Support Vector Regression with Fuzzy Target Output

Yahya Forghani,
Computer Department,
Mashhad Branch, Azad University,
Mashhad, Iran
yahyafor2000@yahoo.com

Hadi Sadoghi Yazdi
Computer department
Ferdowsi university of mashhad
Mashhad, Iran
h-sadoghi@um.ac.ir

Sohrab Effati
Applied Mathematics department
Ferdowsi university of mashhad
Mashhad, Iran
s-effati@um.ac.ir

Abstract—**In this paper, we incorporate the concept of fuzzy set theory into the support vector regression (SVR). In our proposed method, target outputs of training samples are considered to be fuzzy numbers and then, membership function of actual output (objective hyperplane in high dimensional feature space) is obtained.**

Two main properties of our proposed method are: (1) membership function of actual output can be obtained without pre-assumption on type of membership function of the bias term and the components of weight vector; (2) the membership function of target output can be each type of fuzzy number.

Keywords-Fuzzy target output; Fuzzy weight; Fuzzy bias; Support vector regression (SVR).

I. Introduction

Regression analysis is a useful estimation method. This analysis is performed to evaluate the functional relationship between input (independent or explanatory) and output (dependent or response) variables, thus assuming that the difference between the observed and estimated dependent variables is due to random errors. Traditionally, the observations are assumed to be crisp.

In many real-world applications, available information is often uncertain, imprecise, and incomplete and thus usually is represented by fuzzy sets or a generalization of interval data. For handling interval data, fuzzy regression analysis is an important tool and has been successfully applied in different applications such as market forecasting [1] and system identification [2]. Fuzzy regression, first developed by [3] in a linear system, is based on the extension principle. In the experiments that followed this pioneering effort, [4] used fuzzy input experimental data to build fuzzy regression models. A technique for linear least squares fitting of fuzzy variables was developed by [5, 6], giving the solution to an analog of the normal equation of classical least squares. A collection of relevant papers dealing with several approaches to fuzzy regression analysis can be found in [7]. In contrast to the fuzzy linear regression, there have been only a few articles on fuzzy nonlinear regression [8-16]. In this paper, we discuss multivariate fuzzy nonlinear regression by support vector machine. Support vector regression (SVR) [17-19] is used in the function estimation and time-series prediction applications.

In [13], the concept of fuzzy set theory was incorporated into the SVR model. They proposed three models in their paper. In their first model, the bias term, the target output y_i, and the components of input x_i, were considered to be an especial type of fuzzy numbers, namely triangular fuzzy numbers. Pre-assumption on type of fuzzy target output and fuzzy input is a limitation and pre-assumption on type of the fuzzy bias term makes their model inaccurate. Indeed, membership function of the fuzzy bias term must be obtained according to fuzzy training data (input and target output). Moreover, the components of the weight vector, and the slack variables were considered to be crisp which is another limitation of this model. In their second model, the bias term were considered to be crisp. Thus, this model is also an especial case of the first model and has the limitations of the first model. Moreover, the first and the second models were solved only in the input space. Therefore, their model can be used only to estimate a linear function according to training data. In their third model, which was solved in high dimensional feature space, only the bias term and components of the weight vector were considered to be fuzzy numbers. In other words, the target output y_i and the component of input x_i were considered to be crisp.

In [8], the concept of fuzzy set theory was also incorporated into the SVR model. In their proposed model, the bias term, the components of weight vector and the target output data y_i, were considered to be triangular fuzzy numbers. Again, the pre-assumption on the type of fuzzy target output is a limitation and pre-assumption on type of the fuzzy bias term and the components of weight vector makes their model inaccurate.

In our proposed method, the target output is considered to be an optional type of fuzzy number and the membership function of actual output (objective hyperplane in high dimensional feature space), the slack variables, the bias term and the components of weight vector are obtained based on Liu's method [20].

Organization of this paper is as follows: In section 2, will be paid to some preliminaries and in section 3, our novel method will be explained. Section 4 shows experimental results. Finally, section 5 concludes the paper.

II. PRILIMINARIES

Consider the following training set:

$$\{(x_i, \tilde{y}_i), i = 1, \dots, n\},$$

where n is the number of training samples. Formulation of SVR model is as follows:

$$J = min\ \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \hat{\xi}_i)$$

$$s.t. \begin{cases} y_i - w^T g(x_i) - b \le \varepsilon + \xi_i\ , \ \ i = 1, \dots, n; \\ w^T g(x_i) + b - y_i \le \varepsilon + \hat{\xi}_i\ , \ \ i = 1, \dots, n; \\ \xi_i, \hat{\xi}_i \ge 0\ , \ \ i = 1, \dots, n. \end{cases} \tag{1}$$

where $\xi = (\xi_1, \dots, \xi_n)^T$; $\hat{\xi} = (\hat{\xi}_1, \dots, \hat{\xi}_n)^T$; ξ_i and $\hat{\xi}_i$ are slack variables of i-th training sample; C is a penalty term, and $g(.)$ is a mapping function that maps the input space to a high dimensional feature space. The constraints of this program allow a deviation between the target output y_i and the value of the approximated function, $f(x) = w^T g(x) + b$. The slack variable ξ_i is used for exceeding the output value by more than ε and $\hat{\xi}_i$ for being more than ε below the output value. The penalty term C determines the trade-off between the magnitude of the margin and the estimation error of training data.

