On the Comparison of Fuzzy Kernel Regression Estimator and Fuzzy Radial Basis Function Networks

Nimet YAPICI PEHLİVAN1*, Ayşen APAYDIN2

1Selçuk University, Faculty of Science and Art, Department of Statistics, 42031, Konya, Turkey
2Ankara University, Faculty of Science, Department of Statistics, 06100, Ankara, Turkey

Received: 28.09.2005 Revised: 02.01.2008 Accepted: 18.03.2008

ABSTRACT

In this paper, we suggest two fuzzy estimators in nonparametric regression: fuzzy kernel regression (FNPR) estimator and fuzzy radial basis function (FRBF) networks. Both FNPR estimator and FRBF networks are applied to original data taken from an experiment. We obtain MSE values of the FNPR estimator and FRBF networks and then compare them. We show that the FNPR estimator is more efficient than the FRBF networks.

Key Words: Fuzzy number, Fuzzy kernel regression estimator, Nonparametric regression, Neural networks, Fuzzy radial basis function networks.

1. INTRODUCTION

Regression analysis is one of the widely used statistical methods in research. In the least squares method, which is the most widely used one for estimating a regression model’s parameters, the assumptions of “normality” and “homoscedasticity” on error terms must be satisfied. When these assumptions are not satisfied, it is appropriate to use nonparametric regression (NPR) techniques [8]. Various methods have been used in NPR, such as kernel regression estimators, k-nearest neighbourhood estimators, spline smoothing, median smoothing, regressograms, etc. Kernel regression estimators are the most widely used estimators in NPR [11].

Neural networks can be valuable when the functional relationship between dependent and independent variables is not known. Radial basis function (RBF) networks form one of the essential types of neural networks and are used for control, signal processing pattern recognition and time series analysis [10].

The concept “fuzziness” was first proposed by the American philosopher Black in 1937. The fundamentals of Fuzzy Set Theory (FST) were constituted by Zadeh in 1965. FST is applied to the fields of operation research, management, sciences, artificial intelligence/expert systems, statistics and many other fields [14, 20].

Nonparametric regression relies on the experimenter to supply only qualitative information about the regression function. This qualitative information sometimes cannot be determined by precise number; in this case fuzzy numbers are employed. Cheng and Lee (2001) used RBF networks in fuzzy regression analysis without a predefined functional relationship between the input and the output. In that approach, only weights between the hidden unit and the output unit are considered as fuzzy numbers [3].

The main purpose of this study is to present the FNPR estimators and FRBF networks and to compare them. NPR are models which show the functional relationship between the dependent and the independent variables is not known but flexible.

In practice, the values of dependent and/or independent variables can not always be expressed by crisp numbers. In such cases, fuzzy numbers must be used. Therefore, in this study, at first we propose an FNPR estimator where both dependent and independent variables are fuzzy numbers. In the FNPR estimator, we used the Nadaraya-Watson kernel regression estimator which is the most commonly used type of NPR estimator. In the Nadaraya-Watson kernel regression estimator, dependent and independent variables are...
considered as fuzzy numbers and an FNPR estimator is constituted. Secondly, RBF Networks which is one of the types of neural networks are used as alternative methods to the Nadaraya-Watson estimator. In light of this idea, we proposed FRBF networks as an alternative method to the FNPR estimator. FRBF networks are RBF networks where all input, output and weights are fuzzy numbers.

This study is organized as follows: fuzzy arithmetic is considered in Section 2 and the FNPR estimator is given in Section 3. In Section 4, FRBF networks are suggested in detail. An application and results are given in Section 5. The conclusion part of this study is summarized in Section 6.

2. FUZZY ARITHMETIC

“Fuzziness” is the ambiguity that can be found in the definition of a concept or the meaning of a word. For example, the uncertainty in expressions like “old person”, “high temperature” or “small number” can be called fuzziness [18]. Let X be a universe whose generic elements are denoted by x and A be a classical subset of X. The membership function \( \mu_A(x) \) is the degree of membership of \( x \) in \( A \). If the grade of the membership is allowed to be in the set \( \{0,1\} \), then \( A \) is called “classical (crisp) set”. Nevertheless, if the grade of the membership is allowed to be in the interval \( [0,1] \), \( A \) is called “fuzzy set” and generally denoted by \( \tilde{A} \). For \( x \in X \) and \( \tilde{A} \subseteq X \), the membership function of the fuzzy set is denoted by \( \mu_A(x) : X \rightarrow [0,1] \). Fuzzy set theory is a general form of the crisp set [14].

