SUPPORT VECTOR REGRESSION WITH RANDOM OUTPUT VARIABLE AND PROBABILISTIC CONSTRAINTS

M. ABASZADE AND S. EFFATI

Abstract. Support Vector Regression (SVR) solves regression problems based on the concept of Support Vector Machine (SVM). In this paper, a new model of SVR with probabilistic constraints is proposed that any of output data and bias are considered the random variables with uniform probability functions. Using the new proposed method, the optimal hyperplane regression can be obtained by solving a quadratic optimization problem. The proposed method is illustrated by several simulated data and real data sets for both models (linear and nonlinear) with probabilistic constraints.

1. Introduction

Statistical Learning Theory has provided a very effective framework for classification and regression tasks involving features. SVMs are directly derived from this framework and they work by solving a constrained quadratic problem where the convex objective function for minimization is given by the combination of a loss function with a regularization term (the norm of the weights).

SVM is a concept in statistics and computer science for a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. SVMs originally introduced by Vapnik [41].

Least squares SVM (LS-SVM) as a variant of SVM, is a quadratic programming (QP) problem based on the equality constraints [35,45]. Therefore computational speed of LS-SVM is faster than SVM. It is the most important advantage of LS-SVM. In the LS-SVM, every data point is a support vector. Also SVM has been successfully applied to produce fuzzy rules [8,13,14,24].

Lately, Jayadeva et al. [16] proposed a twin support vector machine (TSVM) for binary classification problems. TSVM finds two nonparallel hyper planes by solving a pair of the smaller sized QP Problems. Hence not only computational complexity of TSVM is less than SVM but also learning speed of TSVM is faster than SVM. Least squares twin SVM (LS-TSVM) as a variant of TSVM was proposed that solves two smaller sized QPs instead of a large sized QP in LS-SVM [19]. It is remarkable that the objective function of LS-TSVM only researches the training error of samples of two classes and does not consider the generalization performance

Received: August 2015; Revised: June 2016; Accepted: September 2016

Key words and phrases: Probabilistic constraints, Support Vector Machine, Support Vector Regression, Quadratic programming, Probability function, Monte Carlo simulation.

of classifier. Then in this case, capability of classification decreases. Xu et al. displayed an improved LS-TSVM to increase the accuracy of classification [44].

In practical classification problems such as sensor database, location database, biometric information systems, the samples can not be observed exactly because of measurement errors. In this case, standard SVM may not be successful. For solving this problem, the effect of noise was eliminated using various methods [23] or SVM was investigated in a probabilistic framework [4,12,17,20,22,23,25,30,34,]. A probabilistic SVM [23] was proposed to capture the probabilistic information of the separating margin. Some researchers have presented that robust support vector machines (RSVMs) act better than standard SVMs [3,15,31,37].
Sadoghi et al. [32] proposed a new SVM classifier with probabilistic constraints which constraints boundaries have probability density functions and constraints occur with probability between 0 to 1.

Lanckriet et al. [21] considered the case of binary classification, where only the mean and covariance matrix of the classes are assumed to be known. The minimax probabilistic decision hyperplane is then determined by optimizing the worst-case probabilities over all possible class-conditional distributions. The computational complexity of their method is comparable to the QP that one has to solve for the SVM.

A version of SVM for regression was proposed by Druker et al. [9]. This method is called support vector regression (SVR). SVR is the most common application form of SVMs. An overview of the basic ideas underlying SVMs for regression and function estimation has been given in [11,42].

LS-SVM for the regression problem is called least squares SVR (LS-SVR) that is a QP problem based on the equality constraints [5]. Although LS-SVR can increase the learning speed, the robustness of it is not as good as that of SVR.

Similar to TSVM, twin support vector regression (TSVR) [29] generates two nonparallel functions around the training data. Whereas training samples have different effects on the up and down functions depending on their different situations, Xu et al. [43] proposed a weighted TSVR such that different penalties are given to the training samples in their different situations. the prediction error of weighted TSVR is lower than that of standard SVR and TSVR.
The standard SVR assumes that the training data are known exactly. However frequently in practical regression models, training data containing input and/or output data cannot be observed precisely because of sampling errors, modeling errors or measurement errors. In this case, recent studies have shown that robust support vector regressions (RSVRs) perform better than standard SVRs [6,7,36]. In many existing studies, chance-constrained problem be converted into a second-order cone programming (SOCP) [26] which can be solved efficiently by the interior point method (IPM) [28]. Generally, solving an SOCP is more difficult than solving QP.

Shivaswamy et al. [33] considered the Penalized Linear Regression and SVR when input data are random variables whose first two moments are known.

Main motivation of this paper rely on probabilistic constraints to construct optimal hyperplane regression by using the SVR and importance of each samples in determination of hyperplane parameters.

The rest of the paper is organized as follows: Section 2 introduces the brief study of SVR. In section 3 and 4, we propose a new linear and nonlinear SVR with probabilistic constraints when output data and bias are uniform random variables. Performances on Monte Carlo simulated data sets and real data sets are shown in section 5 and we have discussed the results in this section. The conclusion follows in the final section of this paper.

2. **The Brief Study of SVR**

Suppose that we are given training data $\{(x_1, y_1), \cdots, (x_n, y_n)\} \subset \chi \times \Re$, where χ denotes the space of the input patterns ($e.g.\ \chi = \Re^m$).
In $\epsilon-$SVR [41], we find a function $f(x)$ that has at most ϵ deviation from the actual y_i for all the training data. This relies on defining the loss function that ignores errors which are situated within the certain distance of the true value. This type of function is called ϵ-insensitive loss function as defined as:

$$L_\epsilon(d, y) = \begin{cases} |d - y| - \epsilon & \text{if} \quad |d - y| \geq \epsilon \\ 0 & \text{otherwise.} \end{cases}$$

Also $f(x)$ must be as flat as possible. Consider the case of linear $f(x)$,

$$f(x) = w^T x + b \quad \text{with} \quad w \in \chi,\ b \in \Re, \tag{1}$$

Finding a small w produces flatness in (1). Hence it is required to minimize the Euclidean norm w, i.e. $\|w\|^2 = w^T w$. Therefore this can be stated by the following convex optimization problem:

$$\begin{aligned} &\text{Minimize} && \frac{1}{2}\|w\|^2 \\ &\text{subject to} && \begin{cases} |y_i - w^T x_i - b| \leq \epsilon, \\ i = 1, \cdots, n. \end{cases} \end{aligned} \tag{2}$$

Now we introduce slack variables $\xi_i\,, \xi_i'$ to control the infeasible constraints of the optimization problem (2). Then we get the following optimization problem

$$\begin{aligned} &\text{Minimize} && \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i') \\ &\text{subject to} && \begin{cases} y_i - w^T x_i - b \leq \epsilon + \xi_i, \\ w^T x_i + b - y_i \leq \epsilon + \xi_i', \\ \xi_i, \xi_i' \geq 0, \quad i = 1, \cdots, n. \end{cases} \end{aligned} \tag{3}$$

The constant $C > 0$ has to be selected by the user and determines penalty parameter of the error term.
The quadratic optimization problem (3) is solved easily in its dual problem. For

this, it is required to construct a Lagrange function. It can be shown that the Lagrange function has a saddle point with respect to the primal and dual variables at the solution [27].

