Neural Networks 57 (2014) 1–11

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

# Noise model based $\nu$-support vector regression with its application to short-term wind speed forecasting

Qinghua Hu[a,b,*], Shiguang Zhang[a,c], Zongxia Xie[d], Jusheng Mi[a,e], Jie Wan[f]

[a] *College of Mathematics and Information Science, Hebei Normal University, Shijiazhuang, Hebei, 050024, China*
[b] *School of Computer Science and Technology, Tianjin University, Tianjin, 300072, China*
[c] *College of Mathematics and Computer Science, Hengshui University, Hengshui, Hebei, 053000, China*
[d] *School of Computer Software, Tianjin University, Tianjin, 300072, China*
[e] *Hebei Key Laboratory of Computational Mathematics and Applications, Shijiazhuang, Hebei, 050024, China*
[f] *School of Energy Science and Engineering, Harbin Institute of Technology, Harbin, 150001, China*

ARTICLE INFO

*Article history:*
Received 17 September 2013
Received in revised form 9 March 2014
Accepted 1 May 2014
Available online 13 May 2014

*Keywords:*
Support vector regression
Noise model
Loss function
Inequality constraints
Wind speed forecasting

ABSTRACT

Support vector regression (SVR) techniques are aimed at discovering a linear or nonlinear structure hidden in sample data. Most existing regression techniques take the assumption that the error distribution is Gaussian. However, it was observed that the noise in some real-world applications, such as wind power forecasting and direction of the arrival estimation problem, does not satisfy Gaussian distribution, but a beta distribution, Laplacian distribution, or other models. In these cases the current regression techniques are not optimal. According to the Bayesian approach, we derive a general loss function and develop a technique of the uniform model of $\nu$-support vector regression for the general noise model (N-SVR). The Augmented Lagrange Multiplier method is introduced to solve N-SVR. Numerical experiments on artificial data sets, UCI data and short-term wind speed prediction are conducted. The results show the effectiveness of the proposed technique.

## 1. Introduction

Regression is an old topic in the domain of learning functions from a set of samples (Hastie, Tibshirani, & Friedman, 2009). It provides researchers and engineers with a powerful tool to extract hidden rules of data. The trained model is used to predict future events with the information of past or present events. Regression analysis is now successfully applied in nearly all fields of science and technology, including the social sciences, economics, finance, wind power prediction for grid operation. However this domain is still attracting much attention from research and application domains.

Generally speaking, there are three important issues in designing a regression algorithm: model structures, objective functions and optimization strategies. The model structures include linear or nonlinear functions (Park & Lee, 2005), neural networks (Spech, 1990), decision trees (Esposito, Malerba, & Semeraro, 1997), and so on; optimization objectives include $\epsilon$-insensitive loss (Cortes & Vapnik, 1995; Vapnik, 1995; Vapnik, Golowich, & Smola, 1996), squared loss (Suykens, Lukas, & Vandewalle, 2000; Wu, 2010; Wu & Law, 2011), robust Huber loss (Olvi & David, 2000) etc. According to the formulation of optimization functions, a collection of optimization algorithms (Ma, 2010) have been developed. In this work, we focus on the problem which optimal formulation should be considered with respect to different error models.

Suppose we are given a set of training data

$$D_l = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}, \tag{1}$$

where $x_i \in R^L, y_i \in R, \; i = 1, 2, \ldots, l$. Take a multivariate linear regression task $f$ as an example. The form is

$$f(x) = \omega^T \cdot x + b, \tag{2}$$

where $\omega \in R^L, b \in R, \; i = 1, 2, \ldots, l$. The task is to learn the parameter vectors $\omega$ and parameter $b$, by minimizing the objective function

$$g_{LR} = \sum_{i=1}^{l} (y_i - \omega^T \cdot x_i - b)^2. \tag{3}$$

[*] Corresponding author at: College of Mathematics and Information Science, Hebei Normal University, Shijiazhuang, 050024, China. Tel.: +86 22 27401839.
*E-mail address:* huqinghua@tju.edu.cn (Q. Hu).

http://dx.doi.org/10.1016/j.neunet.2014.05.003
0893-6080/© 2014 Elsevier Ltd. All rights reserved.

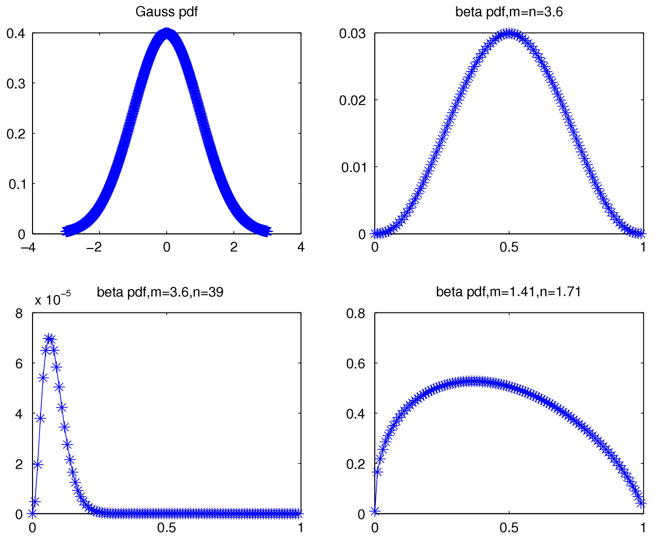

**Fig. 1.** Gaussian PDF and beta PDF of parameters.

The objective function of the sum-of-squares error is usually used in regression. The trained model is optimal, if the samples have been corrupted by independent and identical probability distributions (i.i.d.). Noise satisfying Gaussian distribution with zeros mean and variance $\sigma^2$, i.e., $y_i = f(x_i) + \xi_i,\ i = 1, \ldots, l,\ \xi_i \sim N(0, \sigma^2)$.

In the recent years, the support vector regressor (SVR) is growing up as a popular technique (Cortes & Vapnik, 1995; Cristianini & Shawe, 2000; Smola & Schölkopf, 2004; Vapnik, 1995, 1998, 1999; Vapnik et al., 1996; Wu, 2010; Wu & Law, 2011). It is a universal regression machine based on the V–C dimension theory. This technique is developed with the Structural Risk Minimization (SRM) principle, which has shown its effectiveness in applications. The classical SVR is optimized by minimizing Vapnik's $\epsilon$-insensitive loss function of residuals and has achieved good performance in a variety of practical applications (Bayro-Corrochano & Arana-Daniel, 2010; Duan, Xu, & Tsang, 2012; Huang, Song, Wu, & You, 2012; Kwok & Tsang, 2003; Lopez & Dorronsoro, 2012; Yang & Ong, 2011).

In 1995, $\epsilon$-SVR was proposed by Vapnik and his research team (Cortes & Vapnik, 1995; Vapnik, 1995; Vapnik et al., 1996). In 2000, $\nu$-SVR was introduced by Schölkopf, Smola, Williamson, and Bartlett (2000), which automatically computes $\epsilon$. Suykens et al. (2000) constructed least squares support vector regression with Gaussian noise (LS-SVR). Wu (Wu, 2010; Wu & Law, 2011) and Pontil, Mukherjee, and Girosi (1998) constructed $\nu$-support vector regression with Gaussian noise (GN-SVR). If the noise obeys the Gaussian distribution, the outputs of the models are optimal. However, it was found that the noise in some real-world applications, just like wind power forecast and direction-of-arrival estimation problem, does not satisfy Gaussian distribution, but a beta distribution or Laplace distribution, respectively. In these cases these regression techniques are not optimal.

The principle of $\nu$-support vector regression ($\nu$-SVR) can be written as (Chalimourda, Schölkopf, & Smola, 2004; Chih-Chung & Chih-Jen, 2002; Schölkopf et al., 2000):

$$\min\left\{g_{P_{\nu\text{-SVR}}} = \frac{1}{2}\|\omega\|^2 + C\cdot\left(\nu\epsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right)\right\}$$

$$\begin{aligned}&\text{Subject to :}\\ &\omega^T\cdot x_i + b - y_i \le \epsilon + \xi_i\\ &y_i - \omega^T\cdot x_i - b \le \epsilon + \xi_i^*\\ &\xi_i, \xi_i^* \ge 0, \quad i = 1, 2, \ldots, l,\ \epsilon \ge 0,\end{aligned} \qquad (4)$$

where $\xi_i, \xi_i^*$ are two slack variables. The constant $C > 0$ determines the trade-off between the flatness of $f$ and the amount up to which deviations larger than $\epsilon$ are tolerated. $\nu \in (0, 1]$ is a constant which controls the number of support vectors. In the $\nu$-SVR the size of $\epsilon$ is not given a priori but a variable. Its value is traded off against the model complexity and slack variables via a constant $\nu$ (Chalimourda et al., 2004). This corresponds to dealing with a so-called $\epsilon$-insensitive loss function (Cortes & Vapnik, 1995; Vapnik, 1995) described by

$$c_\epsilon(\xi) = |\xi|_\epsilon = \begin{cases}0, & \text{if } |\xi| \le \epsilon,\\ |\xi| - \epsilon, & \text{otherwise.}\end{cases} \qquad (5)$$

In 2002, $\epsilon$-SVR for a general noise model was proposed in Schölkopf and Smola (2002):

$$\min\left\{g_{\epsilon\text{-SVR}} = \frac{1}{2}\|\omega\|^2 + C\cdot\left(\sum_{i=1}^{l}\widetilde{c}(\xi_i) + \widetilde{c}(\xi_i^*)\right)\right\}$$

$$\begin{aligned}&\text{Subject to :}\\ &\omega^T\cdot x_i + b - y_i \le \epsilon + \xi_i\\ &y_i - \omega^T\cdot x_i - b \le \epsilon + \xi_i^*\\ &\xi_i, \xi_i^* \ge 0, \quad i = 1, 2, \ldots, l,\end{aligned} \qquad (6)$$

where $c(x, y, f(x)) = \widetilde{c}(|y - f(x)|_\epsilon)$ is a general convex loss function in the sample point $(x_i, y_i)$ of $D_l$. $|y - f(x)|_\epsilon$ in (5) is Vapnik's $\epsilon$-insensitive loss function.