Some basic concepts of the fuzzy set \( \tilde{A} \) are, for example, weak \( \alpha \)-level set ( \( \alpha \)-cut), strong \( \alpha \)-level set, convexity and normality. For a normal and convex fuzzy set, \( \forall \alpha \in [0,1] \) : the weak \( \alpha \)-level set is a closed interval, then it is called “fuzzy number”. Triangular and trapezoidal fuzzy numbers are among the most important ones of the several types of fuzzy numbers. Generally, triangular fuzzy numbers are denoted by \( \tilde{X}=(a,c,b) \), where “c” is a center, “a” is a lower and “b” is an upper limit of fuzzy number. A triangular fuzzy number is called a symmetrical triangular fuzzy number, if \( a=c-b \) [14, 18]. \( \alpha \)-level set of the fuzzy number \( \tilde{A} \) can be written as \( [A]_\alpha = \{ x \in A | \mu_A(x) \geq \alpha \} \). Since the \( \alpha \)-level sets of fuzzy numbers become closed intervals, \( [A]_\alpha \) is denoted by \( [A]_\alpha = \bigcup_{\alpha \in [0,1]} [A]_\alpha \). Arithmetic operations on \( \alpha \)-level set of fuzzy numbers are equal to the arithmetic operations on closed intervals [14].

If \( A=[a,b] \) and \( B=[d,e] \) are two intervals, then some arithmetic operations on intervals are given as follows [1, 13, 16].

i) \( A + B = [a,b] + [d,e] = [a+d, b+e] \)

ii) \( A - B = [a,b] - [d,e] = [a-e, b-d] \)

iii) \( A \cdot B = [\min(ad,ae,be,be), \max(ad,ae,be,be)] \)

iv) \( A / B = [\min(a/d,a/e,b/d,b/e), \max(a/d,a/e,b/d,b/e)] \)

v) \( k \cdot A = k[a,b] = [(ka,kb), k \geq 0] \)

vi) \( f (A) = f([a,b]) = [f(a), f(b)] \), when \( f \) is a continuous and monotonic non decreasing function.

vii) \( f (A) = f([a,b]) = [f(b), f(a)] \), when \( f \) is a continuous and monotonic non increasing function.

viii) \( D(A,B) = D([a,b],[c,d]) = \max |a-c|, |b-d| \)

where; \( D \) is a distance between two intervals.

ix) \( |A| = \max |[a,b]|, |A| = \max |[a,b]| \)

where; \( |A| \) is an absolute value of the interval \( [A] \).

3. FUZZY KERNEL REGRESSION ESTIMATOR

In this section, kernel regression estimators and fuzzy kernel regression estimators are introduced.

3.1. Kernel Regression Estimator

Regression analysis is a statistical technique for researching and modeling the relationship between the dependent and the independent variables. There is a functional relationship between these variables. If this relationship is unknown but flexible then it is called a “nonparametric regression model”. The NPR model can be defined as

\[ y_i = m(x_i) + \epsilon_i, \quad i = 1,2,\ldots,n \]  

(1)

In Eqn(1), “\( m \)” is an unknown regression function, “\( \epsilon_i \)” is an independent and identically distributed (i.i.d.) random variable with E(\( \epsilon_i \)) = 0 and Var(\( \epsilon_i \)) = \( \sigma^2 \) [6, 11]. Nonparametric regression estimator is defined as

\[ \hat{m}(x) = \sum_{i=1}^{n} W_{hi}(x) Y_i \]  

(2)

where; \( W_{hi}(x) \) denotes weights and h is a bandwidth parameter. If the weights \( W_{hi}(x) \) are based on kernel functions then NPR is called “kernel regression”. Kernel function \( K \) is a continuous, bounded and symmetrical real function. The most used kernel functions are uniform, normal, Epanechnikov and quartic kernel functions.

Several estimators have been used in kernel regression, such as the Nadaraya-Watson (NW) estimator, the Priestley-Chao estimator and the Gasser-Müller estimator [2, 5, 11]. The Nadaraya-Watson estimator is
defined as follows,
\[
\hat{m}_h(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - X_i) Y_i
\]
\[
= \frac{1}{n} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i
\]
\[
= \frac{1}{n} \sum_{i=1}^{n} \frac{K\left(\frac{x - X_i}{h}\right)}{h}
\]
\[
= \frac{1}{n} \sum_{i=1}^{n} \frac{K\left(\frac{x - X_i}{h}\right)}{h}
\]
In Eqn(3), \(K(.)\) is a kernel function and “\(h\)” is a bandwith parameter. The application of NPR always requires the choice of a bandwith parameter \(h\). The bandwith parameter \(h\) depends on the sample size \(n\). Usually, it is selected to balance the trade-off between variance and squared bias by minimizing some global measures of error [11, 17, 19].