III. OUR PROPOSED METHOD

A. Problem Definition

Consider the following training set:

$$\{(x_i, \tilde{y}_i), i = 1, \dots, n\}.$$

Suppose that the target outputs are approximately known and can be represented by fuzzy numbers $\tilde{y}_i$ $(i = 1, \dots, n)$. Let $\mu_{\tilde{y}_i}$ $(i = 1, \dots., n)$ denote their membership functions. We have

$$\tilde{y}_i = \{(y_i, \mu_{\tilde{y}_i}) | y_i \epsilon S(\tilde{y}_i)\},$$

where $S(\tilde{y}_i)$ is the support of $\tilde{y}_i$. Formulation of SVR for such training data is as follows:

$$\tilde{J} = min\ \frac{1}{2} \tilde{w}^T \tilde{w} + C \sum_{i=1}^{n} (\tilde{\xi}_i + \tilde{\hat{\xi}}_i)$$

$$s.t. \begin{cases} \tilde{y}_i - \tilde{w}^T g(x_i) - \tilde{b} \le \varepsilon + \tilde{\xi}_i, \\ \tilde{w}^T g(x_i) + \tilde{b} - \tilde{y}_i \le \varepsilon + \tilde{\hat{\xi}}_i, \\ \tilde{\xi}_i, \tilde{\hat{\xi}}_i \ge 0\ , \ \ i = 1, \dots, n. \end{cases} \tag{2}$$

where $\tilde{w}$ is fuzzy weight vector; $\tilde{b}$ is fuzzy bias and $\tilde{\xi}_i$ and $\tilde{\hat{\xi}}_i (i = 1, \dots, n)$ are fuzzy slack variables. Without loss of generality, $\tilde{y}_i$ $(i = 1, \dots, n)$ are assumed to be fuzzy numbers. Based on the extension principle [21], we have

$$\mu_{\tilde{J}}(j) = sup_y\ min\{\mu_{\tilde{y}_i}(y_i), \forall i | j = J(y)\}, \tag{3}$$

where $J(y)$ is the function of the program (1) and $y = (y_1, \dots, y_n)^T$ is its parameter. To drive $\mu_{\tilde{J}}$ using Eq. (3) is hardly possible. To find the membership function $\mu_{\tilde{J}}$, it suffices to find the right shape function and left shape function of $\mu_{\tilde{J}}$, which is equivalent to find the upper bound and the lower bound of objective function $\tilde{J}$ at each α-cut, named $\tilde{J}_\alpha^U$ and $\tilde{J}_\alpha^L$, respectively [20]. These bounds can be determined from the following two-level mathematical programming models:

$$\tilde{J}_\alpha^L = \min_y \begin{cases} min_{w,b,\xi,\hat{\xi}}\ \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \hat{\xi}_i) \\ s.t. \begin{cases} y_i - w^T g(x_i) - b \le \varepsilon + \xi_i, \\ w^T g(x_i) + b - y_i \le \varepsilon + \hat{\xi}_i, \\ \xi_i, \hat{\xi}_i \ge 0\ , \ \ i = 1, \dots, n. \end{cases} \end{cases} \tag{4}$$

$$s.t. (y_i)_\alpha^L \le y_i \le (y_i)_{\boxed{?}}^U, \ \ i = 1, \dots, n.$$

$$\tilde{J}_\alpha^U = \max_y \begin{cases} min_{w,b,\xi,\hat{\xi}}\ \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \hat{\xi}_i) \\ s.t. \begin{cases} y_i - w^T g(x_i) - b \le \varepsilon + \xi_i, \\ w^T g(x_i) + b - y_i \le \varepsilon + \hat{\xi}_i, \\ \xi_i, \hat{\xi}_i \ge 0\ , \ \ i = 1, \dots, n. \end{cases} \end{cases} \tag{5}$$

$$s.t. (y_i)_\alpha^L \le y_i \le (y_i)_\alpha^U, \ \ i = 1, \dots, n.$$

B. Solving the Lower Bound Program

The program (4) can be restated as follows:

$$min_{w,b,\xi,\hat{\xi}}\ \frac{1}{2} w^T w + C \sum_{i=1}^{n} \left(\xi_i + \hat{\xi}_i\right)$$

$$s.t. \begin{cases} y_i - w^T g(x_i) - b \le \varepsilon + \xi_i\ , \ \ i = 1, \dots, n; \\ w^T g(x_i) + b - y_i \le \varepsilon + \hat{\xi}_i\ , \ \ i = 1, \dots, n; \\ \xi_i, \hat{\xi}_i \ge 0\ , \ \ i = 1, \dots, n; \\ (y_i)_\alpha^L \le y_i \le (y_i)_\alpha^U, \ \ i = 1, \dots, n. \end{cases} \tag{6}$$

The Lagrangian dual form of the program (6) is as follows:

$$max_{\delta,\gamma,\theta,\hat{\delta},\hat{\gamma},\hat{\theta}}\ L(\delta, \gamma, \theta, \hat{\delta}, \hat{\gamma}, \hat{\theta})$$

$$subject\ to\ \delta_i, \gamma_i, \theta_i, \hat{\delta}_i, \hat{\gamma}_i, \hat{\theta}_i \ge 0, i = 1, \dots, n,$$

where $\delta = (\delta_1, \dots, \delta_n)^T$, $\hat{\delta} = (\hat{\delta}_1, \dots, \hat{\delta}_n)^T$, $\gamma = (\gamma_1, \dots, \gamma_n)^T$, $\hat{\gamma} = (\hat{\gamma}_1, \dots, \hat{\gamma}_n)^T$, $\theta = (\theta_1, \dots, \theta_n)^T$, $\hat{\theta} = (\hat{\theta}_1, \dots, \hat{\theta}_n)^T$ and