Using Lagrange multiplier techniques (Cortes & Vapnik, 1995; Vapnik, 1995), Problem (4) can be transformed to a convex optimization problem with a global minimum. At the optimum, the regression estimate takes the form $f(x) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)(x_i \cdot x) + b$, where $(x_i \cdot x)$ is the inner product.

In 2002, Bofinger, Luig, and Beyer (2002) found that the output of wind turbine systems is limited between zero and the maximum power and the error statistics do not follow a normal distribution. In 2005, Fabbri, Román, Abbad, and Quezada (2005) believed that the normalized produced power $p$ must be within the interval $[0, 1]$ and the beta function is more appropriate to fit the error than the standard normal distribution function. Bludszuweit, Antonio, and Llombart (2008) showed the advantages of using the beta probability distribution function (PDF), instead of the Gaussian PDF, for approximating the forecast error distribution. The error $\epsilon$ between the predicted values $x_p$ and the measured values $x_m$ obeys the beta distribution in the forecast of wind power, and the PDF of $\epsilon$ is $f(\epsilon) = \epsilon^{m-1}\cdot(1-\epsilon)^{n-1}\cdot h$, $\epsilon \in (0, 1)$, the parameters $m$ and $n$ are often called hyperparameters because they control the distribution of the variable $\epsilon$ ($m > 1, n > 1$), $h$ is the normalization factor and parameters $m$ and $n$ are determined by the values of the mean (which is the predicted power) and the standard deviation (Bishop, 2006; Canavos, 1984). Fig. 1 shows plots of Gaussian distribution and the beta distribution for different values of hyperparameters. In 2007, Zhang, Wan, Zhao, and Yang (2007) and Randazzo, Abou-Khousa, Pastorino, and Zoughi (2007) presented the estimation results under a Laplacian noise environment in the direction-of-arrival of coherent electromagnetic waves impinging estimation problem. Laplace distribution is frequently encountered in various machine learning areas, e.g., the over-complete wavelet transform coefficients of images, processing in Natural images, etc. (Eltoft, Kim, & Lee, 2006; Park & Lee, 2005).

Based on the above analysis, we know that the error distributions do not satisfy Gaussian distribution in some real-world applications. We try to study the optimal loss functions for different error models.

It is not suitable to apply the GN-SVR to fit functions from data with non-Gaussian noise. In order to solve the above problems, we derive a general loss function and construct $\nu$-support vector regression machines for a general noise model.

Finally, we design a technique to find the optimal solution to the corresponding regression tasks. While there are a large number of implementations of SVR algorithms in the past years, we introduce the Augmented Lagrange Multiplier (ALM) method, presented in Section 4. If the task is non-differentiable or discontinuous, the sub-gradient descent method can be used (Ma, 2010), and if there are very large scale of samples, SMO can also be used (Shevade, Keerthi, Bhattacharyya, & Murthy, 2000).

The main contributions of our work are listed as follows: (1) we derive the optimal loss functions for different error models by the use of Bayesian approach and optimization theory; (2) we develop the uniform $\nu$-support vector regression model for the general noise with inequality constraints ($N$-SVR); (3) the Augmented Lagrange Multiplier method is applied to solve $N$-SVR, which guarantees the stability and validity of the solution in $N$-SVR; (4) we utilize $N$-SVR to short-term wind speed prediction and show the effectiveness of the proposed model in practical applications.

This paper is organized as follows: in Section 2 we derive the optimal loss function corresponding to a noise model by using the Bayesian approach; in Section 3 we describe the proposed $\nu$-support vector regression technique for general noise model ($N$-SVR); in Section 4 we give the solution and algorithm design of $N$-SVR; numerical experiments are conducted on artificial data sets, UCI data and short-term wind speed prediction in Sections 5 and 6; finally, we conclude the work in Section 7.

## 2. Bayesian approach to the general loss function

Given a set of noisy training samples $D_l$, we require to estimate an unknown function $f(x)$. Following Chu, Keerthi, and Ong (2004); Girosi (1991); Klaus-Robert and Sebastian (2001); Pontil et al. (1998), the general approach is to minimize

$$H[f] = \sum_{i=1}^{l} c(\xi_i) + \lambda \cdot \Phi[f], \tag{7}$$

where $c(\xi_i) = c(y_i - f(x_i))$ is a loss function, $\lambda$ is a positive number and $\Phi[f]$ is a smoothness functional.

We assume the noise is additive

$$y_i = f(x_i) + \xi_i, \quad i = 1, 2, \dots, l, \tag{8}$$

where $\xi_i$ is random, independent, identical probability distributions (i.i.d.) with $P(\xi_i)$ of variance $\sigma$ and mean $\mu$. We want to estimate the function $f(x)$ with the set of data $D_f \subseteq D_l$. We take a probabilistic approach, and regard the function $f$ as the realization of a random field with a known prior probability distribution. We are interested in maximizing the posteriori probability of $f$ given the data $D_f$, namely $P[f|D_f]$, which can be written as

$$P[f|D_f] \propto P[D_f|f] \cdot P[f], \tag{9}$$

where $P[D_f|f]$ is the conditional probability of the data $D_f$ given the function $f$ and $P[f]$ is a priori probability of the random field $f$, which is often written as $P[f] \propto \exp(-\lambda \cdot \Phi[f])$, where $\Phi[f]$ is a smoothness functional. The probability $P[D_f|f]$ is essentially a model of the noise, and if the noise is additive, as in Eq. (8) and i.i.d. with probability distribution $P(\xi_i)$, it can be written as

$$P[D_f|f] = \prod_{i=1}^{l} P(\xi_i). \tag{10}$$

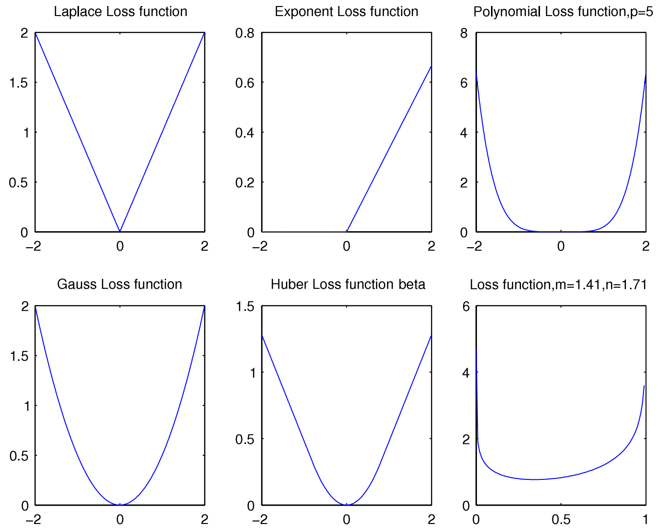

**Fig. 2.** Loss function of the corresponding noise model.

Substituting $P[f]$ and Eq. (10) in (9), we see that the function that maximizes the posterior probability of $f$ is the one which minimizes the following functional

$$\begin{aligned} H[f] &= -\sum_{i=1}^{l} \log[P(y_i - f(x_i)) \cdot e^{-\lambda \cdot \Phi[f]}] \\ &= -\sum_{i=1}^{l} \log P(y_i - f(x_i)) + \lambda \cdot \Phi[f]. \end{aligned} \tag{11}$$

This functional is of the same form as Eq. (7) (Girosi, 1991; Pontil et al., 1998). By Eqs. (7) and (11), the optimal loss function in a maximum likelihood sense is

$$c(x, y, f(x)) = -\log p(y - f(x)), \tag{12}$$

i.e. the loss function $c(\xi)$ is the log-likelihood of the noise.

We assume that the noise in Eq. (8) is Gaussian, with zero mean and variance $\sigma$. By Eq. (12), the loss function corresponding to Gaussian-noise is

$$c(\xi_i) = \frac{1}{2\sigma^2}(y_i - f(x_i))^2. \tag{13}$$

If the noise in Eq. (8) is beta, with mean $\mu \in (0, 1)$ and variance $\sigma$, get $m = (1-\mu)\cdot\mu^2/\sigma^2 - \mu$, $n = ((1-\mu)/\mu)\cdot m$ $(m > 1, n > 1)$, and $h = \Gamma(m + n)/(\Gamma(m) \cdot \Gamma(n))$ is the normalization factor (Bofinger et al., 2002). By Eq. (12), the loss function corresponding to beta-noise is

$$c(\xi_i) = (1 - m)\log(\xi_i) + (1 - n)\log(1 - \xi_i), \tag{14}$$

where parameters $m > 1$, $n > 1$.