3.2. Fuzzy Kernel Regression Estimator

In the fuzzy kernel regression estimator, the Nadaraya-Watson estimator is considered and fuzzified. The fuzzifying operation is performed by considering the dependent and the independent variables as symmetric triangular fuzzy numbers in the Nadaraya-Watson estimator. Independent and dependent variables are denoted by \(X_i = (x_i, t_i)\) and \(Y_i = (y_i, l_i)\), respectively. \(x\)-level sets of \(X_i\), \(Y_i\) and \(h\) are expressed as, \([X_i]_x = [X_i^L, X_i^U]\), \([Y_i]_x = [Y_i^L, Y_i^U]\) and \([h]_x = [h^L, h^U]\). We put their intervals instead of \(X_i\), \(Y_i\) and \(h\) in Eqn(3) and apply arithmetic operations on these intervals. As a result, FNPR estimator is obtained as follows;

\[
\hat{m}_h\left([X^L, X^U]\right) = \sum_{i=1}^{n} \frac{W_{hi}\left([X^L, X^U]\right)}{[Y_i^L, Y_i^U]} 
\]
\[
= \sum_{i=1}^{n} \left[ W_{hi}(x^L, x^U) [Y_i^L, Y_i^U] \right]
\]
\[
= \sum_{i=1}^{n} \left[ W_{hi}(x^L, x^U) [Y_i^L, Y_i^U] \right]
\]
\[
\frac{1}{n} \sum_{i=1}^{n} \left[ \frac{K\left(\frac{x^L - X_i^L}{h^L}\right)}{h^L} \right] Y_i^L + \frac{1}{n} \sum_{i=1}^{n} \left[ \frac{K\left(\frac{x^U - X_i^U}{h^U}\right)}{h^U} \right] Y_i^U
\]
\[
= \frac{1}{n} \sum_{i=1}^{n} \left[ \frac{K\left(\frac{x^L - X_i^L}{h^L}\right)}{h^L} \right] Y_i^L + \frac{1}{n} \sum_{i=1}^{n} \left[ \frac{K\left(\frac{x^U - X_i^U}{h^U}\right)}{h^U} \right] Y_i^U
\]

where; \(K(.)\) is a kernel function, \(h^L\) and \(h^U\) are the lower and the upper limits of bandwith parameter \(h\), respectively. \(X_i^L\) and \(X_i^U\) are the lower and upper limits of \(X_i\) and also \(Y_i^L\) and \(Y_i^U\) are the lower and the upper limits of \(Y_i\), respectively.

4. FUZZY RADIAL BASIS FUNCTION NETWORKS

In this section, RBF networks, FRBF networks and the training of FRBF networks are presented.

4.1. Radial Basis Function Networks

Radial basis function networks were developed by Broomhead&Lawe in 1988 and proposed as an alternative method to multi layer perceptrons [10]. The net input to the hidden layer is the distance from the input vector to the weight vector and computed as follows;

\[
h_{pj} = \exp\left(-\frac{\|X_{pi} - w_{ij}\|^2}{2\sigma_{pj}^2}\right)
\]

where; \(X_{pi}\) is an input vector, \(w_{ij}\)'s are weights between any input unit and hidden unit \(j\). \(\sigma_{pj}^2\) is a normalization factor for the hidden unit \(j\). The distance is usually computed in Euclidean metric. \(w_{ij}\)'s are called “RBF centers” and determined using “c-means clustering algorithm or fuzzy c-means clustering algorithm” [7]. Normalization factor \(\sigma_{pj}^2\)'s are determined by

\[
\sigma_{pj}^2 = \frac{1}{m} \sum_{p=1}^{m} \|X_{pi} - w_{ij}\|^2
\]

where; \(X_{pi}\) is a training pattern in the cluster, \(w_{ij}\)'s are the centers of the cluster associated with the hidden unit \(j\) and \(m\) is the number of training patterns in that cluster.

Output unit is calculated by

\[
\hat{Y}_p = \sum_{j=1}^{n} v_j h_{pj}
\]

where; \(v_j\)'s are the weights from the hidden unit \(j\) to the output unit [7]. In Eqn.s(5), (6) and (7), \(i\) \((i=1,2,\ldots,n_0)\) denotes the number of input units, \(p\) \((p=1,2,\ldots,n)\) shows the observation number and \(j\) \((j=1,2,\ldots,n_h)\) indicates the number of hidden units.

RBF networks are trained as hybrid networks. The weights \(w_{ij}\) and \(v_j\) and also the normalization factor
are updated by backpropagation algorithm. If the error criterion is minimum, then the process will be stopped; otherwise it must be repeated [4, 7].

4.2. Fuzzy Radial Basis Function Networks

Fuzzy radial basis function networks are RBF networks with fuzzy inputs and/or fuzzy outputs and/or fuzzy weights. We introduce the FRBF networks whose inputs $X_p$, outputs $Y_p$ and also weights $w_{ij}$ and $v_j$ are fuzzy. In the proposed FRBF Networks, the FCM algorithm is modified due to the fact that both $X_j$ and $w_{ij}$ are fuzzy numbers. The training algorithms of FRBF networks are constituted by fuzzifying Choi et al. (2003)’s RBF backpropagation algorithm and Ishibuchi et al. (1995)’s backpropagation algorithm.