It follows from the saddle point condition that the partial derivatives of Lagrange function with respect to the primal variables $w\,, b\,, \xi_i\,, \xi_i'$ have to vanish for optimality.

3. The Proposed Probabilistic Constraints SVR

In many real-world applications, available information is often uncertain, imprecise and incomplete, thus usually is presented by random variables. Frequently in practical regression models, training data containing input and/or output data cannot be observed precisely because of sampling errors, modeling errors or measurement errors. Thus usually they are presented by random variables. In order to achieve robustness, the constraints in (3) must be replaced with probability constraints. Probabilistic constraints SVR finds the optimal hyperplane regression with the minimal error. In this section we applied the probability theory to the SVR.

First, we deal with randomized output in the regression task. Suppose that the given output data denoted by Y_i are uniform random variables on interval (y_{il}, y_{iu}) with the following probability function [18]

$$Y_i \sim U(y_{il}, y_{iu}) \implies f_{Y_i}(y_i) = \begin{cases} \frac{1}{y_{iu}-y_{il}} & \text{if} \quad y_i \in (y_{il}, y_{iu}) \\ 0 & \text{o.w.} \end{cases}$$

Then for handling those randomized training data, bias B in this model is also set to be uniform random variable on interval (b_l, b_u) with the following probability function

$$B \sim U(b_l, b_u) \implies f_B(b) = \begin{cases} \frac{1}{b_u-b_l} & \text{if} \quad b \in (b_l, b_u) \\ 0 & \text{o.w.} \end{cases}$$

Also we suppose that B and Y_i are independent together. Then

$$f_{Y_i,B}(y_i, b) = f_{Y_i}(y_i) f_B(b).$$

3.1. Model Structure in Linear Case.

In the proposed algorithm, optimal hyperplane regression can be obtained by solving the following optimization problem

$$\begin{aligned} &\text{Minimize} \qquad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i') \\ &\text{subject to} \quad \begin{cases} P_r(Y_i - w^T x_i - B \le \epsilon + \xi_i) \ge \delta, \\ P_r(w^T x_i + B - Y_i \le \epsilon + \xi_i') \ge \delta, \\ \xi_i, \xi_i' \ge 0, \qquad i = 1, \cdots, n. \end{cases} \end{aligned} \tag{4}$$

where $\delta \in [0, 1]$.

The optimization problem with probability inequality constraints (4) is difficult to solve, we now convert the optimization problem into a solvable QP using the

probability theory.
Therefore

$$\begin{aligned} P_r(Y_i - B \le w^T x_i + \epsilon + \xi_i) &= \int_{b_l}^{b_u} \int_{y_{il}}^{w^T x_i + \epsilon + \xi_i + b} f_{Y_i}(y_i) f_B(b) dy_i\, db \\ &= \int_{b_l}^{b_u} \frac{(w^T x_i + \epsilon + \xi_i - y_{il} + b)}{(y_{iu} - y_{il})(b_u - b_l)} db \\ &= \frac{1}{(y_{iu} - y_{il})} \left[w^T x_i + \epsilon + \xi_i - y_{il} + \frac{1}{2}(b_u + b_l) \right], \end{aligned}$$

and

$$\begin{aligned} P_r(B - Y_i \le \epsilon + \xi_i' - w^T x_i) &= \int_{b_l}^{b_u} \int_{b - \epsilon - \xi_i' + w^T x_i}^{y_{iu}} f_{Y_i}(y_i) f_B(b) dy_i\, db \\ &= \int_{b_l}^{b_u} \frac{y_{iu} + \epsilon + \xi_i' - w^T x_i - b}{(y_{iu} - y_{il})(b_u - b_l)} db \\ &= \frac{1}{(y_{iu} - y_{il})} \left[y_{iu} + \epsilon + \xi_i' - w^T x_i - \frac{1}{2}(b_u + b_l) \right]. \end{aligned}$$

Then optimization problem (4) can be transformed into the following form

$$\begin{aligned} &\text{Minimize} \quad \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \xi_i') \\ &\text{subject to} \quad \begin{cases} w^T x_i + \frac{1}{2}(b_u + b_l) - y_{il} + \epsilon + \xi_i \ge \delta(y_{iu} - y_{il}), \\ y_{iu} - w^T x_i - \frac{1}{2}(b_u + b_l) + \epsilon + \xi_i' \ge \delta(y_{iu} - y_{il}), \\ \xi_i, \xi_i' \ge 0, \qquad i = 1, \cdots, n. \end{cases} \end{aligned} \tag{5}$$

The optimization problem of probabilistic constraints linear SVR (5) can be transformed into its dual problem. It follows from the saddle point condition that the partial derivatives of Lagrange function with respect to the primal variables w , b_l , b_u , ξ_i , ξ_i' have to vanish for optimality.
Optimization procedure continues as follows:

$$\begin{aligned} L(w, b_l, b_u, \xi, \xi', \alpha, \alpha', \beta, \beta') &= \frac{1}{2} w^T w + C \sum_{i=1}^{n} (\xi_i + \xi_i') - \sum_{i=1}^{n} (\beta_i \xi_i + \beta_i' \xi_i') \\ &+ \sum_{i=1}^{n} \alpha_i \left[\delta(y_{iu} - y_{il}) - w^T x_i - \epsilon - \xi_i + y_{il} - \frac{1}{2}(b_u + b_l) \right] \\ &+ \sum_{i=1}^{n} \alpha_i' \left[\delta(y_{iu} - y_{il}) - y_{iu} - \epsilon - \xi_i' + w^T x_i + \frac{1}{2}(b_u + b_l) \right] \end{aligned}$$

where

$$\alpha = (\alpha_1, \cdots, \alpha_n),\ \beta = (\beta_1, \cdots, \beta_n),\ \xi = (\xi_1, \cdots, \xi_n),$$

$$\alpha' = (\alpha_1', \cdots, \alpha_n'),\ \beta' = (\beta_1', \cdots, \beta_n'),\ \xi' = (\xi_1', \cdots, \xi_n').$$

Thus we have

$$\begin{aligned}
\frac{\partial L}{\partial w} &= w - \sum_{i=1}^{n}(\alpha_i - \alpha_i')x_i = 0 \implies \widehat{w} = \sum_{i=1}^{n}(\alpha_i - \alpha_i')x_i,\\
\frac{\partial L}{\partial b_l} &= \frac{1}{2}\sum_{i=1}^{n}(\alpha_i - \alpha_i') = 0,\\
\frac{\partial L}{\partial b_u} &= \frac{1}{2}\sum_{i=1}^{n}(\alpha_i - \alpha_i') = 0,\\
\frac{\partial L}{\partial \xi_i} &= C - \alpha_i - \beta_i = 0,\\
\frac{\partial L}{\partial \xi_i'} &= C - \alpha_i' - \beta_i' = 0, \qquad i = 1, \cdots, n,
\end{aligned}$$

where $\widehat{w}$ is optimal value of w. For solving this problem it is converted to dual form and finding the Lagrange multipliers α, α' that maximize the following objective function.