And if the noise in Eq. (8) is Weibull, with parameters $\theta$ and $k$. By Eq. (12), the loss function should be

$$c(\xi_i) = \begin{cases} (1-k)\log\dfrac{\xi_i}{\theta} + \left(\dfrac{\xi_i}{\theta}\right)^k, & \text{if } \xi_i \geq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The loss functions and their corresponding probability density functions (PDF) of the noise models used in regression problems are listed in Table 1 and shown in Fig. 2.

## 3. Noise model based $\nu$-support vector regression

Given samples $D_l$, we construct a linear regression function $f(x) = \omega^T \cdot x + b$. In order to deal with nonlinear functions the

**Table 1**
Loss function and PDF of the corresponding noise model.

| Function case | PDF of the noise model | Loss function |
|---|---|---|
| Laplace | $P(\xi_i) = \frac{1}{2}e^{-\lvert\xi_i\rvert}$ | $c(\xi_i) = \lvert\xi_i\rvert$ |
| $\epsilon$-insensitive | $P(\xi_i) = \begin{cases} \dfrac{1}{2(1+\epsilon)}, & \text{if } \lvert\xi\rvert \le \epsilon, \\ \dfrac{1}{2(1+\epsilon)}e^{\epsilon-\lvert\xi_i\rvert}, & \text{otherwise.} \end{cases}$ | $c_\epsilon(\xi_i) = \lvert\xi_i\rvert_\epsilon$ |
| Exponent | $P(\xi_i) = \begin{cases} \dfrac{1}{\theta}e^{-\frac{\xi_i}{\theta}}, & \text{if } \xi_i \ge 0, \\ 0, & \text{otherwise.} \end{cases}$ | $c(\xi_i) = \begin{cases} \dfrac{\xi_i}{\theta}, & \text{if } \xi_i \ge 0, \\ 0, & \text{otherwise.} \end{cases}$ |
| Polynomial | $P(\xi_i) = \frac{p}{2\cdot\Gamma(\frac{1}{p})}e^{-\frac{\lvert\xi_i\rvert^p}{p}}$ | $c(\xi_i) = \frac{1}{p}\lvert\xi_i\rvert^p$ |
| Gauss | $c(\xi_i) = \frac{1}{2}\xi_i^2$ | $P(\xi_i) = \frac{1}{\sqrt{2\cdot\pi}}e^{-\frac{\xi_i^2}{2}}$ |
| Huber | $P(\xi_i) = \begin{cases} e^{-\frac{\xi_i^2}{2}}, & \text{if} \xi_i \le \epsilon, \\ e^{\frac{\epsilon^2}{2}-\epsilon\lvert\xi_i\rvert}, & \text{otherwise.} \end{cases}$ | $c(\xi_i) = \begin{cases} \dfrac{\xi_i^2}{2}, & \text{if } \xi_i \le \epsilon, \\ \epsilon\lvert\xi_i\rvert - \dfrac{\epsilon^2}{2}, & \text{otherwise.} \end{cases}$ |
| beta | $P(\xi_i) = \xi_i^{m-1}\cdot(1-\xi_i)^{n-1}\cdot h, h = \frac{\Gamma(m+n)}{\Gamma(m)\cdot\Gamma(n)}$ | $c(\xi_i) = (1-m)\log(\xi_i) + (1-n)\log(1-\xi_i)$ |
| Weibull | $P(\xi_i) = \begin{cases} \dfrac{k}{\theta}\left(\dfrac{x}{\theta}\right)^{k-1}e^{-(\frac{\xi_i}{\theta})^k}, & \text{if } \xi_i \ge 0, \\ 0, & \text{otherwise.} \end{cases}$ | $c(\xi_i) = \begin{cases} (1-k)\log\dfrac{\xi_i}{\theta} + \left(\dfrac{\xi_i}{\theta}\right)^k, & \text{if } \xi_i \ge 0, \\ 0, & \text{otherwise.} \end{cases}$ |

following generalization can be done (Schölkopf & Smola, 2002; Vapnik, 1995, 1998): we map the input vectors $x_i \in R^L$ into a high-dimensional feature space $H$ through some nonlinear mapping, $\Phi : R^L \to H$ ($H$ is Hilbert space), chosen a priori. We then solve the optimization problem (4) in the feature space $H$. In this case, the inner product of the input vectors $(x_i \cdot x)$ is replaced by the inner product of their icons in feature space $H(\Phi(x_i)\cdot\Phi(x_j))$. By using a function $K(\bullet,\bullet)$, the linear model is extended to a nonlinear support vector regression machine

$$K(x_i, x_j) = (\Phi(x_i)\cdot\Phi(x_j)). \tag{16}$$

We develop a technique of the uniform model of $\nu$-support vector regression for the general noise model. The primal problem of $N$-SVR is described as

$$\min\left\{g_{P_{N\text{-SVR}}} = \frac{1}{2}\omega^T\cdot\omega + C\cdot\left(\nu\epsilon + \frac{1}{l}\sum_{i=1}^{l}(c(\xi_i) + c(\xi_i^*))\right)\right\}$$

$$\begin{aligned} &\text{Subject to :} \\ &\quad \omega^T\cdot\Phi(x_i) + b - y_i \le \epsilon + \xi_i \\ &\quad y_i - \omega^T\cdot\Phi(x_i) - b \le \epsilon + \xi_i^* \\ &\quad \xi_i, \xi_i^* \ge 0, \quad i = 1, 2, \ldots, l, \epsilon \ge 0, \end{aligned} \tag{17}$$

$c(\xi_i), c(\xi_i)^* \ge 0$ $(i = 1, 2, \ldots, l)$ are general convex loss functions in the sample point $(x_i, y_i) \in D_l$. $C > 0$ is a penalty parameter.

**Theorem 1.** *The solution of the primal problem* (17) *of $N$-SVR about $\omega$ exists and is unique.*

**Proof.** It is trivial to prove the existence of the solution. The uniqueness of the solution certificate as follows. There are two solutions $(\overline{\varpi}, \overline{b}, \overline{\xi})$ and $(\widetilde{\varpi}, \widetilde{b}, \widetilde{\xi})$ in the primal problem (17). Define $(\omega, b, \xi)$ as

$$\omega = \frac{1}{2}(\overline{\varpi} + \widetilde{\varpi}), \qquad b = \frac{1}{2}(\overline{b} + \widetilde{b}), \qquad \xi = \frac{1}{2}(\overline{\xi} + \widetilde{\xi}). \tag{18}$$

We have

$$\begin{aligned} y_i - \omega^T\cdot\Phi(x_i) - b - \xi_i &= y_i - \frac{1}{2}(\overline{\varpi} + \widetilde{\varpi})^T\cdot\Phi(x_i) \\ &\quad - \frac{1}{2}(\overline{b} + \widetilde{b}) - \frac{1}{2}(\overline{\xi_i} + \widetilde{\xi_i}) \\ &= \frac{1}{2}(y_i - \overline{\varpi}^T\cdot\Phi(x_i) - \overline{b} - \overline{\xi}_i) \\ &\quad + \frac{1}{2}(y_i - \widetilde{\varpi}^T\cdot\Phi(x_i) - \widetilde{b} - \widetilde{\xi_i}) \\ &= 0, \end{aligned} \tag{19}$$

where $\xi_i = \frac{1}{2}(\overline{\xi_i} + \widetilde{\xi_i}) \ge 0$, $i = 1, \ldots, l$. Suppose $(\omega, b, \xi)$ is a feasible solution of the primal problem (17). Then we have

$$\frac{1}{2}\|\omega\|^2 + \frac{C}{l}\sum_{i=1}^{l}c(\xi_i) \ge \frac{1}{2}\|\overline{\varpi}\|^2 + \frac{C}{l}\sum_{i=1}^{l}c(\overline{\xi}_i), \tag{20}$$

$$\frac{1}{2}\|\omega\|^2 + \frac{C}{l}\sum_{i=1}^{l}c(\xi_i) \ge \frac{1}{2}\|\widetilde{\varpi}\|^2 + \frac{C}{l}\sum_{i=1}^{l}c(\widetilde{\xi_i}). \tag{21}$$

By (20) and (21), we have $2\|\omega\|^2 \ge \|\overline{\varpi}\|^2 + \|\widetilde{\varpi}\|^2$. Substitute $2\omega = \overline{\varpi} + \widetilde{\varpi}$ into the above inequalities

$$\|\overline{\varpi} + \widetilde{\varpi}\|^2 \ge 2(\|\overline{\varpi}\|^2 + \|\widetilde{\varpi}\|^2). \tag{22}$$

As $\|\overline{\varpi} + \widetilde{\varpi}\| \le \|\overline{\varpi}\| + \|\widetilde{\varpi}\|$, by (22) we have

$$(\|\overline{\varpi}\| + \|\widetilde{\varpi}\|)^2 \ge \|\overline{\varpi} + \widetilde{\varpi}\|^2 \ge 2(\|\overline{\varpi}\|^2 + \|\widetilde{\varpi}\|^2). \tag{23}$$