$a$-level sets of fuzzy inputs $X_{pi}$ and fuzzy outputs $Y_p$ are expressed as $[X_{pi}]_a = [X^L_{pi}, X^U_{pi}]$ and $[Y_p]_a = [Y^L_p, Y^U_p]$, respectively. The weights between the input units and the hidden units are considered as symmetrical triangular fuzzy numbers and denoted by $w_{ij} = (w^L_{ij}, w^C_{ij}, w^U_{ij})$. Where, $w^C_{ij}$ is the lower limit, $w^C_{ij}$ is the center and $w^U_{ij}$ is the upper limit of $w_{ij}$. $a$-level sets of $w_{ij}$ are written as;

$$[W_{ij}]_a = \left[ W^L_{ij}, W^U_{ij} \right]$$

where;

$$[W^L_{ij}]_a = W^L_{ij} \left( 1 - \frac{a}{2} \right) + W^U_{ij} \left( \frac{a}{2} \right)$$

$$[W^U_{ij}]_a = W^L_{ij} \left( \frac{a}{2} \right) + W^U_{ij} \left( 1 - \frac{a}{2} \right)$$

The weights between the hidden units and the output unit are considered as symmetrical triangular fuzzy numbers and denoted by $v_j = (v^L_j, v^C_j, v^U_j)$. $a$-level sets of $V_j$ is written as;

$$[V_j]_a = \left[ v^L_j, v^U_j \right]$$

where;

$$[V^L_j]_a = v^L_j \left( 1 - \alpha/2 \right) + v^U_j \left( \alpha/2 \right)$$

$$[V^U_j]_a = v^L_j \left( \alpha/2 \right) + v^U_j \left( 1 - \alpha/2 \right)$$

Hidden unit $j$ is calculated by,

$$[h_{pj}]_a = \exp \left\{ -\frac{1}{2} \left( \frac{[X_{pi}]_a - [W_{ij}]_a}{[\sigma_{pj}]_a} \right)^2 \right\}$$

In Eqn(10), $p=1,2,...,n; i=1,2,...,n_i; j=1,2,...,n_h$, where, $[h_{pj}]_a$ is obtained as a crisp number because of arithmetic operations on the interval number.

The normalization factor of the hidden unit $j$ is determined by;

$$[\sigma^2_{pj}]_a = \frac{1}{m} \sum_{p=1}^{m} \left\| X_{pi} - [W_{ij}]_a \right\|^2$$

$$= \frac{1}{m} \left\{ \max \left\{ \left\| X_{pi}^L - [W_{ij}]_a \right\|, \left\| X_{pi}^U - [W_{ij}]_a \right\| \right\} \right\}^2$$

where; $[\sigma^2_{pj}]_a$ is a crisp number.

Fuzzy output unit for observation $p$ is calculated by;

$$\hat{Y}_p = \frac{1}{n_h} \sum_{j=1}^{n_h} \left[ V_j [h_{pj}]_a \right]_a$$

Let $Y_p$ be a fuzzy target corresponding to the fuzzy input $X_p$. The cost function for $a$-level sets of the fuzzy output $Y_p$ and the corresponding fuzzy target $Y_p$ is defined as follows [12].

$$E_{p,a} = E_{p,a}^L + E_{p,a}^U$$

where;

$$E_{p,a}^L = \frac{\alpha}{2} \left\{ Y_p^L - \hat{Y}_p \right\}^2$$

$$E_{p,a}^U = \frac{\alpha}{2} \left\{ Y_p^U - \hat{Y}_p \right\}^2$$
The cost function $E$ is obtained by:

$$E = \sum_{p=1}^{n} \sum_{i=1}^{s} E_{p, i},$$

(14)

### 4.3. The Training of the Fuzzy Radial Basis Function Networks

In the training of FRBF networks, we rearrange BP algorithms proposed by Choi et al. (2003) and Ishibuchi et al. (1995) for fuzzy numbers. The purpose of FRBF networks is to minimize the cost function $E$. Then we use this rearranged BP algorithm for the training of $v_j$, $w_{ij}$ and $\sigma^2_{ij}$.

The fuzzy weights $v_j$’s are updated by:

$$v_j^L(t+1) = v_j^L(t) + \Delta v_j^L(t)$$

(15)

$$v_j^U(t+1) = v_j^U(t) + \Delta v_j^U(t)$$

(16)

If $v_j^L > v_j^U$, then

$$v_j^L(t+1) = \min\{v_j^L(t+1), v_j^U(t+1)\}$$

$$v_j^U(t+1) = \max\{v_j^L(t+1), v_j^U(t+1)\}$$

In the Eqn(15)-(16), $\Delta v_j^L(t)$ and $\Delta v_j^U(t)$ are calculated by:

$$\Delta v_j^L(t) = -\eta \frac{\partial E_{p, i}}{\partial v_j^L} + \lambda \Delta v_j^L(t-1)$$

(17)

$$\Delta v_j^U(t) = -\eta \frac{\partial E_{p, i}}{\partial v_j^U} + \lambda \Delta v_j^U(t-1)$$

(18)