$$\begin{aligned}
\text{Maximize} \quad & - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \alpha_i')(\alpha_j - \alpha_j')x_i^T x_j\\
& - \epsilon\sum_{i=1}^{n}(\alpha_i + \alpha_i') + \sum_{i=1}^{n}\alpha_i\left[\delta\, y_{iu} + (1-\delta)y_{il}\right]\\
& - \sum_{i=1}^{n}\alpha_i'\left[(1-\delta)y_{iu} + \delta\, y_{il}\right]
\end{aligned}$$

$$\text{subject to} \quad \begin{cases} \sum_{i=1}^{n}(\alpha_i - \alpha_i') = 0,\\ \alpha_i\,, \alpha_i' \in [0, C], \quad i = 1, \cdots, n. \end{cases} \tag{6}$$

We get the following properties from the KKT conditions of optimization theory

$$\begin{aligned}
\alpha_i\left[\delta(y_{iu} - y_{il}) - \widehat{w}^T x_i - \epsilon - \xi_i + y_{il} - \frac{1}{2}(b_u + b_l)\right] = 0,\\
\alpha_i'\left[\delta(y_{iu} - y_{il}) - y_{iu} - \epsilon - \xi_i' + \widehat{w}^T x_i + \frac{1}{2}(b_u + b_l)\right] = 0,\\
\beta_i\xi_i = 0 \quad \Longrightarrow \quad (C - \alpha_i)\xi_i = 0,\\
\beta_i'\xi_i' = 0 \quad \Longrightarrow \quad (C - \alpha_i')\xi_i' = 0, \qquad i = 1, \cdots, n.
\end{aligned}$$

The evaluation of which yields so that $\xi_i = 0$ if $\alpha_i \in (0, C)$ and $\xi_i' = 0$ if $\alpha_i' \in (0, C)$, then

$$\begin{cases} \delta(y_{iu} - y_{il}) - \widehat{w}^T x_i - \epsilon + y_{il} - \frac{1}{2}(b_u + b_l) = 0 \quad \text{for} \quad \alpha_i \in (0, C), \quad i = 1, \cdots, n\\ \delta(y_{iu} - y_{il}) - y_{iu} - \epsilon + \widehat{w}^T x_i + \frac{1}{2}(b_u + b_l) = 0 \quad \text{for} \quad \alpha_i' \in (0, C), \quad i = 1, \cdots, n. \end{cases}$$

We know that mathematical expectation of uniform random variable B, is equal to

$\frac{1}{2}(b_u+b_l)$ that be shown by μ_B. We represent optimal value of μ_B by $\widehat{\mu}_B$. Therefore $\widehat{\mu}_B$ is equal to average between $\widehat{\mu}_B$'s that are computed by

$$\begin{cases} \widehat{\mu}_B &= \delta(y_{iu}-y_{il}) - \widehat{w}^T x_i - \epsilon + y_{i1} \quad \text{for} \quad \alpha_i \in (0,C), \quad i=1,\cdots,n \\ \widehat{\mu}_B &= y_{iu} - \delta(y_{iu}-y_{il}) - \widehat{w}^T x_i + \epsilon \quad \text{for} \quad \alpha_i' \in (0,C), \quad i=1,\cdots,n. \end{cases}$$

Thus, we can find optimal hyperplane regression as:

$$\widehat{E}_{f(x)} = \sum_{i=1}^{n} (\alpha_i - \alpha_i') x_i^T x + \widehat{\mu}_B$$

where $\widehat{E}_{f(x)}$ is estimation of mathematical expectation of uniform random variable $f(x)$,

$$f(x) = w^T x + B.$$

4. **The Proposed Probabilistic Constraint Nonlinear SVR**

Nonlinear support vector regression with probabilistic constraints is accomplished by fitting a linear regression in a higher dimensional feature space. A nonlinear transformation ϕ is used to transform data points from the input space of dimension m into a feature space having a higher dimension m_1. The nonlinear mapping is denoted by $\phi:\ R^m \to R^{m_1}$.
In the probabilistic constraints nonlinear SVR, optimal hyperplane regression can be obtained by solving the following optimization problem

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i') \\ \text{subject to} \quad & \begin{cases} P_r(Y_i - w^T\phi(x_i) - B \le \epsilon + \xi_i) \ge \delta, \\ P_r(w^T\phi(x_i) + B - Y_i \le \epsilon + \xi_i') \ge \delta, \\ \xi_i, \xi_i' \ge 0, \qquad i=1,\cdots,n. \end{cases} \end{aligned}$$

where $\phi(x_i) = [\phi^1(x_i),\cdots,\phi^{m_1}(x_i)]^T$.

Similar to sub-section 3.1, optimal hyperplane regression can be obtained by solving the following optimization problem

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{n}(\xi_i + \xi_i') \\ \text{subject to} \quad & \begin{cases} w^T\phi(x_i) + \frac{1}{2}(b_u+b_l) - y_{il} \ge \delta(y_{iu}-y_{il}) - \epsilon - \xi_i, \\ y_{iu} - w^T\phi(x_i) - \frac{1}{2}(b_u+b_l) \ge \delta(y_{iu}-y_{il}) - \epsilon - \xi_i', \\ \xi_i, \xi_i' \ge 0, \qquad i=1,\cdots,n. \end{cases} \end{aligned} \tag{7}$$

The optimization problem of probabilistic constraints nonlinear SVR (7) can be transformed into its dual problem. It follows from the saddle point condition that the partial derivatives of Lagrange function with respect to the primal variables

w , b_l , b_u , ξ_i , ξ_i' have to vanish for optimality.
Optimization procedure continues as follows:

$$L(w,b_l,b_u,\xi,\xi',\alpha,\alpha',\beta,\beta') = \frac{1}{2}w^T w + C\sum_{i=1}^{n}(\xi_i+\xi_i') - \sum_{i=1}^{n}(\beta_i\xi_i+\beta_i'\xi_i')$$

$$+\sum_{i=1}^{n}\alpha_i\left[\delta(y_{iu}-y_{il}) - w^T\phi(x_i) - \epsilon - \xi_i + y_{il} - \frac{1}{2}(b_u+b_l)\right]$$

$$+\sum_{i=1}^{n}\alpha_i'\left[\delta(y_{iu}-y_{il}) - y_{iu} - \epsilon - \xi_i' + w^T\phi(x_i) + \frac{1}{2}(b_u+b_l)\right]$$

where

$$\alpha = (\alpha_1,\cdots,\alpha_n),\ \beta = (\beta_1,\cdots,\beta_n),\ \xi = (\xi_1,\cdots,\xi_n),$$

$$\alpha' = (\alpha_1',\cdots,\alpha_n'),\ \beta' = (\beta_1',\cdots,\beta_n'),\ \xi' = (\xi_1',\cdots,\xi_n').$$