$$L\left(\delta, \gamma, \theta, \hat{\delta}, \hat{\gamma}, \hat{\theta}\right) = inf \left\{\frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \hat{\xi}_i)\right.$$
$$- \sum_{i=1}^{n} \delta_i \left(\varepsilon + \xi_i - y_i + w^T g(x_i) + b\right)$$
$$- \sum_{i=1}^{n} \hat{\delta}_i \left(\varepsilon + \hat{\xi}_i + y_i - w^T g(x_i) - b\right)$$
$$- \sum_{i=1}^{n} \gamma_i\, \xi_i - \sum_{i=1}^{n} \hat{\gamma}_i\, \hat{\xi}_i$$
$$\left. - \sum_{i=1}^{n} \theta_i\, ((y_i)_\alpha^U - y_i) - \sum_{i=1}^{n} \hat{\theta}_i\, (y_i - (y_i)_\alpha^L)\right\}.$$

For the optimal solution, the following conditions are satisfied:

$$\frac{\partial L}{\partial w} = 0 \to w = \sum_{i=1}^{n} (\delta_i - \hat{\delta}_i) g(x_i), \tag{7}$$

$$\frac{\partial L}{\partial b} = 0 \to \ \sum_{i=1}^{n} (\delta_i - \hat{\delta}_i) = 0, \tag{8}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \to \ \delta_i = C - \gamma_i\ , \ i = 1, \dots, n, \tag{9}$$

$$\frac{\partial L}{\partial \hat{\xi}_i} = 0 \to \ \hat{\delta}_i = C - \hat{\gamma}_i\ , \ i = 1, \dots, n, \tag{10}$$

$$\frac{\partial L}{\partial y_i} = 0 \to \ \delta_i - \hat{\delta}_i + \theta_i - \hat{\theta}_i = 0, \ i = 1, \dots, n, \tag{11}$$

$$\delta_i\left(\varepsilon + \xi_i - y_i + w^T g(x_i) + b\right) = 0, \ \ i = 1, \dots, n, \tag{12}$$

$$\hat{\delta}_i\left(\varepsilon + \hat{\xi}_i + y_i - w^T g(x_i) - b\right) = 0, \ \ i = 1, \dots, n, \tag{13}$$

$$\gamma_i \xi_i = 0,\ \ i = 1, \dots, n, \tag{14}$$

$$\hat{\gamma}_i \hat{\xi}_i = 0,\ \ i = 1, \dots, n, \tag{15}$$

$$\theta_i ((y_i)^U_{\boxed{?}} - y_i) = 0,\ \ i = 1, \dots, n, \tag{16}$$

$$\hat{\theta}_i (y_i - (y_i)^L_\alpha) = 0,\ \ i = 1, \dots, n, \tag{17}$$

$$\delta_i, \gamma_i, \theta_i, \hat{\delta}_i, \hat{\gamma}_i, \hat{\theta}_i \geq 0,\ \ i = 1, \dots, n. \tag{18}$$

Using the above conditions, $L(\delta, \gamma, \theta, \hat{\delta}, \hat{\gamma}, \hat{\theta})$ is transformed into

$$\sum_{i=1}^n \hat{\theta}_i\, (y_i)^L_\alpha - \sum_{i=1}^n \theta_i (y_i)^U_\alpha - \varepsilon \sum_{i=1}^n (\hat{\delta}_i + \delta_i)$$
$$-\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \big(\delta_i - \hat{\delta}_i\big)\big(\delta_j - \hat{\delta}_j\big) K(x_i, x_j)\,,$$

where $K(x_i, x_j) = g(x_i)^T g(x_j)$ is a kernel function. Since $\delta_i, \gamma_i \geq 0$, from Eq. (9) we have $0 \leq \delta_i \leq C$. Also, since $\hat{\delta}_i, \hat{\gamma}_i \geq 0$, from Eq. (10) we have $0 \leq \hat{\delta}_i \leq C$. Therefore, the Lagrangian dual form of the program (6) becomes as follows:

$$max_{\delta,\theta,\hat{\delta},\hat{\theta}} \sum_{i=1}^n \hat{\theta}_i\, (y_i)^L_\alpha - \sum_{i=1}^n \theta_i (y_i)^U_\alpha$$
$$-\varepsilon \sum_{i=1}^n (\hat{\delta}_i + \delta_i)$$
$$-\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \big(\delta_i - \hat{\delta}_i\big)\big(\delta_j - \hat{\delta}_j\big) K(x_i, x_j)$$
$$s.t. \begin{cases} \sum_{i=1}^n (\delta_i - \hat{\delta}_i) = 0\,, \\ \delta_i - \hat{\delta}_i + \theta_i - \hat{\theta}_i = 0,\ \ i = 1, \dots, n, \\ 0 \leq \delta_i \leq C,\ \ i = 1, \dots, n, \\ 0 \leq \hat{\delta}_i \leq C,\ \ i = 1, \dots, n, \\ \theta_i \geq 0,\ \ i = 1, \dots, n, \\ \hat{\theta}_i \geq 0,\ \ i = 1, \dots, n, \end{cases}$$

which is a conventional quadratic program. From Eq. (12), if $\delta_i > 0$, $y_i - w^T g(x_i) - b = \varepsilon + \xi_i$. From Eq. (9), if $\delta_i < C$, $\gamma_i > 0$, and from Eq. (14), if $\gamma_i > 0$, $\xi_i = 0$. Thus, if $0 < \delta_i < C$,

$$y_i - w^T g(x_i) - b = \varepsilon, \tag{19}$$

where from Eq. (16), if $\theta_i > 0$, $y_i = (y_i)^U_\alpha$ and from Eq. (17) if $\hat{\theta}_i > 0$, $y_i = (y_i)^L_\alpha$.