In the Eqn(17)-(18), $\frac{\partial E_{p, i}}{\partial v_j^L}$ and $\frac{\partial E_{p, i}}{\partial v_j^U}$ can be written as:

$$\frac{\partial E_{p, i}}{\partial v_j^L} = -\alpha \left( y_{lp}^L \frac{y_{pq}^L}{2} + y_{lp}^L \frac{y_{pq}^L}{2} \right) - \alpha \left( y_{lp}^L \frac{y_{pq}^L}{2} + y_{lp}^L \frac{y_{pq}^L}{2} \right)$$

The fuzzy weights $w_{ij}$’s are updated by:

$$w_{ij}^L(t+1) = w_{ij}^L(t) + \Delta w_{ij}^L(t)$$

(19)

$$w_{ij}^U(t+1) = w_{ij}^U(t) + \Delta w_{ij}^U(t)$$

(20)

If $w_{ij}^L > w_{ij}^U$ then

$$w_{ij}^L(t+1) = \min\{w_{ij}^L(t+1), w_{ij}^U(t+1)\}$$

$$w_{ij}^U(t+1) = \max\{w_{ij}^L(t+1), w_{ij}^U(t+1)\}$$

In the Eqn(19)-(20), $\Delta w_{ij}^L(t)$ and $\Delta w_{ij}^U(t)$ are calculated by:

$$\Delta w_{ij}^L(t) = -\eta \frac{\partial E_{p, i}}{\partial w_{ij}^L} + \lambda \Delta w_{ij}^L(t-1)$$

(21)

$$\Delta w_{ij}^U(t) = -\eta \frac{\partial E_{p, i}}{\partial w_{ij}^U} + \lambda \Delta w_{ij}^U(t-1)$$

(22)

In the Eqn(21)-(22), $\frac{\partial E_{p, i}}{\partial w_{ij}^L}$ and $\frac{\partial E_{p, i}}{\partial w_{ij}^U}$ can be written as:

$$\frac{\partial E_{p, i}}{\partial w_{ij}^L} = \frac{\partial E_{p, i}}{\partial w_{ij}^L} \frac{\partial w_{ij}^L}{\partial w_{ij}^L} + \frac{\partial E_{p, i}}{\partial w_{ij}^U} \frac{\partial w_{ij}^L}{\partial w_{ij}^U}$$

$$\frac{\partial E_{p, i}}{\partial w_{ij}^U} = \frac{\partial E_{p, i}}{\partial w_{ij}^L} \frac{\partial w_{ij}^L}{\partial w_{ij}^L} + \frac{\partial E_{p, i}}{\partial w_{ij}^U} \frac{\partial w_{ij}^U}{\partial w_{ij}^U}$$

where:

$$\frac{\partial E_{p, i}}{\partial w_{ij}^L}$$

and $\frac{\partial E_{p, i}}{\partial w_{ij}^U}$ can be written as follows:

i) If

$$\max\{x_{p}^L - y_{j}^L, y_{j}^L - x_{p}^U\} \left| x_{p}^L - y_{j}^L \right| = \left| x_{p}^L - y_{j}^L \right|$$

then

$$\frac{\partial E_{p, i}}{\partial w_{ij}^L} = -\alpha \left( y_{lp}^L \frac{y_{pq}^L}{2} + y_{lp}^L \frac{y_{pq}^L}{2} \right) - \alpha \left( y_{lp}^L \frac{y_{pq}^L}{2} + y_{lp}^L \frac{y_{pq}^L}{2} \right)$$

$$\frac{\partial E_{p, i}}{\partial w_{ij}^L} = -\alpha \left( y_{lp}^L \frac{y_{pq}^L}{2} + y_{lp}^L \frac{y_{pq}^L}{2} \right) - \alpha \left( y_{lp}^L \frac{y_{pq}^L}{2} + y_{lp}^L \frac{y_{pq}^L}{2} \right)$$

$$\frac{\partial E_{p, i}}{\partial w_{ij}^U} = 0$$
ii) If
\[
\max \left( \left| X_{pi}^{U} - [W_{ij}^{U}] \right|, \left| X_{pi}^{U} - [W_{ij}^{U}] \right| \right) = \left| X_{pi}^{U} - [W_{ij}^{U}] \right|
\]
then
\[
\frac{\partial E_{p,\alpha}}{\partial W_{ij}^{U}} = 0
\]

\[
\frac{\partial E_{p,\alpha}}{\partial W_{ij}^{U}} = \alpha \left[ \left( Y_{p[k_i]} - \hat{Y}_{p[k_i]} \right) h_{pj}(\sigma_{pj})^{-2} \right] \left| X_{pi}^{U} - [W_{ij}^{U}] \right|
\]

\[
- \alpha \left( Y_{p[k_i]} - \hat{Y}_{p[k_i]} \right) h_{pj}(\sigma_{pj})^{-2} \left| X_{pi}^{U} - [W_{ij}^{U}] \right|
\]