Thus we have

$$\begin{aligned}
\frac{\partial L}{\partial w} &= w - \sum_{i=1}^{n}(\alpha_i-\alpha_i')\phi(x_i) = 0 \implies \widehat{w} = \sum_{i=1}^{n}(\alpha_i-\alpha_i')\phi(x_i),\\
\frac{\partial L}{\partial b_l} &= \frac{1}{2}\sum_{i=1}^{n}(\alpha_i-\alpha_i') = 0,\\
\frac{\partial L}{\partial b_u} &= \frac{1}{2}\sum_{i=1}^{n}(\alpha_i-\alpha_i') = 0,\\
\frac{\partial L}{\partial \xi_i} &= C-\alpha_i-\beta_i = 0,\\
\frac{\partial L}{\partial \xi_i'} &= C-\alpha_i'-\beta_i' = 0, \qquad i = 1,\cdots,n.
\end{aligned}$$

For solving this problem, it is converted to dual form and finding the Lagrange multipliers α and α' that maximize the following objective function.

$$\begin{aligned}
\text{Maximize} \quad & -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i-\alpha_i')(\alpha_j-\alpha_j')K(x_i,x_j)\\
& -\epsilon\sum_{i=1}^{n}(\alpha_i+\alpha_i') + \sum_{i=1}^{n}\alpha_i\left[\delta\, y_{iu} + (1-\delta)y_{il}\right]\\
& -\sum_{i=1}^{n}\alpha_i'\left[(1-\delta)y_{iu} + \delta\, y_{il}\right]
\end{aligned}$$

$$\text{subject to} \quad \begin{cases} \sum_{i=1}^{n}(\alpha_i-\alpha_i') = 0,\\ \alpha_i\,,\alpha_i' \in [0,C], \quad i = 1,\cdots,n. \end{cases} \tag{8}$$

where matrix K is termed as a kernel matrix and its elements are given by

$$K(x_i, x_j) = \phi^T(x_i)\phi(x_j).$$

There are different kernel functions such as:

$$K(x, y) = (x.y)^p, \quad p = 2, 3, \cdots \qquad \text{(polynomial of degree } p\text{)}$$

$$K(x, y) = exp(-\frac{\|x - y\|^2}{2\sigma^2}) \qquad \text{(Gaussian)}.$$

We get the following properties from the KKT conditions of optimization theory

$$\alpha_i \left[\delta(y_{iu} - y_{il}) - \widehat{w}^T\phi(x_i) - \epsilon - \xi_i + y_{il} - \frac{1}{2}(b_u + b_l)\right] = 0,$$
$$\alpha_i' \left[\delta(y_{iu} - y_{il}) - y_{iu} - \epsilon - \xi_i' + \widehat{w}^T\phi(x_i) + \frac{1}{2}(b_u + b_l)\right] = 0,$$
$$\beta_i\xi_i = 0 \quad \Longrightarrow \quad (C - \alpha_i)\xi_i = 0,$$
$$\beta_i'\xi_i' = 0 \quad \Longrightarrow \quad (C - \alpha_i')\xi_i' = 0, \qquad i = 1, \cdots, n.$$

The evaluation of which yields so that $\xi_i = 0$ if $\alpha_i \in (0, C)$ and $\xi_i' = 0$ if $\alpha_i' \in (0, C)$ then $\widehat{\mu}_B$ is equal to average between $\widehat{\mu}_B$'s that are computed by

$$\begin{cases} \widehat{\mu}_B = \delta(y_{iu} - y_{il}) - \widehat{w}^T\phi(x_i) - \epsilon + y_{il} & \text{for} \quad \alpha_i \in (0, C), \quad i = 1, \cdots, n \\ \widehat{\mu}_B = y_{iu} - \delta(y_{iu} - y_{il}) - \widehat{w}^T\phi(x_i) + \epsilon & \text{for} \quad \alpha_i' \in (0, C), \quad i = 1, \cdots, n. \end{cases}$$

Thus, we can find optimal hyperplane regression as:

$$\widehat{E}_{f(x)} = \sum_{i=1}^{n}(\alpha_i - \alpha_i')K(x_i, x) + \widehat{\mu}_B$$

where $\widehat{E}_{f(x)}$ is the estimation of mathematical expectation of uniform random variable $f(x)$,

$$f(x) = w^T\phi(x) + B.$$

5. **Numerical Experiments**

5.1. **Monte Carlo Simulation.**

We start by testing the proposed method for linear case by using the Monte Carlo simulation as follows:
In the Monte Carlo simulations, for each sample size n, we generate randomly x_i for $i = 1, \cdots, n$ from uniform distribution on $(0, 10)$. Note that n is the number of samples. Also we compute the corresponding y_{il} and y_{iu} were chosen as:

$$y_{il} = w_0^T x_i + \mu_{0B} - \eta_i,$$
$$y_{iu} = w_0^T x_i + \mu_{0B} + \eta_i$$

where $\mu_{0B} = 5$, η_i is a random point on interval $(0, 1)$ and w_0 varies over the different values. Finally normal distributed noise with zero mean and a random chosen covariance on interval $(0, 1)$ is added to x_i, y_{il} and y_{iu}. Afterwards we generate $y_i \in U(y_{il}, y_{iu})$.

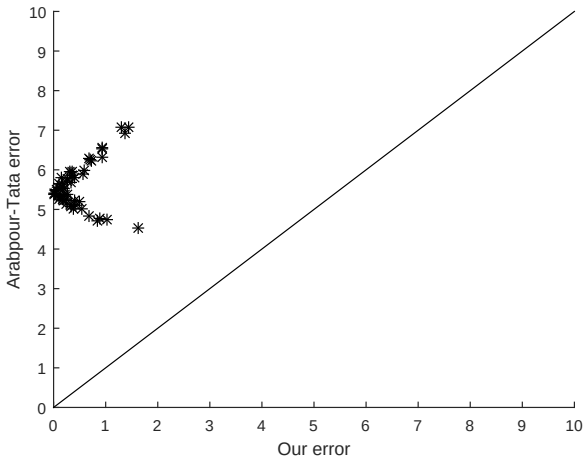

FIGURE 1. Error of Simulation Data

We consider $C = 100$, $\epsilon = 0.1$ and $\delta = 0.99$. By using Matlab software and solving the optimization problem (6) for simulated data, we get the optimal multipliers α_i, α_i' and optimal hyperplane regression.

We know that fuzzy linear regression models are applied to obtain an appropriate relation between input and output variables in a fuzzy system.

