodesic distance to optimize neighborhood graph for isometric data embedding,” *Pattern Recognition*, vol. 41, no. 7, pp. 2226-2236, 2008. <https://doi.org/10.1016/j.patcog.2007.12.015>
- [22] W. Zhang, X. Xue, Z. Sun, H. Lu, and Y. F. Guo, “Metric learning by discriminant neighborhood embedding,” *Pattern Recognition*, vol. 41, no. 6, pp. 2086-2096, 2008. <https://doi.org/10.1016/j.patcog.2007.11.023>
- [23] W. Pedrycz, “Conditional Fuzzy C-Means,” *Pattern Recognition Letter*, vol. 17, no. 6, pp. 625-631, 1996. [https://doi.org/10.1016/0167-8655\(96\)00027-X](https://doi.org/10.1016/0167-8655(96)00027-X)
- [24] B. Bouchon-Meunier, “Linguistic hedges and fuzzy logic,” in *Proceedings of IEEE International Conference on Fuzzy Systems*, San Diego, CA, 1992, pp. 247-254. <https://doi.org/10.1109/FUZZY.1992.258625>

- [25] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10-18, 2009. <https://doi.org/10.1145/1656274.1656278>

Seok-Beom Roh received his M. Sc. degrees in computer engineering and the Ph.D. degrees in control and measurement engineering, from Wonkwang University, Korea, in 1996 and 2006, respectively. He is currently a Research Profes-

sor with the Department of Electrical Engineering, University of Suwon, Hwasung, Korea. His research interests include fuzzy system, fuzzy neural networks, advanced computational intelligence, and machine learning.

E-mail: nado9025@gmail.com

Yong Soo Kim graduated in electrical engineering from Yonsei University, Seoul, Korea, in 1981. He received the Ph.D. degree in electrical engineering from Texas Tech University, USA, in 1993. From 1995, he has worked in the Department of

Computer Engineering at the Daejeon University, Korea. He is now a professor in the same university. His research activities are focused on fuzzy neural networks, deep neural networks, pattern recognition, and image processing.

E-mail: kystj@dju.kr

Tae-Chon Ahn graduated and received the Ph. D. degree in electrical engineering from Yonsei University, Seoul, Korea, in 1978 and 1986, respectively. From 1981, he has worked in the Department of Electronics Convergence Engineering at

the Wonkwang University. He is now a professor in the same university. His research activities are focused on fuzzy systems, fuzzy neural networks, deep neural networks, computational intelligence, machine learning, and information data classifier.

E-mail: tcahn@wku.ac.kr