The normalization factors \( \sigma_{pj}^2 \)'s are updated by
\[
\sigma_{pj}(t+1) = \sigma_{pj}(t) + \Delta \sigma_{pj}(t)
\]

In the Eqn (23), \( \Delta \sigma_{pj}(t) \) is calculated by;
\[
\Delta \sigma_{pj}(t) = -\eta \frac{\partial E_{p,\alpha}}{\partial \sigma_{pj}} + \lambda \Delta \sigma_{pj}(t-1)
\]

In the Eqn (24), \( \frac{\partial E_{p,\alpha}}{\partial \sigma_{pj}} \) can be written as follows;
\[
\frac{\partial E_{p,\alpha}}{\partial \sigma_{pj}} = \zeta^L + \zeta^U
\]

where; \( \zeta^L \) and \( \zeta^U \) are defined as;

i) If
\[
\max \left( \left| X_{pi}^{L} - [W_{ij}^{L}] \right|, \left| X_{pi}^{L} - [W_{ij}^{L}] \right| \right) = \left| X_{pi}^{L} - [W_{ij}^{L}] \right|
\]
then
\[
\zeta^L = -\alpha \left( Y_{p[k_i]} - \hat{Y}_{p[k_i]} \right) h_{pj}(\sigma_{pj})^{-3} \left| X_{pi}^{L} - [W_{ij}^{L}] \right|
\]
\[
\zeta^U = -\alpha \left( Y_{p[k_i]} - \hat{Y}_{p[k_i]} \right) h_{pj}(\sigma_{pj})^{-3} \left| X_{pi}^{L} - [W_{ij}^{L}] \right|
\]

ii) If
\[
\max \left( \left| X_{pi}^{U} - [W_{ij}^{U}] \right|, \left| X_{pi}^{U} - [W_{ij}^{U}] \right| \right) = \left| X_{pi}^{U} - [W_{ij}^{U}] \right|
\]
then
\[
\zeta^L = -\alpha \left( Y_{p[k_i]} - \hat{Y}_{p[k_i]} \right) h_{pj}(\sigma_{pj})^{-3} \left| X_{pi}^{U} - [W_{ij}^{U}] \right|
\]
\[
\zeta^U = -\alpha \left( Y_{p[k_i]} - \hat{Y}_{p[k_i]} \right) h_{pj}(\sigma_{pj})^{-3} \left| X_{pi}^{U} - [W_{ij}^{U}] \right|
\]

In Eqn.s (17), (18), (21), (22) and (24), \( \eta \) is a learning constant, \( \lambda \) is a momentum constant and \( t \) shows the number of repetitions.

An algorithm for the training of FRBF networks is given as follows;

Step 1: Fuzzy weights \( w_{ij} \)'s are determined by a modified FCM algorithm. Fuzzy weights \( v_{j} \)'s are initialized to random fuzzy number.

Step 2: Step 3 must be repeated for \( \alpha \)-level sets such as \( \alpha_1, \alpha_2, ..., \alpha_n \).

Step 3: The following i), ii), iii) and iv) must be repeated for \( p=1,2,...,n \).

i) Fuzzy weights \( w_{ij} \)'s are calculated by Eqn.(10)-(13).

ii) Fuzzy weights \( v_{j} \)'s are updated by Eqn.(15)-(18).

iii) Fuzzy weights \( w_{ij} \)'s are updated by Eqn.(19)-(22).

iv) Normalization factors \( \sigma_{pj}^2 \)'s are updated by Eqn.(23)-(24).

Step 4: If the iteration number reaches to a predefined value and then stop. Otherwise go to Step2.

5. APPLICATION

We have applied the FNPR estimator and the FRBF Networks to the data of the “Turkish Pastrami” production experiment. In this experiment, chemical, microbiological and organoleptic analyses are made. Pastrami production is made in three trials. In the production stage, measurements are taken on the 1st, 7th, 15th, 30th and 60th days. As a result of the chemical analysis, amounts of humidity, salt, value of pH and value of water activity(\( a_w \)) are determined. As a result of microbiological analysis, total number of mezophil aerob microorganism(PCA), number of staphylococcus-micrococcus microorganism(MSA), number of lactobacillus microorganism(RA) and number of yeast and mold microorganisms(PDA) are determined. As a result of organoleptic analysis, colour, taste, appearance and texture are evaluated by a test panel of six persons [9].

In this study, the functional relationship between chemical factors and microbiological factors and also the functional relationship between chemical factors and organoleptic factors are estimated. The efficiencies of these factors on pastrami production are evaluated. Independent variables are taken as salt, pH and water activity(\( a_w \)). Dependent variables are taken as PCA, MSA, RA, PDA, taste, texture and color.