As $2\|\overline{\varpi}\|\cdot\|\widetilde{\varpi}\| \le \|\overline{\varpi}\|^2 + \|\widetilde{\varpi}\|^2$, by (23) we derive

$$\begin{aligned} 2\|\overline{\varpi}\|\cdot\|\widetilde{\varpi}\| &= \|\overline{\varpi}\|^2 + \|\widetilde{\varpi}\|^2, \qquad \|\overline{\varpi}\| = \|\widetilde{\varpi}\|, \qquad \|\overline{\varpi} + \widetilde{\varpi}\| \\ &= \|\overline{\varpi}\| + \|\widetilde{\varpi}\|. \end{aligned}$$

Thus $\widetilde{\varpi} = \lambda\cdot\overline{\varpi}$. Therefore $\lambda = 1$ or $\lambda = -1$. If $\lambda = -1$, $\overline{\varpi} + \widetilde{\varpi} = 0$. Thus $\|\overline{\varpi} + \widetilde{\varpi}\| = \|\overline{\varpi}\| + \|\widetilde{\varpi}\| = 0$. So $\|\overline{\varpi}\| = \|\widetilde{\varpi}\| = 0$, namely, $\overline{\varpi} = \widetilde{\varpi} = 0$. If $\lambda = 1$, $\overline{\varpi} = \widetilde{\varpi}$.

To sum up, the solution of the primal problem (17) about $\omega$ exists and is unique.

**Theorem 2.** *The dual problem of the primal problem* (17) *of* $N$-SVR *is*

$$\max\left\{g_{D_{N\text{-SVR}}} = -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^* - \alpha_i)\cdot(\alpha_j^* - \alpha_j)\cdot K(x_i, x_j) + \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)\cdot y_i + \frac{C}{l}\sum_{i=1}^{l}(T(\xi_i(\alpha_i)) + T(\xi_i^*(\alpha_i^*)))\right\}$$

Subjected to

$$\begin{aligned}&\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0\\ &0 \le \alpha_i^{(*)} \le \frac{C}{l}, \quad i = 1, 2, \ldots, l\\ &\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) \le C\cdot\nu, \quad i = 1, 2, \ldots, l\end{aligned} \tag{24}$$

*where* $T(\xi_i^{(*)}(\alpha_i^{(*)})) = c(\xi_i^{(*)}(\alpha_i^{(*)})) - \xi_i^{(*)}(\alpha_i^{(*)})\cdot\frac{\partial(c(\xi_i^{(*)}(\alpha_i^{(*)})))}{\partial(\xi_i^{(*)}(\alpha_i^{(*)}))}$, $C > 0$, $0 < \nu \le 1$ *is constant.*

**Proof.** We introduce the Lagrange functional $L(\omega, b, \alpha, \alpha^*, \xi, \xi^*, \eta, \eta^*, \epsilon)$ as

$$\begin{aligned}L = {}&\frac{1}{2}\omega^T\cdot\omega + C\cdot\left(\nu\epsilon + \frac{1}{l}\sum_{i=1}^{l}(c(\xi_i) + c(\xi_i^*))\right) - \gamma\epsilon\\ &-\sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) - \sum_{i=1}^{l}\alpha_i(\xi_i + y_i - \omega^T\cdot\Phi(x_i) - b + \epsilon)\\ &-\sum_{i=1}^{l}\alpha_i^*(\xi_i^* - y_i + \omega^T\cdot\Phi(x_i) + b + \epsilon).\end{aligned}$$

To minimize $L$, we derive partial derivatives $\omega, b, \xi, \xi^*, \epsilon$, respectively. According to the KKT (Karush–Kuhn–Tucker) conditions, we have

$$\nabla_\omega(L) = 0, \qquad \nabla_b(L) = 0, \qquad \nabla_\epsilon(L) = 0, \qquad \nabla_{\xi^{(*)}}(L) = 0.$$

And we get

$$\begin{aligned}&\omega_i = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)\cdot\Phi(x_i), \qquad \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0,\\ &C\cdot\nu - \gamma - \sum_{i=1}^{l}(\alpha_i^* + \alpha_i) = 0,\\ &\frac{C}{l}\cdot\frac{\partial(c(\xi_i^{(*)}))}{\partial(\xi_i^{(*)})} - \eta_i^{(*)} - \alpha_i^{(*)} = 0.\end{aligned}$$

Substituting the extreme conditions into $L$ and seeking maximum of $\alpha, \alpha^*$, we obtain the dual problem (24) of the primal problem (17).

We have

$$\begin{aligned}\omega_i &= \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot\Phi(x_i),\\ b &= \frac{1}{2l}\left[\sum_{j=1}^{l}\left(y_j - \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot K(x_i, x_j)\right) + \sum_{k=1}^{l}\left(y_k - \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot K(x_i, x_k)\right)\right].\end{aligned}$$

To estimate $\epsilon$ for

$$\epsilon = \frac{1}{l}\sum_{j=1}^{l}\left(\sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot K(x_i, x_j) + b - y_j\right),$$

or

$$\epsilon = \frac{1}{l}\sum_{k=1}^{l}\left(y_k - \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot K(x_i, x_k) - b\right).$$

Then the decision-making function of $\nu$-support vector regression with inequality constraints ($N$-SVR) can be written as

$$f(x) = \omega^T\cdot\Phi(x) + b = \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)K(x_i, x) + b,$$

where RSVs are the samples corresponding to $\alpha_i^* - \alpha_i \ne 0$ (called support vectors).

The dual problem (24) of $\nu$-SVR, GN-SVR, HN-SVR and BN-SVR is described as follows.

(1) $\nu$-SVR: In Chalimourda et al. (2004), Chih-Chung and Chih-Jen (2002) and Spech (1990), the loss function for sample point $(x_i, y_i) \in D_l$ is $c(\xi_i) = \xi_i$, the dual problem of $\nu$-SVR is

$$\max\left\{g_{D_{\nu\text{-SVR}}} = -\frac{1}{2}\sum_{i\in\text{RSV}}\sum_{j\in\text{RSV}}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) + \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot y_i\right\}$$

Subject to

$$\begin{aligned}&\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0\\ &0 \le \alpha_i^{(*)} \le \frac{C}{l}, \quad i = 1, \ldots, l\\ &\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \le C\cdot\nu, \quad i = 1, \ldots, l.\end{aligned} \tag{25}$$

(2) Gaussian noise model GN-SVR: Suykens et al. (2000); Wu (2010); Wu and Law (2011) studied the support vector regression machine with equality constraints, inequality constraints, respectively. The function corresponding to Gaussian noise is $c(\xi_i) = \xi_i^2/2$. Thus the dual problem for the Gaussian noise model is

$$\max\left\{g_{D_{\text{GN-SVR}}} = -\frac{1}{2}\sum_{i\in\text{RSV}}\sum_{j\in\text{RSV}}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) + \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)y_i - \frac{l}{2C}\sum_{i=1}^{l}(\alpha_i^2 + (\alpha_i^*)^2)\right\}$$

Subject to

$$\begin{aligned}&\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0\\ &0 \le \alpha_i^{(*)} \le \frac{C}{l}, \quad i = 1, \ldots, l\\ &\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \le C\cdot\nu, \quad i = 1, \ldots, l.\end{aligned} \tag{26}$$

(3) HN-SVR for the Robust model: The robust Huber loss function is $c(\xi_i) = \begin{cases}\frac{\xi_i^2}{2}, & \text{if } \xi_i \le \epsilon,\\ \epsilon|\xi_i| - \frac{\epsilon^2}{2}, & \text{otherwise.}\end{cases}$ (Huber, 1964, 1981; Olvi &

David, 2000; Vapnik, 1995). The dual problem of HN-SVR is

$$\max\left\{g_{D_{\text{HN-SVR}}} = -\frac{1}{2}\sum_{i\in\text{RSV}}\sum_{j\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot(\alpha_j^* - \alpha_j)\cdot K(x_i, x_j) + \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot y_i - \frac{l}{2C}\sum_{i=1}^{l}(\alpha_i^2 + (\alpha_i^*)^2)\right\}$$

Subject to

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{27}$$

$$0 \le \alpha_i^{(*)} \le \frac{C\epsilon}{l}, \quad i = 1, \dots, l$$

$$\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \le C\cdot\nu, \quad i = 1, \dots, l.$$

(4) SVR for beta noise (BN-SVR): The loss function for beta noise is $c(\xi_i) = (1-m)\log(\xi_i) + (1-n)\log(1-\xi_i)$, $0 < \xi_i < 1$, $m > 1$, $n > 1$. On account of $\frac{\partial(c(\xi_i))}{\partial(\xi_i)} = \frac{1-\alpha}{\xi_i} - \frac{1-\beta}{1-\xi_i}$, by $\frac{C}{l}\frac{\partial(c(\xi_i))}{\partial(\xi_i)} - \alpha_i = 0$, we have $(\alpha_i l/C)\cdot\xi_i^2 - (2 + \alpha_i l/C - m - n)\cdot\xi_i + 1 - m = 0$. We derive

$$\xi_{i1}(\alpha_i) = \frac{2 + \alpha_i l/C - m - n + \Delta^{\frac{1}{2}}}{2\alpha_i l/C},$$

$$\xi_{i2}(\alpha_i) = \frac{2 + \alpha_i l/C - m - n - \Delta^{\frac{1}{2}}}{2\alpha_i l/C},$$

where $\Delta = (\alpha_i l/C + m - n)^2 + 4(1 + mn - m - n)$, $m > 1$, $n > 1$. As $0 < \xi_i(\alpha_i) < 1$, we reject $\xi_{i2}(\alpha_i)$ and adopt $\xi_{i1}(\alpha_i)$, let $\xi_i(\alpha_i) = \xi_{i1}(\alpha_i)$.