Arabpour and Tata estimated parameters of fuzzy linear regression model [1]. Sum of errors obtained by their method is smaller than the previous methods. Whereas membership function in fuzzy logic is similar to distribution function of uniform random variable we compare our method in linear case with Arabpour-Tata method [1]. For this, we consider fuzzy output $\widetilde{y}_i = (y_{il}, \frac{y_{il}+y_{iu}}{2}, y_{iu})$ and crisp input x_i. Afterwards by using Arabpour-Tata method [1], we estimate parameters of linear regression model. They have used the method of least squares to estimate the parameters when input or output are fuzzy variables.

Figure 1 shows a plot of our errors and the errors of the Arabpour-Tata method [1] when $n = 50$ and $w_0 = 1.4$. All of points lie above the bisector and we conclude that our errors in the linear case were smaller than the errors estimating by the Arabpour-Tata method [1].

10000 replications are made. We find optimal hyperplane regression using standard linear SVR, fuzzy linear regression model [1] and our method in any of replications. Then we compute simulated sum of squared error ($SSSE$) and simulated root mean squared error ($SRMSE$) of prediction for any of methods. Conclusions are shown in Table 1 and Table 2. It is clear that $SSSE$ and $SRMSE$ given by our method is less than the other methods. Also we compute simulated bias of $\widehat{w}$ ($SB(\widehat{w})$), simulated bias of $\widehat{\mu}_B$ ($SB(\widehat{\mu}_B)$), simulated mean squared error of $\widehat{w}$ ($SMSE(\widehat{w})$) and simulated mean squared error of $\widehat{\mu}_B$ ($SMSE(\widehat{\mu}_B)$) for standard linear SVR and our method . Conclusions are shown in Table 3.

$SSSE$, $SRMSE$, $SB(\widehat{w})$, $SB(\widehat{\mu}_B)$, $SMSE(\widehat{w})$ and $SMSE(\widehat{\mu}_B)$ for our method are computed using the following formulas

$$SSE(r) = \sum_{i=1}^{n} \left(\frac{y_{il} + y_{iu}}{2} - \widehat{E}_{rf(x)} \right)^2, \quad r = 1, \cdots, 10000,$$

$$RMSE(r) = \left(\frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_{il} + y_{iu}}{2} - \widehat{E}_{rf(x)} \right)^2 \right)^{\frac{1}{2}}, \quad r = 1, \cdots, 10000,$$

$$SSSE = \frac{1}{10000} \sum_{r=1}^{10000} SSE(r),$$

$$SRMSE = \frac{1}{10000} \sum_{r=1}^{10000} RMSE(r),$$

$$SB(\widehat{w}) = \frac{1}{10000} \sum_{r=1}^{10000} \widehat{w}_{(r)} - w_0,$$

$$SB(\widehat{\mu}_B) = \frac{1}{10000} \sum_{r=1}^{10000} \widehat{\mu}_{B(r)} - \mu_{0B},$$

$$SMSE(\widehat{w}) = \frac{1}{10000} \sum_{r=1}^{10000} (\widehat{w}_{(r)} - \frac{1}{10000} \sum_{r=1}^{10000} \widehat{w}_{(r)})^2,$$

$$SMSE(\widehat{\mu}_B) = \frac{1}{10000} \sum_{r=1}^{10000} (\widehat{\mu}_{B(r)} - \frac{1}{10000} \sum_{r=1}^{10000} \widehat{\mu}_{B(r)})^2,$$

where $\widehat{w}_{(r)}$, $\widehat{\mu}_{B(r)}$, $SSE(r)$, $RMSE(r)$ and $\widehat{E}_{rf(x)}$ are respectively $\widehat{w}$, $\widehat{\mu}_B$, SSE, $RMSE$ and $\widehat{E}_{f(x)}$ in r'th replication.

With regard to the Table 3, simulated bias and mean squared error of $\widehat{w}$ and $\widehat{\mu}_B$ for our method are less than standard linear SVR. Also using our method, values of them are very small and decrease when n increases.

Simulated data for nonlinear case are generated by the following way:
We generate randomly $x_i = [x_i^1, \cdots, x_i^5]^T$ for $i = 1, \cdots, n$ from uniform distribution on $(0, 10)$. Then we compute the corresponding y_{il} and y_{iu} were chosen as:

$$y_{il} = exp(-\frac{\|x_i\|}{2}) + \mu_{0B} - \eta_i,$$

$$y_{iu} = exp(-\frac{\|x_i\|}{2}) + \mu_{0B} + \eta_i$$

where $\mu_{0B} = -7$ and η_i is a random point on interval $(0, 1)$. Finally normal distributed noise with zero mean and a random chosen covariance on interval $(0, 1)$ is added to x_i, y_{il} and y_{iu}. Afterwards we generate $y_i \in U(y_{il}, y_{iu})$. We consider the kernel function such as Gaussian, $C = 100$, $\epsilon = 0.1$ and $\delta = 0.99$. By using Matlab software and solving the optimization problem (8) for simulated data, we get the optimal multipliers α_i, α_i' and optimal hyperplane regression.

n	w_0	Performance measures	Standard linear SVR	Our method
20	0.6	$SSSE$	9.5550	7.6204
		$SRMSE$	0.6557	0.5618
	1.0	$SSSE$	14.4538	12.4073
		$SRMSE$	0.8117	0.7383
	1.4	$SSSE$	19.0082	17.9116
		$SRMSE$	0.9113	0.8601
	$[0.6, 1.4]^T$	$SSSE$	17.7881	16.5936
		$SRMSE$	0.8825	0.8364
	$[1.4, 1.0]^T$	$SSSE$	32.1784	28.7869
		$SRMSE$	1.1863	1.1105
	$[0.6, 1, 1.4]^T$	$SSSE$	16.4497	14.5334
		$SRMSE$	0.8623	0.8005
	$[5, 4, 3, 2, 1]^T$	$SSSE$	405.2819	386.2526
		$SRMSE$	4.0280	3.9205
50	0.6	$SSSE$	30.0708	28.8353
		$SRMSE$	0.7344	0.6939
	1.0	$SSSE$	42.9503	36.8587
		$SRMSE$	0.8890	0.8163
	1.4	$SSSE$	54.7997	48.6567
		$SRMSE$	0.9907	0.9144
	$[0.6, 1.4]^T$	$SSSE$	64.2218	48.7018
		$SRMSE$	1.0597	0.9283
	$[1.4, 1.0]^T$	$SSSE$	74.7913	67.3692
		$SRMSE$	1.1661	1.1083
	$[0.6, 1, 1.4]^T$	$SSSE$	86.4914	78.4294
		$SRMSE$	1.2800	1.2106
	$[5, 4, 3, 2, 1]^T$	$SSSE$	738.7500	731.0520
		$SRMSE$	3.3245	3.2974