Errors are computed by using mean squared error(MSE) defined in terms of the difference between two fuzzy numbers suggested by Ishibuchi et.al.(1995) and Lin et.al.(2000) as follows;

\[
\text{MSE} = \frac{1}{n} \sum_{p=1}^{n} \sum_{i=1}^{q} \left( (\hat{Y}_{p[k_i]}^{L} - Y_{p[k_i]}^{L})^2 + (\hat{Y}_{p[k_i]}^{U} - Y_{p[k_i]}^{U})^2 \right)
\]
The general functional structure between the dependent and independent variables is defined as \( Y = f(X) \). Then, fifteen different functions are constituted and fuzzy intervals are formed by computing the values of mean ± standard deviation. Means and standard deviations of the measurements are taken at the end of three trials of the “Turkish Pastrami” experiment. Fuzzy estimates are from the FNPR estimator and the FRBF networks after forming fuzzy intervals for all fuzzy dependent and independent variables. MSE values belonging to 15 different models are evaluated by the FNPR estimator and the FRBF networks.

These 15 models are,

- \( PCA = f(Salt) \), \( MSA = f(Salt) \)
- \( RA = f(Salt) \), \( PDA = f(Salt) \)
- \( Taste = f(Salt) \)
- \( PCA = f(pH) \), \( MSA = f(pH) \)
- \( RA = f(pH) \), \( PDA = f(pH) \)
- \( Texture = f(pH) \)
- \( PCA = f(a_0) \), \( MSA = f(a_0) \)
- \( RA = f(a_0) \), \( PDA = f(a_0) \)
- \( Color = f(a_0) \)

Bandwidth parameters of FNPR estimator are determined by XPLORE package programme for each model and given in Table 1.

Table 1. The values of bandwidth parameter \( h \) for FNPR estimator

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter ( h )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( PCA = f(Salt) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( MSA = f(Salt) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( RA = f(Salt) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( PDA = f(Salt) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( Taste = f(Salt) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( PCA = f(pH) )</td>
<td>[0.15, 0.15]</td>
</tr>
<tr>
<td>( MSA = f(pH) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( RA = f(pH) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( PDA = f(pH) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( Texture = f(pH) )</td>
<td>[10, 10]</td>
</tr>
<tr>
<td>( PCA = f(a_0) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( MSA = f(a_0) )</td>
<td>[0.01, 0.01]</td>
</tr>
<tr>
<td>( RA = f(a_0) )</td>
<td>[0.99, 0.99]</td>
</tr>
<tr>
<td>( PDA = f(a_0) )</td>
<td>[0.07, 0.07]</td>
</tr>
<tr>
<td>( Color = f(a_0) )</td>
<td>[0.01, 0.01]</td>
</tr>
</tbody>
</table>

FRBF networks with a single input unit, three hidden units and a single output unit are constituted for considered data. These networks were trained by the proposed training algorithm given in Section 4. In the training algorithm, the learning constant is \( \eta = 0.01 \) and the momentum constant is \( \lambda = 0.1 \).

Beginning values of weights \( w_{ij} \)’s between input-hidden units are determined by a modified FCM algorithm. In this algorithm, \( w_{ij} \)’s are calculated by independent variables taken as Salt, pH and \( a_0 \), respectively. Beginning values of normalization factor \( \sigma_j \)’s of hidden units are calculated by Eqn(6). Beginning values of weights \( v_{ij} \)’s between hidden-output units are determined as arbitrary fuzzy numbers for all hidden units (\( j = 1, 2, 3 \)). The values of \( w_{ij} \), \( \sigma_j \) and \( v_{ij} \)’s are summarized in Table 2.

Table 2. The values of weights and normalization factor for FRBF Networks

<table>
<thead>
<tr>
<th>Model</th>
<th>The weights ( w_{ij} ) between input-hidden unit</th>
<th>The weights ( v_{ij} ) between hidden-output unit</th>
<th>The normalization factor ( \sigma_j )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( PCA = f(Salt) )</td>
<td>( w_{1j} = [5.99, 4.00, 5.84] ) ( \sigma_j = [1.57, 1.68, 1.12] )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( MSA = f(Salt) )</td>
<td>( w_{2j} = [7.87, 6.72, 6.20] ) ( \sigma_j = [3, 3, 2] )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( RA = f(Salt) )</td>
<td>( w_{3j} = [5.94, 5.70, 5.63] ) ( \sigma_j = [0.28, 0.17, 0.18] )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( PDA = f(Salt) )</td>
<td>( w_{4j} = [5.80, 5.66, 5.45] ) ( \sigma_j = [1, 2, 1] )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( Taste = f(Salt) )</td>
<td>( w_{5j} = [5.84, 5.70, 5.63] ) ( \sigma_j = [3, 3, 2] )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( PCA = f(pH) )</td>
<td>( w_{1j} = [0.90, 0.81, 0.77] ) ( \sigma_j = [0.09, 0.04, 0.07] )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( MSA = f(pH) )</td>
<td>( w_{2j} = [0.90, 0.83, 0.81] ) ( \sigma_j = [3, 3, 2] )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( RA = f(pH) )</td>
<td>( w_{3j} = [0.99, 0.99, 0.99] ) ( \sigma_j = [0.01, 0.01] )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( PDA = f(pH) )</td>
<td>( w_{4j} = [0.07, 0.07, 0.07] ) ( \sigma_j = [0.01, 0.01] )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Estimated values of dependent variables for each model are calculated by FNPR estimator and FRBF networks. Then real values and estimated values are given in detail in [19]. MSE values of FNPR estimator and FRBF networks are shown in Table 3.