Analogically, let

$$\xi_i^*(\alpha_i^*) = \frac{2 + \alpha_i^* l/C - m - n + \Delta^{\frac{1}{2}}}{2\alpha_i^* l/C},$$

where $\Delta = (\alpha_i^* l/C + m - n)^2 + 4(1 + mn - m - n)$, $m > 1$, $n > 1$.

Then the dual problem of BN-SVR is

$$\max\left\{g_{D_{\text{BN-SVR}}} = -\frac{1}{2}\sum_{i\in\text{RSV}}\sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) + \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)y_i - \sum_{i=1}^{l}(\alpha_i\xi_i(\alpha_i)) - \sum_{i=1}^{l}(\alpha_i^*\xi_i^*(\alpha_i^*)) + C\sum_{i=1}^{l}((1-m)\log(\xi_i(\alpha_i)) + (1-n)\log(\xi_i(1-\alpha_i))) + C\sum_{i=1}^{l}((1-m)\log(\xi_i^*(\alpha_i^*)) + (1-n)\log(\xi_i^*(1-\alpha_i^*)))\right\} \tag{28}$$

Subjected to

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0$$

$$0 \le \alpha_i^{(*)} \le \frac{C}{l}, \quad i = 1, \dots, l$$

$$\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \le C\cdot\nu, \quad i = 1, \dots, l.$$

## 4. Solution based on the Augmented Lagrange Multiplier method

Theorems 1 and 2 supply an algorithm to effectively recognize the model of $N$-SVR. We obtain the solution based on the Augmented Lagrange Multiplier (ALM) method and the algorithm design of $\nu$-support vector regression machine with the noise model ($N$-SVR) in this section.

(1) Let data set $D_l = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$, where $x_i \in R^L$, $y_i \in R$, $i = 1, \dots, l$.

(2) Use a 10-fold cross validation strategy to search the optimal parameters $C, \nu, m, n$, and select a kernel function $K(\bullet, \bullet)$.

(3) Construct and solve the optimization problem

$$\max\left\{g_{D_{N\text{-SVR}}} = -\frac{1}{2}\sum_{i\in\text{RSV}}\sum_{j\in\text{RSV}}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) + \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)y_i + \frac{C}{l}\sum_{i=1}^{l}(T(\xi_i(\alpha_i)) + T(\xi_i^*(\alpha_i^*)))\right\}$$

Subject to

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{29}$$

$$0 \le \alpha_i^{(*)} \le \frac{C}{l}, \quad i = 1, \dots, l$$

$$\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) \le C\cdot\nu, \quad i = 1, \dots, l$$

where $T(\xi_i^{(*)}(\alpha_i^{(*)})) = c(\xi_i^{(*)}(\alpha_i^{(*)})) - \xi_i^{(*)}(\alpha_i^{(*)})\cdot\frac{\partial(c(\xi_i^{(*)}(\alpha_i^{(*)})))}{\partial(\xi_i^{(*)}(\alpha_i^{(*)}))}$, $0 < \nu \le 1$, $C > 0$ is constant.

We obtain the optimal solution $\alpha = (\alpha_1, \dots, \alpha_l, \alpha_1^*, \dots, \alpha_l^*)$.

(4) Construct the function as

$$f(x) = \omega^T\cdot\Phi(x) + b = \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot K(x_i, x) + b,$$

where $b = \frac{1}{2l}[\sum_{j=1}^{l}(y_j - \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot K(x_i, x_j)) + \sum_{k=1}^{l}(y_k - \sum_{i\in\text{RSV}}(\alpha_i^* - \alpha_i)\cdot K(x_i, x_k))]$, RSVs are the samples corresponding to $\alpha_i^* - \alpha_i \ne 0$ (called support vectors).

The ALM method is a class of algorithms for solving equality or inequality constrained optimization problems. For nonlinear programming problems with equality constraints, Powell and Hestenes (Rockafellar, 1973) designed a dual method of solution, where squares of the constraint functions are added as penalties to the Lagrangian, and a certain simple rule is used for updating the Lagrange multipliers after each cycle (called the PH algorithm). Rockafellar (1973, 1974) extended it for solving inequality constrained tasks (called the PHR algorithm). The algorithm starts from the Lagrange function of the original problem, coupled with an appropriate penalty function. The original problem is transformed into solving a series of unconstrained optimization sub-problems.

The ALM method was employed to solve Problem (29) by applying Newton's method to a sequence of equality and inequality constrained problems. Any equality and inequality constrained minimization problem can be reduced to an equivalent unconstrained problem by eliminating the equality and inequality constrains. The Gradient descent method or Newton's method can be used to solve the problem (Boyd & Vandenberghe, 2004; Chen & Jian, 1994; Fiacco & McCormick, 1990; Ma, 2010). If there are large-scale training samples, some fast optimization techniques can also be combined with the proposed objective functions, such as stochastic gradient decent (SDG) (Bottou, 2010).

In this work, we solve the $\nu$-support vector regression machine with the ALM algorithm. ALM is used to solve Problem (29) by applying Newton's method to a sequence of equality and inequality constrained problems.

Problem (29) is equivalent to

$$\begin{aligned} &\min f(x) \\ &h_i(x) = 0, \quad i = 1, \ldots, L_1 \\ &g_i(x) \geq 0, \quad i = 1, \ldots, L_2 \end{aligned} \tag{30}$$

with

$$f(x) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) - \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)y_i - \frac{C}{l}\sum_{i=1}^{l}(T(\xi_i(\alpha_i)) + T(\xi_i^*(\alpha_i^*))), \tag{31}$$

where $x = (\alpha_1, \alpha_2, \ldots, \alpha_l, \alpha_1^*, \alpha_2^*, \ldots, \alpha_l^*)$, and $l$ is the number of training samples.

Construct the augmented Lagrange function of Problem (30)

$$\Psi(x, \mu, \lambda, \sigma) = f(x) - \sum_{i=1}^{L_1}\mu_i h_i(x) + \frac{\sigma}{2}\sum_{i=1}^{L_1}(h_i(x))^2 + \frac{1}{2\sigma}\sum_{i=1}^{L_2}([\min\{0, \sigma g_i(x) - \lambda_i\}]^2 - \lambda_i^2). \tag{32}$$

Formulation iteration of the multiplier method is

$$\begin{aligned} &(\mu_{k+1})_i = (\mu_k)_i - \sigma h_i(x_k), \quad i = 1, \ldots, L_1 \\ &(\lambda_{k+1})_i = \max\{0, (\lambda_k)_i - \sigma g_i(x_k)\}, \quad i = 1, \ldots, L_2. \end{aligned} \tag{33}$$

Let

$$\beta_k = \sqrt{\sum_{i=1}^{L_1}(h_i(x_k))^2 + \sum_{i=1}^{L_2}[\min\{g_i(x_k), (\lambda_k)_i/\sigma\}]^2} \tag{34}$$

the stopping rule is $\beta_k \leq \epsilon$, where $\epsilon \geq 0$ is a given threshold.

Finally, the algorithm of ALM is described as follows.

Step 1 Selection starting value. Take $x_0 \in R^n, \mu_1 \in R^{L_1}, \lambda_1 \in R^{L_2}, \sigma_1 > 0, 0 \leq \epsilon \ll 1, \delta \in (0, 1), \eta > 1, k := 1$.

Step 2 Solve the Augment Lagrange function. Take starting point $x_{k-1}$, solve the minimum point $x_k$ of unrestrained subproblem $\Psi(x, \mu_k, \lambda_k, \sigma_k)$ in (32).

Step 3 Examine termination conditions. If $\beta_k \leq \epsilon$ in (34), end iteration (32), and get the approximate minimum point $x_k$ of optimization problem (29); otherwise, turn Step 4.

Step 4 Update penalty parameter. If $\beta_k \geq \delta\beta_{k-1}$, take $\sigma_{k+1} = \eta\sigma_k$; otherwise, $\sigma_{k+1} = \sigma_k$.

Step 5 Update the multiplier vector. Calculation $(\mu_{k+1})_i, (\lambda_{k+1})_i$ in (33).

Step 6 Take $k := k + 1$, turn Step 1.

In the ALM algorithm, parameters $\mu_1 = (0.1, 0.1, \ldots, 0.1)^T$, $\lambda_1 = (0.1, 0.1, \ldots, 0.1)^T$, $\sigma_1 = 0.4$, $\epsilon = 10^{-5}$, $\eta = 2$, $\delta = 0.8$.