TABLE 1. $SSSE$ and $SRMSE$ for Standard Linear SVR and Our Method

n	w_0	Performance measures	Arabpour-Tata method [1]	Our method
20	0.2	$SSSE$	25.8642	4.9984
		$SRMSE$	1.2932	0.2499
	0.4	$SSSE$	40.8242	6.6710
		$SRMSE$	2.0412	0.3336
	0.6	$SSSE$	79.7847	9.5874
		$SRMSE$	3.9892	0.4794
	0.8	$SSSE$	113.8117	11.0959
		$SRMSE$	5.6906	0.5548
	1.0	$SSSE$	172.9224	12.7020
		$SRMSE$	8.6471	0.6351
	1.2	$SSSE$	258.5798	13.0071
		$SRMSE$	12.9290	0.6504
	1.4	$SSSE$	404.1369	15.5450
		$SRMSE$	20.2068	0.7772
50	0.2	$SSSE$	85.3292	29.4378
		$SRMSE$	1.7066	0.5888
	0.4	$SSSE$	131.7421	32.9411
		$SRMSE$	2.6348	0.6588
	0.6	$SSSE$	295.2296	38.5403
		$SRMSE$	5.9046	0.7708
	0.8	$SSSE$	343.9855	26.9529
		$SRMSE$	6.8797	0.5391
	1.0	$SSSE$	476.8091	43.9053
		$SRMSE$	9.5362	0.8781
	1.2	$SSSE$	890.7823	41.2825
		$SRMSE$	17.8156	0.8257
	1.4	$SSSE$	1.2490e+03	56.5445
		$SRMSE$	24.9792	1.1309

TABLE 2. $SSSE$ and $SRMSE$ for Arabpour-Tata Method [1] and Our Method

In order to compare our method with the standard nonlinear SVR, we apply Monte Carlo simulation with 10000 replications and find optimal hyperplane regression using standard nonlinear SVR and our method in any of replications.

n	w_0	Performance measures	Standard linear SVR	Our method
20	1.4	$SB(\widehat{w})$	-0.0553	-0.0421
		$SMSE(\widehat{w})$	0.0143	0.0124
50	1.4	$SB(\widehat{w})$	-0.0536	-0.0311
		$SMSE(\widehat{w})$	0.0035	0.0002
20	1.4	$SB(\widehat{\mu}_B)$	0.2584	0.2443
		$SMSE(\widehat{\mu}_B)$	0.4436	0.3833
50	1.4	$SB(\widehat{\mu}_B)$	0.2570	0.1033
		$SMSE(\widehat{\mu}_B)$	0.0613	0.0044

TABLE 3. SB and $SMSE$ of Parameters for Standard Linear SVR and Our Method

n	Performance measures	Standard nonlinear SVR	Our method
20	$SSSE$	2.2627e+03	1080.1792
	$SRMSE$	9.6029	6.9993
50	$SSSE$	3.9912e+03	2.2548e+03
	$SRMSE$	8.7109	7.0064

TABLE 4. $SSSE$ and $SRMSE$ for Standard Nonlinear SVR and Our Method

	Number of samples	Number of features
Slump Test	103	7
Pyrim	74	27
Diabetes	43	2

TABLE 5. The Characteristics of Data sets

$SSSE$ and $SRMSE$ for both methods are shown in Table 4. It is clear that $SSSE$ and $SRMSE$ given by our method is less than the standard nonlinear SVR.

5.2. **Performances on Real Datasets.**

In this subsection, we test our method on the UCI database [2], LIBSVM Data [38] and repositories of data sets [39]. Three data sets are selected to demonstrate the efficiency of our method. They are Concrete (Slump Test when output variable is 28-day Compressive Strength (Mpa)), Qualitative Structure Activity Relationships (Pyrim) and Diabetes. The characteristics of these datasets are shown in Table 5.

We consider any of output data as mean of a uniform random variable on interval (y_{il}, y_{iu}) that

$$y_{il} = i\text{'th output datum} - \eta_i,$$

$$y_{iu} = i\text{'th output datum} + \eta_i, \qquad \eta_i > 0$$

where η_i is randomly chosen.

In order to compare our method with the standard SVR, we consider crisp input data x_i and output data y_i that are perturbed by a random chosen value on interval (y_{il}, y_{iu}).

By using Matlab software and solving the optimization problem (6) or (8), we get the optimal multipliers α_i, α_i' and optimal hyperplane regression.

Linear kernel and Gaussian kernel are used in our method and standard SVR. We consider $\epsilon = 0.05$ and $\delta = 0.9$. Firstly, the model parameters including the

	Linear		Gaussian kernel	
	Our method	Standard linear SVR	Our method	Standard nonlinear SVR
Slump Test	2.2794±0.8142	5.1083±6.5442	2.3007±0.4026	2.7621 ±0.6515
Pyrim	0.0639±0.0141	0.1270 ±0.0361	0.0564±0.0242	0.1276±0.0458
Diabetes	0.6426±0.1253	0.8142±0.1597	0.5856±0.1936	0.7511±0.2987

TABLE 6. The Mean and Standard Deviation of Testing $RMSE$ of Three Data sets

trade-off parameters C and the kernel parameters σ are trained and tested from all the data by 5-fold cross validation. The combinations of $C = [2^{-5}, 2^{-4}, \cdots, 2^{10}]$ and $\sigma = [0.1, 1, 5, 10, 15, 20, 25, 40, 50, 75, 90]$ will form different models. The model with the smallest testing root mean squared error ($RMSE$) is selected for further training. Afterwards, we randomly pick 90% of the patterns for training and the rest for testing. The bootstrap re-sampling technique [10] is used to select samples. To obtain credible performance of models, the training and testing procedure will be repeated 10 times for each of the different training and testing sets obtained from the bootstrap re-sampling technique. The mean testing $RMSE$ and corresponding standard deviation over 10 random partition of the training set and testing set are given in Table 6. It is obvious testing error by our method is less than the standard SVR. The results of three databases demonstrate that our method produces better results than the standard SVR.

Yang et al. proposed a novel robust LS-SVR (RLS-SVR) [45]. They compared their proposed method with LS-SVR, weighted LS-SVR (WLS-SVR) and iteratively reweighted LS-SVR (IRLS-SVR). The following statistical test has been used in their paper,

$$R^2 = 1 - \left(\frac{median(|y_i - f(x_i)|)}{median(|y_i - median(y_i)|)} \right)^2 .$$

Generally, $0 \leq R^2 \leq 1$ indicates a reasonable model. We know that $R^2 = 1$ corresponds to a perfect fit and $R^2 < 0$ corresponds to a bad fit. Now we test our method on Nelson data set with 128 samples and 2 features [40]. Let

$$y_{il} = y_i - \eta_i, \qquad y_{iu} = y_i + \eta_i$$

where η_i is randomly chosen. The model parameters including $C = [2^{-5}, 2^{-4}, \cdots, 2^{10}]$ and $\sigma = [0.1, 1, 5, 10, 15, 20, 25, 40, 50, 75, 90]$ are trained and tested from all the data by 10-fold cross validation. We consider Gaussian kernel, $\delta = 0.9$ and $\epsilon = 0.001$. The optimal R^2 obtained by LS-SVR, WLS-SVR, IRLS-SVR, RLS-SVR and our method is represented in Table 7. With regard to the Table 7, we understand that our method produces better result than other methods.