By investigating Table 3, it can be seen that the MSE values of the FNPR estimator are less than the MSE values of FRBF networks for all models. Therefore, the FNPR estimator is a more efficient method than the FRBF networks for these data.
Therefore, it can be concluded that the FNPR estimator are less than the MSE values of FRBF from the results in Table 3. MSE values of the FNPR FRBF networks are given in Table 3. As can be seen networks. MSE values of the FNPR estimator and output units were summarized in Table 2 for FRBF beginning values of weights’s between hidden-input units, beginning values of normalization factor’s of hidden units and beginning values of weights’s between hidden-output units were summarized in Table 2 for FRBF networks. MSE values of the FNPR estimator and FRBF networks are given in Table 3. As can be seen from the results in Table 3, MSE values of the FNPR estimator are less than the MSE values of FRBF networks. Therefore, it can be concluded that the FNPR estimator is more efficient than the FRBF networks for these data.

Table 3. MSE values regarding fuzzy estimates evaluated by FNPR estimator and FRBF networks

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE_{\text{FNPR}}</th>
<th>MSE_{\text{FRBF}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA = f(Salt)</td>
<td>0.0024</td>
<td>0.1973</td>
</tr>
<tr>
<td>MSA = f(Salt)</td>
<td>0.0791</td>
<td>1.2612</td>
</tr>
<tr>
<td>RA = f(Salt)</td>
<td>0.2382</td>
<td>12.5177</td>
</tr>
<tr>
<td>PDA = f(Salt)</td>
<td>0.0720</td>
<td>3.6713</td>
</tr>
<tr>
<td>Taste = f(Salt)</td>
<td>0.0071</td>
<td>0.1397</td>
</tr>
<tr>
<td>PCA = f(ph)</td>
<td>0.1270</td>
<td>0.2707</td>
</tr>
<tr>
<td>MSA = f(ph)</td>
<td>0.4701</td>
<td>1.2754</td>
</tr>
<tr>
<td>RA = f(ph)</td>
<td>2.1372</td>
<td>12.5428</td>
</tr>
<tr>
<td>PDA = f(ph)</td>
<td>1.1405</td>
<td>3.9400</td>
</tr>
<tr>
<td>Texture = f(ph)</td>
<td>0.1885</td>
<td>0.4029</td>
</tr>
<tr>
<td>PCA = f(a_1)</td>
<td>0.0864</td>
<td>0.2711</td>
</tr>
<tr>
<td>MSA = f(a_1)</td>
<td>0.2756</td>
<td>1.2766</td>
</tr>
<tr>
<td>RA = f(a_1)</td>
<td>6.6424</td>
<td>12.4789</td>
</tr>
<tr>
<td>PDA = f(a_1)</td>
<td>2.6516</td>
<td>3.9400</td>
</tr>
<tr>
<td>Color = f(a_1)</td>
<td>0.1749</td>
<td>0.2265</td>
</tr>
</tbody>
</table>

6. CONCLUSION

In practice, the values of dependent and/or independent variables can not always be expressed by crisp numbers. In such cases, fuzzy numbers must be used. Therefore, in this study, we proposed the fuzzy kernel regression estimator and also the fuzzy radial basis function networks. First of all, we proposed a FNPR estimator where both dependent and independent variables are fuzzy numbers. In the FNPR estimator, we used the Nadaraya-Watson kernel regression estimator. In a Nadaraya-Watson kernel regression estimator, dependent and independent variables are considered as fuzzy numbers and the FNPR estimator is constituted. The bandwidth parameters of each model are given in Table 1. Secondly, we proposed FRBF networks as an alternative method to the FNPR estimator. FRBF networks are RBF networks where all input, output and weights are fuzzy numbers. Beginning values of weights $w_j$’s between input-hidden units, beginning values of normalization factor $\sigma_j$’s of hidden units and beginning values of weights $v_j$’s between hidden-output units were summarized in Table 2 for FRBF networks. MSE values of the FNPR estimator and FRBF networks are given in Table 3. As can be seen from the results in Table 3, MSE values of the FNPR estimator are less than the MSE values of FRBF networks. Therefore, it can be concluded that the FNPR estimator is more efficient than the FRBF networks for these data.

ACKNOWLEDGMENT

This study was supported by Selçuk University, Coordinating Office of Science Research Projects with project number 2002/020. We thank the anonymous referees for their helpful and constructive comments on the previous version of the manuscript which improved the presentation of this article.

REFERENCES