Again, from Eq. (13), if $\hat{\delta}_i > 0$, $-y_i + w^T g(x_i) + b = \varepsilon + \hat{\xi}_i$. From Eq. (10), if $\hat{\delta}_i < C, \tilde{\gamma}_i > 0$, and from Eq. (15) if $\tilde{\gamma}_i > 0$, $\hat{\xi}_i = 0$. Thus, if $0 < \hat{\delta}_i < C$,

$$-y_i + w^T g(x_i) + b = \varepsilon. \tag{20}$$

where from Eq. (16), if $\theta_i > 0$, $y_i = (y_i)^U_\alpha$ and from Eq. (17), if $\hat{\theta}_i > 0, y_i = (y_i)^L_\alpha$.

From Eq. (7), the approximated hyperplane or actual output is given by

$$f(x) = w^T g(x) + b = \sum_{i=1}^n (\delta_i - \hat{\delta}_i) g(x_i)^T g(x) + b$$
$$= \sum_{i=1}^n (\delta_i - \hat{\delta}_i) K(x_i, x) + b\,,$$

where from Eq. (20) and Eq. (21) the bias satisfies

$$b = (y_i)^U_{\boxed{?}} - w^T g(x_i) - \varepsilon, \text{ for } 0 < \delta_i < C \text{ and } \theta_i > 0, \tag{21}$$

$$b = (y_i)^L_\alpha - w^T g(x_i) - \varepsilon, \text{ for } 0 < \delta_i < C \text{ and } \hat{\theta}_i > 0, \tag{22}$$

$$b = (y_i)^U_\alpha - w^T g(x_i) + \varepsilon, \text{ for } 0 < \hat{\delta}_i < C \text{ and } \theta_i > 0, \tag{23}$$

$$b = (y_i)^L_\alpha - w^T g(x_i) + \varepsilon, \text{ for } 0 < \hat{\delta}_i < C \text{ and } \hat{\theta}_i > 0. \tag{24}$$

In calculating bias, to avoid calculation errors, we average biases that satisfy Eq. (21)-(24).

C. *Solving the Upper Bound Program*

Consider the inner level of the upper bound program (5):

$$min_{w,b,\xi,\hat{\xi}}\ \frac{1}{2} w^T w + C \sum_{i=1}^n (\xi_i + \hat{\xi}_i)$$
$$s.t. \begin{cases} y_i - w^T g(x_i) - b \leq \varepsilon + \xi_i\,, \\ w^T g(x_i) + b - y_i \leq \varepsilon + \hat{\xi}_i\,, \\ \xi_i, \hat{\xi}_i \geq 0\,,\ \ i = 1, \dots, n. \end{cases} \tag{25}$$

The Lagrangian dual form of the program (25) is as follows:

$$max_{\delta,\gamma,\hat{\delta},\hat{\gamma}}\ L(\delta, \gamma, \hat{\delta}, \hat{\gamma})$$
$$subject\ to\ \delta_i, \gamma_i, \hat{\delta}_i, \hat{\gamma}_i \geq 0, i = 1, \dots, n, \tag{26}$$

where

$$L(\delta, \gamma, \hat{\delta}, \hat{\gamma}) = inf \Big\{ \frac{1}{2} w^T w + C \sum_{i=1}^n (\xi_i + \hat{\xi}_i)$$
$$- \sum_{i=1}^n \delta_i \big(\varepsilon + \xi_i - y_i + w^T g(x_i) + b\big)$$
$$- \sum_{i=1}^n \hat{\delta}_i \big(\varepsilon + \hat{\xi}_i + y_i - w^T g(x_i) - b\big)$$
$$- \sum_{i=1}^n \gamma_i\, \xi_i - \sum_{i=1}^n \hat{\gamma}_i\, \hat{\xi}_i \Big\}. \tag{27}$$

For the optimal solution, the following conditions are satisfied:

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^n (\delta_i - \hat{\delta}_i) g(x_i), \tag{28}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow\ \sum_{i=1}^n (\delta_i - \hat{\delta}_i) = 0, \tag{29}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \rightarrow\ \delta_i = C - \gamma_i\,,\ i = 1, \dots, n, \tag{30}$$

$$\frac{\partial L}{\partial \hat{\xi}_i} = 0 \rightarrow\ \hat{\delta}_i = C - \hat{\gamma}_i\,,\ i = 1, \dots, n, \tag{31}$$

$$\delta_i \big(\varepsilon + \xi_i - y_i + w^T g(x_i) + b\big) = 0,\ \ i = 1, \dots, n, \tag{32}$$

$$\hat{\delta}_i \big(\varepsilon + \hat{\xi}_i + y_i - w^T g(x_i) - b\big) = 0,\ \ i = 1, \dots, n, \tag{33}$$

$$\gamma_i \xi_i = 0,\ \ i = 1, \dots, n, \tag{34}$$

$$\hat{\gamma}_i \hat{\xi}_i = 0,\ \ i = 1, \dots, n, \tag{35}$$

$$\delta_i, \gamma_i, \hat{\delta}_i, \hat{\gamma}_i \geq 0,\ \ i = 1, \dots, n. \tag{36}$$