Xu et al. [43] proposed a weighted TSVR. They tested their proposed algorithm on the data sets from the UCI machine learning repository [2] and compared their method with standard SVR and TSVR. We have selected three data sets that are Auto-price, Machine cpu and Body fat. The characteristics of these data sets are shown in Table 8.

Method	LS-SVR	WLS-SVR	IRLS-SVR	RLS-SVR	Our method
C	2^5	2^6	2^5	2^7	2^4
σ	1	1	1	1	1
R^2	0.6517	0.6713	0.6719	0.7128	0.8961

TABLE 7. The Results of LS-SVR, WLS-SVR, IRLS-SVR, RLS-SVR and Our Method

	Number of samples	Number of features
Auto-price	159	14
Machine cpu	209	6
Body fat	252	14

TABLE 8. The Characteristics of Data sets

Method	Standard nonlinear SVR	TSVR	Weighted TSVR	Our method
Auto-price	6713.3± 2534.7	5043.0 ± 597.7	5659.9 ± 1775.5	4208.4±1483.6
Machine cpu	159.82 ±68.629	144.844 ±60.122	143.868 ± 60.200	115.434±55.3912
Body fat	0.061±0.032	0.016±0.011	0.016±0.011	0.0108±0.002

TABLE 9. The Mean and Standard Deviation of Testing $RMSE$ of Three Data sets

We consider any of output data as mean of a uniform random variable on interval (y_{il}, y_{iu}) that

$$y_{il} = i\text{'th output datum} - \eta_i,$$

$$y_{iu} = i\text{'th output datum} + \eta_i, \qquad \eta_i > 0$$

where η_i is randomly chosen. We consider Gaussian kernel and $\delta = 0.9$. The model parameters including $C = \{2^i | i = -3, \cdots, 8\}$, $\sigma = \{2^i | i = -4, \cdots, 8\}$ and $\epsilon = \{\frac{i}{10} | i = 1, \cdots, 9\}$ are trained and tested from all the data by 4-fold cross validation. The optimal parameters corresponding to the lowest $RMSE$ were used to further training. Afterwards, we randomly pick 80% of the patterns for training and the rest for testing. This process is repeated five times. The mean testing $RMSE$ and corresponding standard deviation over 5 random partition of the training set and testing set are given in Table 9. With regard to the Table 9, we understand that our method produces better result than weighted TSVR [43].

Finally we compare our method with the proposed fuzzy SVRs in [13]. For this, we use one data set containing crisp inputs x_i and fuzzy outputs $\widetilde{y}_i$ [13] that are shown in Table 10.

The given output data, denoted by $\widetilde{y}_i = (y_i, e_i)$, are symmetric triangular fuzzy numbers, where y_i is a center and e_i is a width. Hao and Chiang have applied polynomial kernel function of degree 5 and degree 3 [13]. In order to use our method,

i	1	2	3	4
x_i	0.1	0.2	0.3	0.4
$\widetilde{y}_i = (y_i, e_i)$	(2.25,0.75)	(2.875,0.875)	(2.5,1.0)	(4.25,1.75)
i	5	6	7	8
x_i	0.5	0.6	0.7	0.8
$\widetilde{y}_i = (y_i, e_i)$	(4.0,1.5)	(5.25,1.25)	(7.5,2.0)	(8.5,1.5)

TABLE 10. Crisp Inputs and Fuzzy Outputs

Method	Nonlinear SVR	Tanaka's fuzzy SVR [13]	Proposed fuzzy SVR by Hao and Chiang [13]	Proposed fuzzy SVR by Hao and Chiang [13]	Our mthod
Kernel	Polynomial of degree 5	Linear	Polynomial of degree 5	Polynomial of degree 3	Gaussian with σ=1
$RMSE$	1.3409	0.7190	0.5519	0.4525	0.299

TABLE 11. The Results of $RMSE$

we consider $y_{il} = y_i - e_i$ and $y_{iu} = y_i + e_i$. Afterwards we apply Gaussian kernel. The combinations of $C = [2^{-5}, 2^{-4}, \cdots, 2^{10}]$, $\sigma = [0.1, 1, 5, 10, 15, 20, 25, 40, 50, 75, 90]$ and $\epsilon = [0.1, 0.2, \cdots, 2]$ form different models. Using our method in nonlinear case, the resulted $RMSE$ is small when $\epsilon = 1.1$, $C = 2^4$ and $\sigma = 1$. The results are shown in Table 11. It is obvious that the resulted $RMSE$ by our method is less than other methods.

6. **Conclusion**

The standard SVR assumes that the training data are known exactly. However frequently in practical regression models, training data containing input and/or output data cannot be observed precisely because of sampling errors, modeling errors or measurement errors. Thus usually they are presented by random variables. Therefore probabilistic constraints SVR play an important role in determination of optimal hyperplane regression with the minimal error. In this case probabilistic constraints achieve robustness.

In this paper, we presented a new SVR with probabilistic constraints which any of output variables is the uniform random variable. Then for handling those randomized training data, bias term using in the model is also set to be uniform random variable. We obtained the optimal hyperplane regression by solving a QP problem. However in many existing studies, probabilistic constrained problem is converted into an SOCP. We know that solving an SOCP is more difficult than solving QP. In the linear case, computing simulated SSE and $RMSE$ using Monte Carlo simulation showed that the ability of our method for obtaining optimal hyperplane regression is more than the linear standard SVR and Arabpour-Tata method [1]. simulated bias and mean squared error of $\widehat{w}$ and $\widehat{\mu}_B$ for our method are very small and decrease when n increases. In addition, values of them are less than standard linear SVR.

Also the Gaussian kernel function was used for function estimation for nonlinear SVR with probabilistic constraints. In this case, computing simulated SSE and $RMSE$ showed that the ability of our method for obtaining optimal hyperplane regression is more than the nonlinear standard SVR.

The proposed method has been applied to the artificial data set, UCI databases and real-world applications [2,13,38,39,40] producing better results than those achieved by the standard SVR, TSVR and weighted TSVR.

References

[1] A. R. Arabpour and M. Tata, *Estimating the parameters of a fuzzy linear regression model*, Iranian Journal of Fuzzy Systems, **5(2)** (2008), 1–19.

[2] K. Bache and M. Lichman, *UCI machine learning repository*, Available on-line at: http://archive.ics.uci.edu/ml/machine-learning-databases, 2013.

[3] A. Ben-Tal, S. Bhadra, C. Bhattacharyya and J. S. Nath, *Chance constrained uncertain classification via robust optimization*, Math. Program., **127(1)** (2011), 145–173.

[4] P. Bosch, J. Lopez, H. Ramirez and H. Robotham, *Support vector machine under uncertainty: an application for hydroacoustic classification of fish-schools in Chile*, Expert Systems with Applications, **40** (2013), 4029–4034.

[5] K. D. Brabanter, J. D. Brabanter, J. A. K. Suykens and B. D. Moor, *Approximate confidence and prediction intervals for least squares support vector regression*, IEEE Transactions on Neural Networks, **22** (2011), 110–120.