$N$-SVR has been implemented in Matlab 7.1 programming language. The initial parameters of the proposed ALM method are $C \in [1, 201]$, $\nu \in (0, 1]$, $m, n \in (1, 21)$. We use the 10-fold cross validation strategy to search the optimal positive parameters $C, \nu, m, n$, the selection technique of parameters $C, \nu, m, n$ was studied in detail in Chalimourda et al. (2004) and Cherkassky and Ma (2004). Many actual applications suggest that polynomial and Gaussian kernel function tend to perform well under general smoothness assumptions, so that it should be considered especially if no additional information is used as the kernel function of $N$-SVR. In this work, the polynomial function and the Gaussian kernel function are used in $\nu$-SVR, GN-SVR and BN-SVR (Schölkopf & Smola, 2002).

$$K(x_i, x_j) = ((x_i, x_j) + 1)^d,$$

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{\sigma^2}},$$

where $d$ is a positive integer, and $\sigma$ is positive.

**Table 2**
Error statistic of three models on artificial data with Gaussian noise.

| Model | MAE | RMSE | MAPE (%) |
|---|---|---|---|
| $\nu$-SVR | 1.1647 | 1.4670 | 36.59 |
| GN-SVR | 0.9860 | 1.2517 | 27.88 |
| BN-SVR | 1.0036 | 1.2651 | 28.54 |

## 5. Experimental analysis

In this section, we present some experiments to test the proposed model on several regression tasks with artificial data and UCI data.

The following criteria, including mean absolute error (MAE), mean absolute percentage error (MAPE), the root mean square error (RMSE), and the standard error of prediction (SEP), are introduced to evaluate the performance of the $\nu$-SVR, GN-SVR and BN-SVR models:

$$\text{MAE} = \frac{1}{l}\sum_{i=1}^{l}|y_i - x_i|, \tag{35}$$

$$\text{MAPE} = \frac{1}{l}\sum_{i=1}^{l}\frac{|y_i - x_i|}{x_i}, \tag{36}$$

$$\text{RMSE} = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(y_i - x_i)^2}, \tag{37}$$

$$\text{SEP} = \frac{\text{RMSE}}{\bar{x}}, \tag{38}$$

where $l$ is the size of the samples, $x_i$ is the $i$th sample and $y_i$ is the forecasting result of the $i$th data, $\bar{x}$ is the average of all the samples (Bludszuweit et al., 2008; Fan et al., 2009; Guo, Zhao, Zhang, & Wang, 2011).

### 5.1. Artificial data sets

To illustrate $N$-SVR, artificial data sets with Gauss-noise and beta-noise are studied.

(1) Artificial data with Gauss-noise: as a training set, we use 300 examples $(x_i, y_i)$, the relationships between inputs $x_i$ and targets $y_i$ are as follows: $y_i = 2\sin(3x_i + 2) + 6\cos(2x_i + 1) + \text{norm}(0, 1, 1000, 1)$, here the $x_i$ is drawn uniformly from the interval $[0.01 \times \pi, 10 \times \pi]$. Let the number of training samples be 150 (from 1 to 150), and the number of tested samples be 150 (from 151 to 300). Fig. 3 gives the forecasting results of artificial data with Gaussian noise given by $\nu$-SVR, GN-SVR and BN-SVR, respectively.

The criteria of MAE, MAPE and RMSE are used to evaluate the forecasting performance of three models, shown in Table 2.

It is easy to see that GN-SVR is better than $\nu$-SVR and BN-SVR on the artificial data with Gaussian noise.

(2) Artificial data with beta-noise: as a training set, we use 300 examples $(x_i, y_i)$, the relationships between inputs $x_i$ and targets $y_i$ are $y_i = 2\sin(3x_i+2)+6\cos(2x_i+1)+6\,\text{beta}(1.41, 1.71, 1000, 1)$, where $x_i$ is drawn uniformly from the interval $[0.01 \times \pi, 10 \times \pi]$. Let the number of training samples be 150 (from 1 to 150), and the number of tested samples be 150 (from 151 to 300). Fig. 4

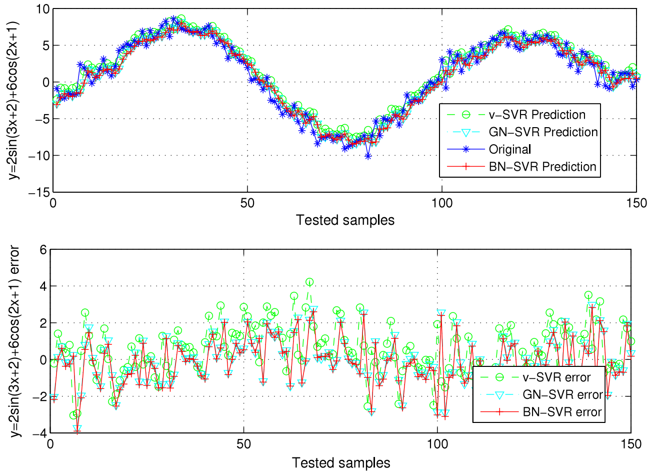

**Fig. 3.** The forecast result of three models on artificial data with Gaussian noise.

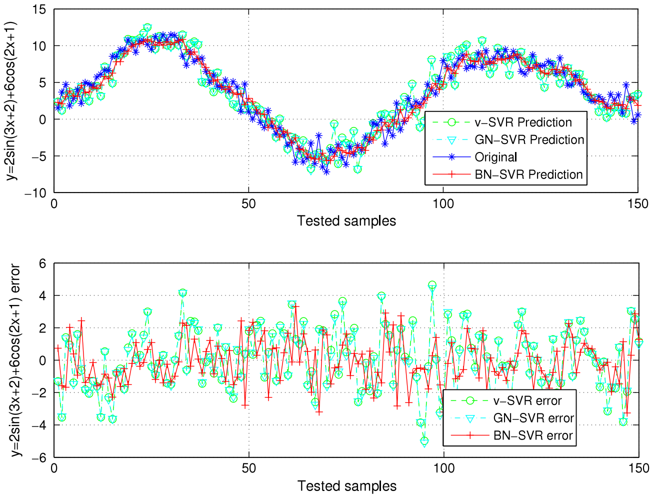

**Fig. 4.** The forecast result of three models on artificial data with beta noise.

**Table 3**
Error statistic of three models on artificial data with beta noise.

| Model | MAE | RMSE | MAPE (%) |
|---|---|---|---|
| $\nu$-SVR | 1.5857 | 1.9154 | 17.857 |
| GN-SVR | 1.5580 | 1.8909 | 18.378 |
| BN-SVR | 1.1467 | 1.3965 | 11.325 |

illuminates the forecast results of artificial data with beta noise given by $\nu$-SVR, GN-SVR and BN-SVR.

The criteria of MAE, MAPE and RMSE are used to evaluate the forecast results of three models shown in Table 3.

Experimental results on artificial data with beta noise demonstrate that BN-SVR is better than the classical models.

### 5.2. UCI data sets

UCI data include abalone and stock, etc. We first add beta-noise to two data sets above, let the number of training samples be 240 (from 1 to 240), and the number of tested samples be 480 (from 241 to 720). Figs. 5 and 6 illuminate the forecast results of abalone and stock data sets with beta noise given by $\nu$-SVR, GN-SVR and BN-SVR, respectively.

The criteria of MAE, MAPE and RMSE are used to evaluate the forecast results of three models shown in Tables 4 and 5.

The experiment results on UCI data with beta noise show that BN-SVR is better than $\nu$-SVR and GN-SVR.

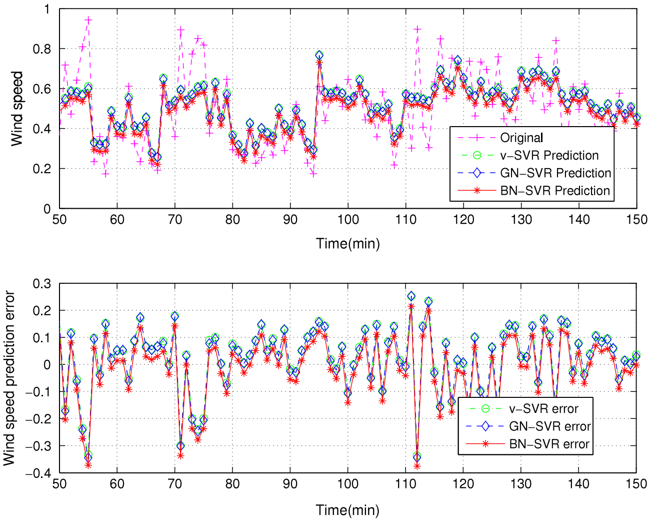

**Fig. 5.** The forecast result of three models on abalone data set.

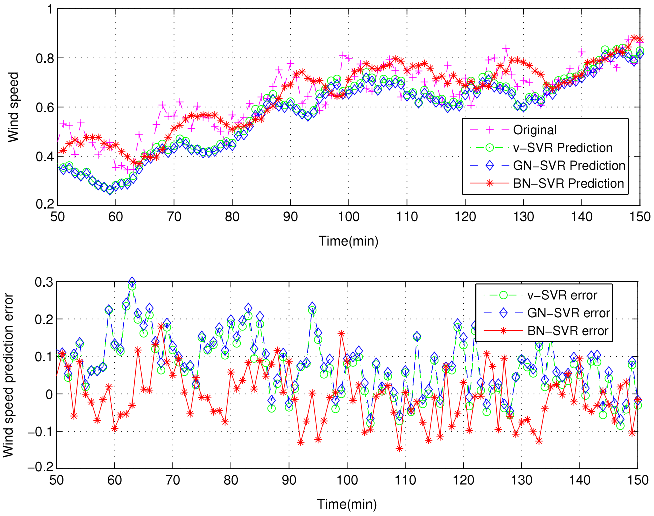

**Fig. 6.** The forecast result of three models on stock data set.

**Table 4**
Error statistic of three models on abalone data set.

| Model | MAE | RMSE | MAPE (%) |
|---|---|---|---|
| $\nu$-SVR | 0.0932 | 0.1158 | 21.92 |
| GN-SVR | 0.0932 | 0.1164 | 21.82 |
| BN-SVR | 0.0892 | 0.1128 | 18.94 |