Using the above conditions, $L(\delta, \gamma, \hat{\delta}, \hat{\gamma})$ is transformed into

$$\sum_{i=1}^n (\delta_i - \hat{\delta}_i) y_i - \varepsilon \sum_{i=1}^n (\hat{\delta}_i + \delta_i)$$
$$-\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \big(\delta_i - \hat{\delta}_i\big)\big(\delta_j - \hat{\delta}_j\big) K(x_i, x_j)\,. \tag{37}$$

Since $\delta_i, \gamma_i \geq 0$, from Eq. (30) we have $0 \leq \delta_i \leq C$. Also, since $\tilde{\delta}_i, \hat{\gamma}_i \geq 0$, from Eq. (31) we have $0 \leq \hat{\delta}_i \leq C$. Thus, the Lagrangian dual form of the program (25) becomes as follows:

$$max_{\delta,\hat{\delta}} \sum_{i=1}^n (\delta_i - \hat{\delta}_i) y_i - \varepsilon \sum_{i=1}^n (\hat{\delta}_i + \delta_i)$$
$$-\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \big(\delta_i - \hat{\delta}_i\big)\big(\delta_j - \hat{\delta}_j\big) K(x_i, x_j)$$
$$s.t. \begin{cases} \sum_{i=1}^n (\delta_i - \hat{\delta}_i) = 0\,; \\ 0 \leq \delta_i \leq C,\ \ i = 1, \dots, n; \\ 0 \leq \hat{\delta}_i \leq C,\ \ i = 1, \dots, n. \end{cases} \tag{38}$$

Therefore, the upper bound program (5) can be restated as follows:

$$\tilde{J}_\alpha^U = \max_y \begin{cases} max_{\delta,\hat{\delta}} \sum_{i=1}^n (\delta_i - \hat{\delta}_i) y_i - \varepsilon \sum_{i=1}^n (\hat{\delta}_i + \delta_i) \\ -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\delta_i - \hat{\delta}_i)(\delta_j - \hat{\delta}_j) K(x_i, x_j) \\ s.t. \begin{cases} \sum_{i=1}^n (\delta_i - \hat{\delta}_i) = 0\,; \\ 0 \le \delta_i \le C, \\ 0 \le \hat{\delta}_i \le C, \;\; i = 1, \dots, n. \end{cases} \end{cases}$$

$$s.t. (y_i)_\alpha^L \le y_i \le (y_i)_\alpha^U, \;\; i = 1, \dots, n,$$

or equivalently

$$\begin{aligned} & max_{y,\delta,\hat{\delta}} \sum_{i=1}^n (\delta_i - \hat{\delta}_i) y_i - \varepsilon \sum_{i=1}^n (\hat{\delta}_i + \delta_i) \\ & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\delta_i - \hat{\delta}_i)(\delta_j - \hat{\delta}_j) K(x_i, x_j) \\ & s.t. \begin{cases} \sum_{i=1}^n (\delta_i - \hat{\delta}_i) = 0\,; \\ 0 \le \delta_i \le C, \\ 0 \le \hat{\delta}_i \le C, \\ (y_i)_\alpha^L \le y_i \le (y_i)_\alpha^U, \;\; i = 1, \dots, n. \end{cases} \end{aligned} \tag{39}$$

The program (39) is a non-convex quadratic program with linear constraint and bounded variables. Therefore, its global optimal solution can be obtained using the reformulation-linearization/convexification technique (RLT) [22].

From Eq. (32), if $\delta_i > 0$, $y_i - w^T g(x_i) - b = \varepsilon + \xi_i$. From Eq. (30), if $\delta_i < C, \gamma_i > 0$ and if $\gamma_i > 0$, from Eq. (34), $\xi_i = 0$. Thus, if $0 < \delta_i < C$,

$$y_i - w^T g(x_i) - b = \varepsilon. \tag{40}$$

Also, from Eq. (33), if $\hat{\delta}_i > 0$, $-y_i + w^T g(x_i) + b = \varepsilon + \hat{\xi}_i$. From Eq. (31), if $\hat{\delta}_i < C, \tilde{\gamma}_i > 0$ and if $\tilde{\gamma}_i > 0$ from Eq. (35), $\hat{\xi}_i = 0$. Thus, if $0 < \hat{\delta}_i < C$,

$$-y_i + w^T g(x_i) + b = \varepsilon. \tag{41}$$

From Eq. (28), the approximated hyperplane in feature space or actual output is given by

$$\begin{aligned} f(x) &= w^T g(x) + b = \sum_{i=1}^n (\delta_i - \hat{\delta}_i) g(x_i)^T g(x) + b \\ &= \sum_{i=1}^n (\delta_i - \hat{\delta}_i) K(x_i, x) + b\,, \end{aligned}$$

where from Eq. (40) and Eq. (41) the bias satisfies

$$b = y_i - w^T g(x_i) - \varepsilon, \text{ for } 0 < \delta_i < C, \tag{42}$$

$$b = y_i - w^T g(x_i) + \varepsilon, \text{ for } 0 < \hat{\delta}_i < C. \tag{43}$$

In calculating bias, to avoid calculation errors, we average biases that satisfy (42) and (43).

D. Obtaining the Fuzzy Actual Output

In the previous sub-section, we obtained $\tilde{J}_\alpha^L$ and $\tilde{J}_\alpha^U$ for some $\alpha \in [0,1]$. Therefore, the membership function of the fuzzy function $\tilde{J}$ was determined.