[6] E. Carrizosa, J. E. Gordillo and F. Plastria, *Kernel support vector regression with imprecise output*, Dept. MOSI. Vrije Univ. Brussel. Belgium. Tech. Rep., Available on-line at: `http://www.optimization-online.org/DB_FILE/2008/01/1896.pdf`, 2008.

[7] E. Carrizosa, J. E. Gordillo and F. Plastria, *Support vector regression for imprecise data*, Dept. MOSI. Vrije Univ. Brussel. Belgium. Tech. Rep., Available on-line at: `http://www.optimization-online.org/DB_HTML/2007/11/1826.html`, 2007.

[8] J. H. Chiang and P. Y. Hao, *Support vector learning mechanism for fuzzy rule-based modeling: a new approach*, IEEE Trans. Fuzzy Syst., **12(1)** (2004), 1–12.

[9] H. Drucker, Ch. J. C. Burges, L. Kaufman, A. Smola and V. Vapnik, *Support vector regression machines*, Adv. Neural Inform. Process. Syst., **9** (1997), 155–161.

[10] B. Efron, *Bootstrap methods: Another look at the jackknife*, Annals of Statistics, **7** (1979) 1–26.

[11] A. Farag and R. M. Mohamed, *Classification of multispectral data using support vector machines approach for density estimation*, IEEE Seventh International Conference on Intell. Eng. Syst., (2003), 4–6.

[12] J. B. Gao, S. R. Gunn, C. J. Harris and M. Brown, *A probabilistic framework for SVM regression and error bar estimation*, Machine Learning, **46** (2002), 71–89.

[13] P. Y. Hao and J. H. Chiang, *A fuzzy model of support vector regression machine*, International Journal of Fuzzy Systems, **9(1)** (2007), 45–49.

[14] H. P. Huang and Y. H. Liu, *Fuzzy support vector machines for pattern recognition and data mining*, International Journal of Fuzzy Systems, **4** (2002), 826–835.

[15] G. Huang, S. Song, C. Wu and K. You, *Robust support vector regression for uncertain input and output data*, IEEE Transactions on Neural Networks and Learning Systems, **23(11)** (2012), 1690–1700.

[16] R. K. Jayadeva, R. Khemchandani and S. Chandra, *Twin support vector machines for pattern classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **29(5)** (2007), 905–910.

[17] Y. Jinglin, H. X. Li and H. Yong, *A probabilistic SVM based decision system for pain diagnosis*, Expert Systems with Applications, **38** (2011), 9346–9351.

[18] A. F. Karr, *probability*, Springer, New york, (1993), 52–74.

[19] M. A. Kumar and M. Gopal, *Least squares twin support vector machines for pattern classification*, Expert Systems with Applications, **36(4)** (2009), 7535–7543.

[20] J. T. Y. Kwok, *The evidence framework applied to support vector machines*, IEEE Transactions on Neural Networks, **11** (2000), 1162–1173.

[21] G. R. G. Lanckriet, L. E. Ghaoui, Ch. Bhattacharyya and M. I. Jordan, *A robust minimax approach to classification*, J. Mach. Learn. Res., **3** (2002), 555–582.

[22] Y. J. Lee and S. Y. Huang, *Reduced support vector machines: a statistical theory*, IEEE Transactions on Neural Networks, **18** (2007), 1–13.
[23] H. Li, J. Yang, G. Zhang and B. Fan, *Probabilistic support vector machines for classification of noise affected data*, Information Sciences, **221** (2013), 60–71.
[24] C. F. Lin and S. D. Wang, *Fuzzy support vector machine*, IEEE Transactions on Neural Networks, **13** (2002), 464–471.
[25] W. Y. Liu, K. Yue and M. H. Gao, *Constructing probabilistic graphical model from predicate formulas for fusing logical and probabilistic knowledge*, Information Sciences, **181(18)** (2011), 3828–3845.
[26] M. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, *Applications of second-order cone programming*, Linear Algebra Its Appl., **284** (1998), 193–228.
[27] O. L. Mangasarian, *Nonlinear Programming*, McGraw-Hill, New York, (1969), 69–75.
[28] S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM J. Optim., **2** (1992), 575–601.
[29] X. Peng, *TSVR: an efficient twin support vector machine for regression*, Neural Networks., **23(3)** (2010), 365–372.
[30] J. C. Platt, *Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods*, Advances in Large Margin Classifiers, **10(3)** (1999), 61–74.
[31] Z. Qi, Y. Tian and Y. Shi, *Robust twin support vector machine for pattern classification*, Pattern Recognition, **46** (2013), 305–316.
[32] H. Sadoghi Yazdi, S. Effati and Z. Saberi, *The probabilistic constraints in the support vector machine*, App. Math. Comput., **194** (2007), 467–479.
[33] P. K. Shivaswamy, Ch.Bhattacharyya and A.J.Smola, *Second order cone programming approaches for handling missing and uncertain data*,J. Mach. Learn. Res.,**7** (2006), 1283-1314.
[34] P. Sollich, *Bayesian methods for support vector machines: evidence and predictive class probabilities*, Machine Learning, **46** (2002), 21–52.
[35] J. A. K. Suykens and J. Vandewalle, *Least squares support vector machine classifiers*, Neural Processing Letters, **9(3)** (1999), 293–300.
[36] T. B. Trafalis and S. A. Alwazzi, *Support vector regression with noisy data: a second order cone programming approach*, Int. J. General Syst., **36** (2007), 237–250.
[37] T. B. Trafalis and R. C. Gilbert, *Robust classification and regression using support vector machines*, Eur. J. Oper. Res., **173(3)** (2006), 893–909.
[38] URL http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/regression.html.
[39] URL http://www.dcc.fc.up.pt/ ltorgo/Regression/DataSets.html.
[40] URL `http://www.itl.nist.gov/div898/strd/nls/nls_main.shtml`.
[41] V. Vapnik, *The nature of statistical learning theory*, Springer-Verlag, New York, (1995), 123-146, 181-186.
[42] V. Vapnik, S. Golowich and A. Smola, *Support vector method for multivariate density estimation*, Adv. Neural Inform. Process. Syst., **12** (1999), 659–665.
[43] Y. Xu and L. Wang, *A weighted twin support vector regression*, Knowledge-Based Syst., **33** (2012), 92–101.
[44] Y. Xu, W. Xi, X. Lv and R. Guo, *An improved least squares twin support vector machine*, Journal of information and computational science, **9(4)** (2012), 1063–1071.
[45] X. Yang, L. Tan and L. He, *A robust least squares support vector machine for regression and classification with noise*, Neurocomputing, **140** (2014), 41–52.

Maryam Abaszade, Department of Statistics, Ferdowsi University of Mashhad, Mashhad, Iran
E-mail address: `m_abaszade@yahoo.com`

Sohrab Effati*, Department of Applied Mathematics, Ferdowsi University of Mashhad, Mashhad, Iran
E-mail address: `s-effati@um.ac.ir`

*Corresponding author