**Table 5**
Error statistic of three models on stock data set.

| Model | MAE | RMSE | MAPE (%) |
|---|---|---|---|
| $\nu$-SVR | 0.0943 | 0.1142 | 14.27 |
| GN-SVR | 0.0938 | 0.1135 | 14.39 |
| BN-SVR | 0.0801 | 0.1002 | 11.34 |

## 6. Short-term wind speed prediction with the proposed algorithm

The forecasting model of $N$-SVR is applied in a real-world sample set of wind speed in Heilongjiang Province. The data record more than a year wind speeds. The average wind speeds in 10 min are stored. As a whole, 62 466 samples with 4 attributes, mean, variance, minimum, maximum, are given.

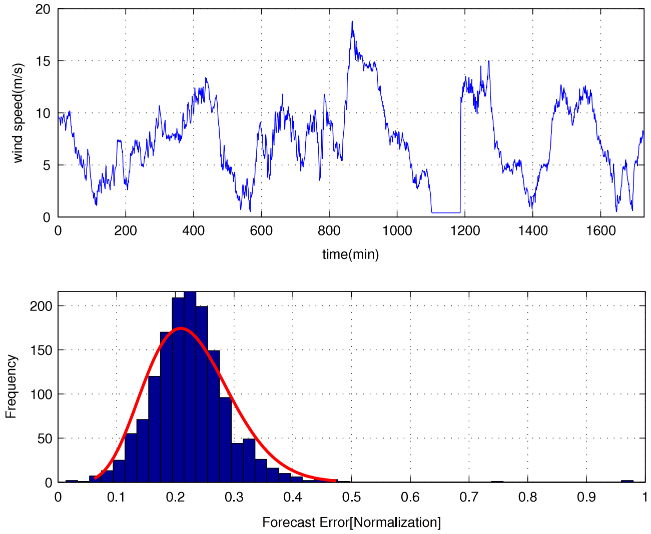

**Fig. 7.** The beta distribution of the wind speed forecasting error with the persistence method.

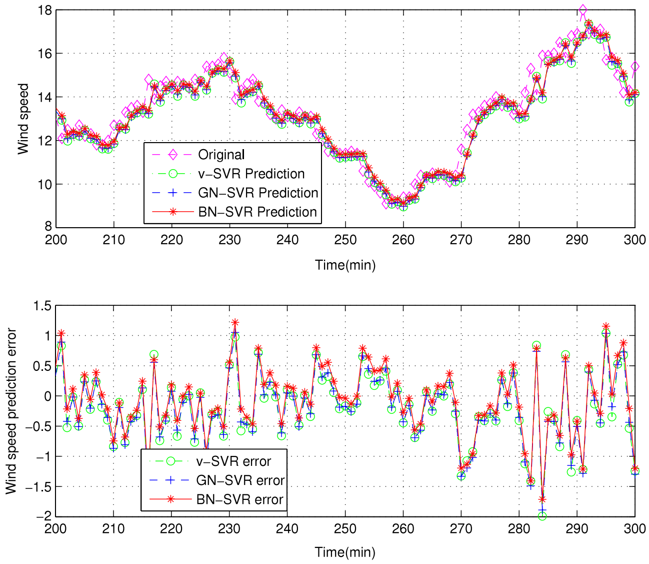

**Fig. 8.** The forecasting result of three models on wind speed after 10 min.

We analyze one-month time series of wind speeds. Let us investigate the error distribution by using the persistence method (Bludszuweit et al., 2008). The result shows that the error $\epsilon$ of wind speed with the persistence forecast does not obey the Gaussian distribution, while obeys the beta distribution, and the PDF of $\epsilon$ is $f(\epsilon) = \epsilon^{7.22} \cdot (1-\epsilon)^{27.32}$, $\epsilon \in (0, 1)$, as shown in Fig. 7.

Now, we use 288 samples (from 1 to 288, the time length is 48 h) as a training set, and 576 samples as a test set (from 289 to 864, the time length is 96 h). The input vector is $\overrightarrow{x_i} = (x_{i-11}, x_{i-10}, \ldots, x_{i-1}, x_i)$, the output value is $x_{i+\text{step}}$, where step $= 1, 6$. Namely we use the above model to predict the wind speed after 10 and 60 min of each point $x_i$, respectively. Figs. 8–11 give the forecasting results given by $\nu$-SVR, GN-SVR and BN-SVR.

Figs. 8 and 9 illuminate the forecasting results after 10 min of every point $x_i$ given by $\nu$-SVR, GN-SVR and BN-SVR.

The indicators of MAE, MAPE, RMSE and SEP are shown in Table 6.

Figs. 10 and 11 present the forecasting results after 60 min of every point $x_i$ given by $\nu$-SVR, GN-SVR and BN-SVR.

The indicators of MAE, MAPE, RMSE and SEP are shown in Table 7.

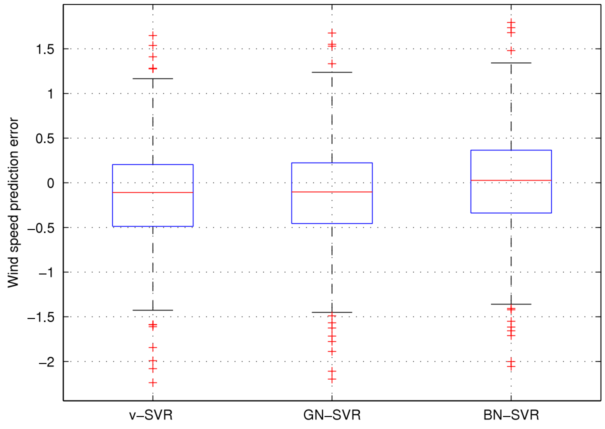

**Fig. 9.** The boxplot of three models on wind speed prediction after 10 min.

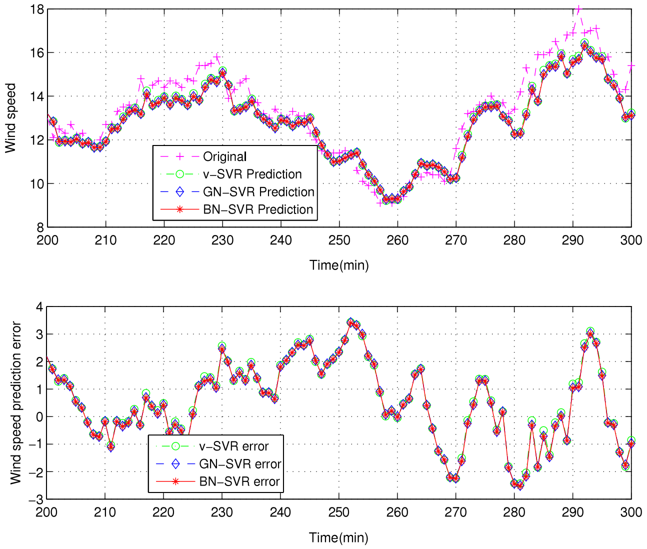

**Fig. 10.** The forecasting result of three models on wind speed after 60 min.

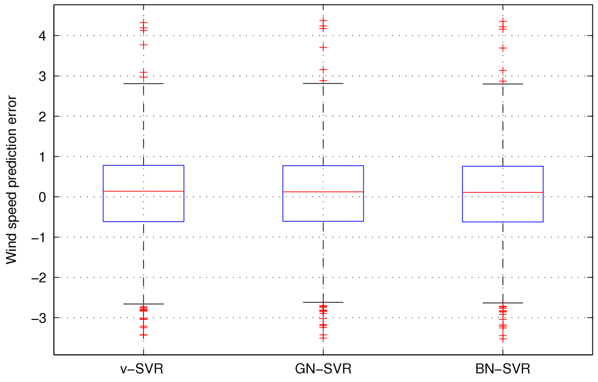

**Fig. 11.** The boxplot of three models on wind speed prediction after 60 min.

**Table 6**
Error statistic of three models on wind speed prediction after 10 min.

| Model | MAE (m/s) | RMSE (m/s) | MAPE (%) | SEP (%) |
|---|---|---|---|---|
| $\nu$-SVR | 0.4485 | 0.5848 | 3.86 | 4.96 |
| GN-SVR | 0.4465 | 0.5823 | 3.84 | 4.94 |
| BN-SVR | 0.4391 | 0.5692 | 3.79 | 4.83 |

From Box–whisker Plots 9 and 11, Figs. 8 and 10, Tables 6 and 7, it is to derive that the errors computed with BN-SVR are a little

**Table 7**
Error statistic of three models on wind speed prediction after 60 min.

| Model | MAE (m/s) | RMSE (m/s) | MAPE (%) | SEP (%) |
|---|---|---|---|---|
| $\nu$-SVR | 1.3388 | 1.5988 | 12.53 | 13.59 |
| GN-SVR | 1.3304 | 1.5912 | 12.50 | 13.52 |
| BN-SVR | 1.3206 | 1.5811 | 12.40 | 13.44 |

smaller than those with $\nu$-SVR and GN-SVR in the most cases. As the prediction horizon increases to 60 min, both the errors derived with different models rise, the relative difference decreases. So it is not so significant in these cases. However, we can know Beta model is still better than the classical models in terms of all the criteria of MAE, MAPE, RMSE and SEP, seen from Tables 6 and 7.