The lower bound of fuzzy weight vector and fuzzy bias at α-cut, namely $\widetilde{w}_\alpha^L$ and $\tilde{b}_\alpha^L$, are indeed the optimal solution of $\tilde{J}_\alpha^L$. Also, the upper bound of fuzzy weight vector and fuzzy bias at α-cut, namely $\widetilde{w}_\alpha^U$ and $\tilde{b}_\alpha^U$, are indeed the optimal solution of $\tilde{J}_\alpha^U$. The lower and upper bound of fuzzy actual output or fuzzy hyperplane in high dimensional feature space $\tilde{f}(.)$ at α-cut can be obtained as follows:

$$\tilde{f}_\alpha^L(x) = \widetilde{w}_\alpha^{L^T} g(x) + \tilde{b}_\alpha^L,$$

$$\tilde{f}_\alpha^U(x) = \widetilde{w}_\alpha^{U^T} g(x) + \tilde{b}_\alpha^U.$$

For a fuzzy set A and for each $\alpha \le \beta \in [0,1]$, we have $A_\alpha^L \le A_\beta^L$ and $A_\alpha^U \ge A_\beta^U$. The function $\tilde{J}$ satisfies this condition, but, the bias term $\tilde{b}$ and actual output $\tilde{f}(.)$ don’t satisfy. Therefore, we change the bias term $\tilde{b}$ and actual output $\tilde{f}(.)$ as follows to satisfy this condition:

$$\tilde{b}_\alpha^L(x) = \min\{\tilde{b}_\alpha^L, \tilde{b}_\beta^L\},$$

$$\tilde{b}_\alpha^U(x) = \max\{\tilde{b}_\alpha^U, \tilde{b}_\beta^U\},$$

$$\tilde{f}_\alpha^L(x) = \min\{\tilde{f}_\alpha^L(x), \tilde{f}_\beta^L(x)\},$$

$$\tilde{f}_\alpha^U(x) = \max\{\tilde{f}_\alpha^U(x), \tilde{f}_\beta^U(x)\},$$

where $\alpha \le \beta \in [0,1]$.

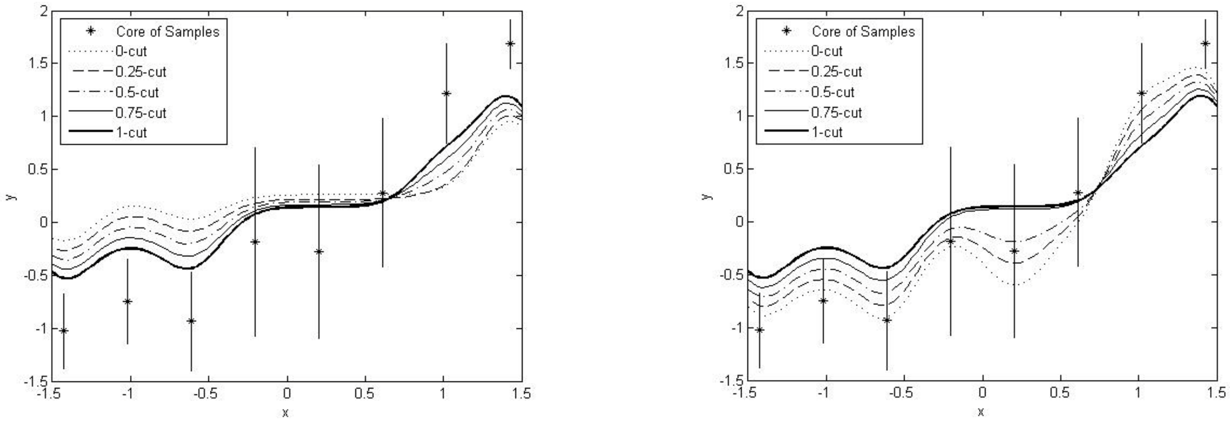
IV. Experimental Results

In this section, the proposed algorithm is utilized using some training samples. Here, for ease of evaluation, the target output $\tilde{y}_i$ is considered to be symmetric triangular fuzzy numbers. Moreover, we use the Gaussian kernel function, namely $k(x,z) = e^{\frac{\|x-z\|^2}{2\sigma^2}}$. We set $C = 1000$, $\sigma = 0.2$ and $\varepsilon = 0.5$.

Figure 5 plots the lower bound and the upper bound of the fuzzy hyperplane in feature space (fuzzy actual output) at some distinct α values. The lower bound and the upper bound of fuzzy hyperplane have been constructed based on the optimistic and the pessimistic value of target outputs, respectively. Therefore, as it can be seen, the lower bound of fuzzy hyperplane has less curvature than the upper bound of fuzzy hyperplane specially at 0-cut.

Table 1 lists the α-cuts of the fuzzy function $\tilde{J}$, the fuzzy bias $\tilde{b}$ and the fuzzy hyperplane $\tilde{f}(x) = \widetilde{w}^T g(x) + \tilde{b}$ for $x = -0.5, 0$ and 0.5, at some distinct values of α. The α-cut of $\tilde{J}, \tilde{b}$ and $\tilde{f}(x)$ represents the possibility that the objective function, bias and output will appear in the associated range, respectively. The α value indicates the level of possibility and the degree of uncertainty of the obtained information. The greater the α value, the greater the level of possibility and the lower the degree of uncertainty is. Different α-cuts of $\tilde{J}, \tilde{b}$ and $\tilde{f}(x)$ show the different intervals and the uncertainty levels of $\tilde{J}, \tilde{b}$ and $\tilde{f}(x)$, respectively. Specifically, $\tilde{J}, \tilde{b}$ and $\tilde{f}(x)$, at $\alpha = 0$ have the widest interval indicating that $\tilde{J}, \tilde{b}$ and $\tilde{f}(x)$ will definitely fall into the corresponding range. At the other extreme end, the corresponding possibility level $\alpha = 1$ is the most possible value of $\tilde{J}, \tilde{b}$ and $\tilde{f}(x)$.