## 7. Conclusions and future work

Most existing regression techniques take the assumption that the error model is Gaussian. However, it was found that the noise in some real-world applications, just like wind speed forecast, does not satisfy Gaussian distribution, but a beta distribution. In this case, these regression techniques are not optimal. In this work, we describe the main results of our work: (1) we derive the optimal loss functions for different error models; (2) we develop the uniform $\nu$-support vector regression for the general noise model ($N$-SVR); (3) the solution of the primal problem of $N$-SVR about $\omega$ exists and is unique; (4) introducing the Lagrange functional and according to KKT conditions, we obtain the dual problem of $N$-SVR; (5) the Augmented Lagrange Multiplier method is applied to solve $N$-SVR, which guarantees the stability and validity; (6) experimental results on artificial data sets, UCI data and the real-world data of wind speed confirm the performance of the proposed technique.

Similarly, we can derive the model of $\nu$-support vector classification for the general noise model, which will be successfully used to solve the classification problem for the noise model.

In practical regression problems, data uncertainty is inevitable. The observed data are usually described in linguistic levels or ambiguous metrics. Like the weather forecast, the forecast results of dry and wet, or sunny and cloudy, and so on. We should consider developing fuzzy $\nu$-support vector regression algorithms with different noise models.

## Acknowledgments

This work was supported by National Program on Key Basic Research Project (973 Program) under Grant 2012CB215201, and National Natural Science Foundation of China (NSFC) under Grants 61222210, 61170107, 61105054, and 61300121.

## References

Bayro-Corrochano, E. J., & Arana-Daniel, N. (2010). Clifford support vector machines for classification, regression, and recurrence. *IEEE Transactions on Neural Networks*, *21*(11), 1731–1746.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. (pp. 71–73). New York: Springer.

Bludszuweit, H., Antonio, J., & Llombart, A. (2008). Statistical analysis of wind power forecast error. *IEEE Transactions on Power Systems*, *23*(3), 983–991.

Bofinger, S., Luig, A., & Beyer, H. G. (2002). Qualification of wind power forecasts. In *Proc. global wind power conf., Paris, France*.

Bottou, Léon (2010). Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier, & Gilbert Saporta (Eds.), *Proceedings of the 19th international conference on computational statistics* (pp. 177–187). Paris, France: Springer.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. (pp. 521–620). Cambridge University Press.

Canavos, G. C. (1984). *Applied probability and statistical methods*. Toronto: Little, Brown and Company.

Chalimourda, A., Schölkopf, B., & Smola, A. J. (2004). Experimentally optimal $\nu$ in support vector regression for different noise models and parameter settings. *Neural Networks*, *17*(1), 127–141.

Chen, D. S., & Jian, R. C. (1994). A robust backpropagation learning algorithm for function approximation. *IEEE Transactions on Neural Networks*, *5*(3), 467–479.

Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, *17*, 113–126.

Chih-Chung, C., & Chih-Jen, L. (2002). Training v-support vector regression: theory and algorithms. *Neural Computation*, *14*, 1959–1977.

Chu, W., Keerthi, S. S., & Ong, C. J. (2004). Bayesian support vector regression using a unified loss function. *IEEE Transactions on Neural Networks*, *22*(1), 29–44.

Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning*, *20*(3), 273–297.

Cristianini, N., & Shawe, T. (2000). *An introduction to support vector machines*. Cambridge University Press.

Duan, L., Xu, D., & Tsang, I. W. H. (2012). Domain adaptation from multiple sources: a domain-dependent regularization approach. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(3), 504–518.

Eltoft, T., Kim, T., & Lee, T. W. (2006). On the multivariate Laplace distribution. *IEEE Signal Processing Letters*, *13*(5), 300–303.

Esposito, F., Malerba, D., & Semeraro, G. (1997). A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(5), 476–491.

Fabbri, A., Román, T. G. S., Abbad, J. R., & Quezada, V. H. M. (2005). Assessment ofthe cost associated with wind generation prediction errors in a liberalized electricity market. *IEEE Transactions on Power Systems*, *20*(3), 1440–1446.

Fan, S., Liao, J. R., et al. (2009). Forecasting the wind generation using a two-stage network based on meteorological information. *IEEE Transactions on Energy Conversion*, *24*(2), 474–482.

Fiacco, A. V., & McCormick, G. P. (1990). *Nonlinear programming. Sequential unconstrained minimization techniques*. Society or Industrial and Applied Mathematics, First published in 1968 by Research Analysis Corporation.

Girosi, F. (1991). Models of noise and robust estimates. A.I. memo 1287, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Guo, Z. H., Zhao, J., Zhang, W. Y., & Wang, J. Z. (2011). A corrected hybrid approach for wind speed prediction in Hexi Corridor of China. *Energy*, *36*, 1668–1679.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer.

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

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, *35*(1), 73–101.

Huber, P. J. (1981). *Robust statistics*. New York: Wiley.

Klaus-Robert, M., & Sebastian, M. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, *12*(2), 181–202.

Kwok, J. T., & Tsang, I. W. (2003). Linear dependency between and the input noise in $\epsilon$-support vector regression. *IEEE Transactions on Neural Networks*, *14*(3), 544–553.

Lopez, J., & Dorronsoro, J. R. (2012). Simple proof of convergence of the SMO algorithm for different SVM variants. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(7), 1142–1147.

Ma, C. F. (2010). *Optimization method and the matlab programing design*. (pp. 121–131). China: Science Press.

Olvi, L. M., & David, R. M. (2000). Robust linear and support vector regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(9), 950–955.

Park, H. J., & Lee, T. W. (2005). Modeling nonlinear dependencies in natural images using mixture of Laplacian distribution. In *Advances in neural information processing systems, Vol. 17*. Cambridge, MA: MIT Press.

Pontil, M., Mukherjee, S., & Girosi, F. (1998). On the noise model of support vector machines regression. A.I. memo 1651, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Randazzo, A., Abou-Khousa, M. A., Pastorino, M., & Zoughi, R. (2007). Direction of arrival estimation based on support vector regression: experimental validation and comparison with music. *IEEE Antennas and Wireless Propagation Letters*, *6*, 379–382.

Rockafellar, R. T. (1973). The multiplier method of Hestenes and Powell applied to convex programming. *Journal of Optimization Theory and Applications*, *12*(6), 555–562.

Rockafellar, R. T. (1974). Augmented Lagrange multiplier functions and duality in nonconvex programming. *SIAM Journal on Control*, *12*(2), 268–285.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, MA: MIT Press.

Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, *12*, 1207–1245.

Shevade, S. K., Keerthi, S. S., Bhattacharyya, C., & Murthy, K. R. K. (2000). Improvements to the SMO algorithm for SVM regression. *IEEE Transactions on Neural Networks*, *11*(5), 1188–1193.

Smola, A., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222.

Spech, D. F. (1990). Probabilistic neural networks. *Neural Networks*, *3*, 109–118.

Suykens, J. A. K., Lukas, L., & Vandewalle, J. (2000). Sparse approximation using least square vector machines. In *IEEE International symposium on circuits and systems, Genvea, Switzerland* (pp. 757–760).

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.

Vapnik, V. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, *10*(5), 988–999.

Vapnik, V., Golowich, S. E., & Smola, A. (1996). Support vector method for function approximation, regression estimation, and signal processing. In *Advances in neural information processings systems, Vol. 9* (pp. 281–287). Cambridge, MA: The MIT Press.

Wu, Q. (2010). A hybrid-forecasting model based on Gaussian support vector machine and chaotic particle swarm optimization. *Expert Systems with Applications*, *37*, 2388–2394.

Wu, Q., & Law, R. (2011). The forecasting model based on modified SVRM and PSO penalizing Gaussian noise. *Expert Systems with Applications*, *38*(3), 1887–1894.

Yang, J. B., & Ong, C. J. (2011). Feature selection using probabilistic prediction of support vector regression. *IEEE Transactions on Neural Networks*, *22*(6), 954–962.

Zhang, Y., Wan, Q., Zhao, H. P., & Yang, W. L. (2007). Support vector regression for basis selection in Laplacian noise environment. *IEEE Signal Processing*, *14*(11), 871–874.