Figure 6 plots the membership function of fuzzy bias, and Figure 7 plots the fuzzy hyperplane in feature space (fuzzy actual output) for $x = -0.5, 0$ and 0.5. As it can be seen, however we used training samples whose target outputs were symmetric triangular fuzzy numbers, the membership function of the fuzzy bias is not triangular. Therefore, as it was stated earlier, the pre-assumption on the membership function of fuzzy bias and the components of weight vector makes the model inaccurate.



(a)

(b)

Figure 1. (a) Lower bound of fuzzy hyperplane (b) Upper bound of fuzzy hyperplane (actual output) at α-cut.
*: Core of training samples.

TABLE I. THE LOWER BOUND AND THE UPPER BOUND OF THE FUZZY FUNCTION $\tilde{J}$, THE FUZZY BIAS $\tilde{B}$ AND THE FUZZY HYPERPLANE $\tilde{F}(X) = \tilde{W}^T G(X) + \tilde{B}$ FOR X $= -0.5, 0$ AND 0.5 AT SOME DISTINCT A-CUT.

			$\tilde{f}^L_\alpha(x)$					$\tilde{f}^U_\alpha(x)$		
α	$\tilde{J}^L_\alpha$	$\tilde{b}^L_\alpha$	$x = -0.5$	$x = 0$	$x = 0.5$	$\tilde{J}^U_\alpha$	$\tilde{b}^U_\alpha$	$x = -0.5$	$x = 0$	$x = 0.5$
0.0	0.3603	0.2607	0.0679	0.2586	0.2607	2.5722	-0.0740	-0.7553	-0.3932	-0.2130
0.25	0.4789	0.2164	-0.0354	0.2137	0.2181	2.0957	-0.0161	-0.6490	-0.2476	-0.1115
0.5	0.6336	0.1921	-0.1346	0.1886	0.1983	1.6858	0.0418	-0.5427	-0.1020	-0.0099
0.75	0.8283	0.1678	-0.2339	0.1636	0.1786	1.3379	0.1193	-0.4323	0.1134	0.1390
1.0	1.0631	0.1435	-0.3331	0.1385	0.1588	1.0631	0.1435	-0.3331	0.1385	0.1588

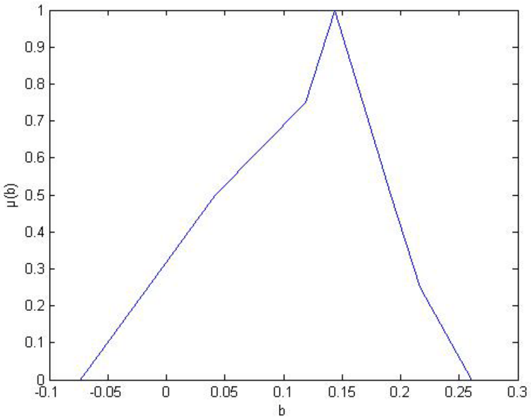
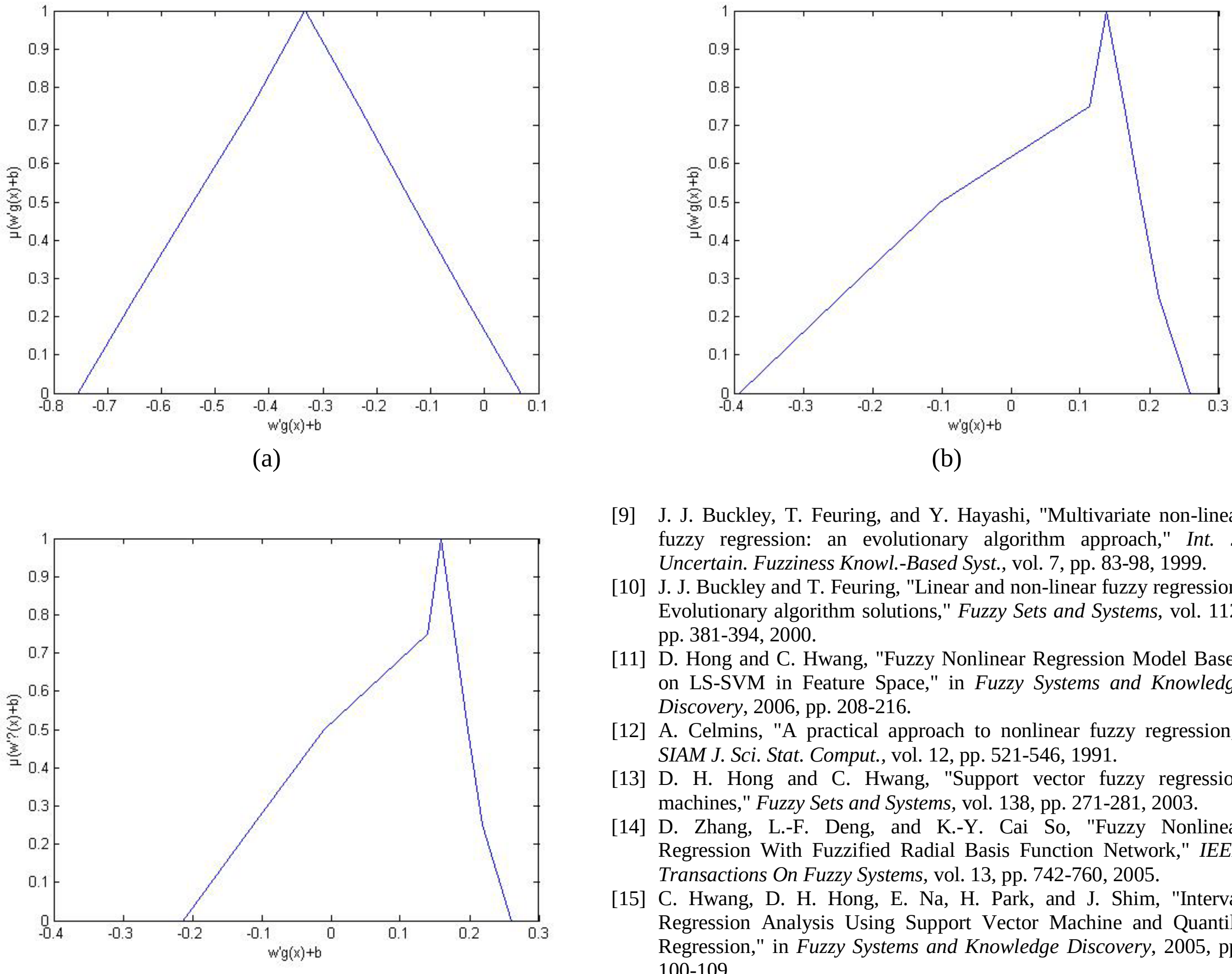


Figure 2. The membership function of the fuzzy bias $\tilde{b}$.



(a)

(b)

(c)

Figure 3. The membership function of the fuzzy hyperplane in feature space (fuzzy actual output) for (a) $x = -0.5$ (b) $x = 0$ (c) $x = 0.5$

REFERENCES

[1] B. Heshmaty and A. Kandel, "Fuzzy linear regression and its applications to forecasting in uncertain environment," *Fuzzy Sets and Systems*, vol. 15, pp. 159-191, 1985.

[2] M. Kaneyoshi, H. Tanaka, M. Kamei, and H. Farata, "New system identification technique using fuzzy regression analysis," in *Uncertainty Modeling and Analysis*, 1990, pp. 528–533.

[3] H. Tanaka, S. Uejima, and K. Asai, "Fuzzy linear regression model," in *Applied Systems and Cybernetics*, Acapulco, Mexico, 1980, pp. 12-15.

[4] H. Tanaka, S. Uejima, and K. Asai, "Linear regression analysis with fuzzy model," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 12, pp. 903–907, 1982.

[5] P. Diamond, "Fuzzy least squares," *Information Sciences,* vol. 46, pp. 141-157, 1988.

[6] P. Diamond, "Least squares fitting of several fuzzy variables," in *IFSA Congress,* Tokyo, Japan, 1987, pp. 329-331.

[7] J. Kacprzyk and M. Fedrizzi, *Fuzzy Regression Analysis*. Heidelberg, Germany: Physica-Verlag, 1992.

[8] P.-Y. Hao and J.-H. Chiang, "Fuzzy Regression Analysis by Support Vector Learning Approach," *IEEE Transactions On Fuzzy Systems,* vol. 16, pp. 428-441, 2008.

[9] J. J. Buckley, T. Feuring, and Y. Hayashi, "Multivariate non-linear fuzzy regression: an evolutionary algorithm approach," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 7, pp. 83-98, 1999.

[10] J. J. Buckley and T. Feuring, "Linear and non-linear fuzzy regression: Evolutionary algorithm solutions," *Fuzzy Sets and Systems,* vol. 112, pp. 381-394, 2000.

[11] D. Hong and C. Hwang, "Fuzzy Nonlinear Regression Model Based on LS-SVM in Feature Space," in *Fuzzy Systems and Knowledge Discovery*, 2006, pp. 208-216.

[12] A. Celmins, "A practical approach to nonlinear fuzzy regression," *SIAM J. Sci. Stat. Comput.*, vol. 12, pp. 521-546, 1991.

[13] D. H. Hong and C. Hwang, "Support vector fuzzy regression machines," *Fuzzy Sets and Systems,* vol. 138, pp. 271-281, 2003.

[14] D. Zhang, L.-F. Deng, and K.-Y. Cai So, "Fuzzy Nonlinear Regression With Fuzzified Radial Basis Function Network," *IEEE Transactions On Fuzzy Systems,* vol. 13, pp. 742-760, 2005.

[15] C. Hwang, D. H. Hong, E. Na, H. Park, and J. Shim, "Interval Regression Analysis Using Support Vector Machine and Quantile Regression," in *Fuzzy Systems and Knowledge Discovery*, 2005, pp. 100-109.

[16] J.-T. Jeng, C.-C. Chuang, and S.-F. Su, "Support vector interval regression networks for interval regression analysis," *Fuzzy Sets Syst.*, vol. 138, pp. 283-300, 2003.

[17] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*. vol. 9: MIT Press, 1996, pp. 155-161.

[18] V. Vapnik, S. E. Golowich, and A. J. Smola, "Support vector method for function approximation, regression estimation, and signal processing," *Advances in Neural Information Processing Systems,* vol. 9, pp. 281–287, 1996.

[19] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing,* vol. 14, pp. 199-222, 2004.

[20] S.-T. Liu, "A revisit to quadratic programming with fuzzy parameters," *Chaos, Solitons & Fractals,* vol. 41, pp. 1401-1407, 2009.

[21] L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems,* vol. 100, pp. 9-34, 1978.

[22] M. S. Bazara, H. D. Sherali, and C. M. Shetty, *Nonlinear programming*, 3 ed.: John Willey & Sons, Inc., 2006.