Journal of the Korean Data & Information Science Society 2018, **29**(6), 1707–1719

Support vector regression with the weighted absolute deviation error loss function^{\dagger}

Kang-Mo Jung¹

¹Department of Statistics and Computer Science, Kunsan National University Received 29 October 2018, revised 16 November 2018, accepted 18 November 2018

Abstract

In this paper we propose robust support vector regression algorithms to deal with noisy data sets. We adopt the absolute deviation error function for a loss function of regression model, and the proposed algorithms preserves the structure of the least squares support vector regression. The proposed algorithms are very fast and the procedures are much simpler than other support vector machine algorithms. They are robust to regression outliers, because the loss functions are less increasing than the squares error function for large errors and it uses a weight function for each observation. By comparing the proposed algorithms with other methods for the simulated datasets and benchmark datasets, the proposed methods are more robust than the least squares support vector regression when outliers exist.

 $\mathit{Keywords}:$ Absolute deviation, least squares, outliers, robust methods, support vector regression, weights.

1. Introduction

Support vector machine (SVM), introduced by Vapnik (1995, 1998) has been very important machine learning methodology for classification and regression estimation. It has been successfully applied into many applications such as text classification, feature extraction and function estimation, see also Amayri and Bouguila (2010), Cao and Tay (2003), Isa *et al.* (2008), Mitra *et al.* (2007), Shen *et al.* (2008), Lee *et al.* (2013). Also, Hwang (2011) proposed a weighted least squares support vector machine for asymmetric least squares regression and Seok (2014) applied support vector machine to labeled and unlabeled data.

However, SVM has much computational burden, because it solves the optimization problem with inequality constraints. There are two approaches to reduce the computational effort. One is to train the data set by decomposition techniques such as sequential minimal optimization (SMO) (Keerthi *et al.*, 2001) or chucking methods which solve the quadratic

[†] This research was supported by Basic Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Engineering(NRF-2015R1D1A1A01060528).

¹ Professor, Department of Statistics and Computer Science, Kunsan National University, Kunsan 54150, Korea. E-mail: kmjung@kunsan.ac.kr

programming problem that arises for training SVM. The other is to use the equality constraints instead of the inequality ones, which is called the least squares support vector regression (LS-SVR) suggested by Suykens and Vandelwalle (1999). LS-SVR solves a series of linear equations and so the computational amount reduces. It can be a particular version of penalized estimation which maintains the balance between empirical risks and generalization. LS-SVR uses the square loss function of training samples and the quadratic penalty function for the exactness and the generalization of the model, respectively.

Unfortunately, there are two drawbacks in LS-SVR. The fist one is that its solution is not sparse because of the equality constraints, and the second one is that it is not robust to outliers because of the square loss function (Wen *et al.*, 2010). To overcome the first drawback Suykens *et al.* (2002) proposed a pruning method. To deal with the second problem some researchers developed weighted LS-SVR to reduce the influence of outliers. The main idea of this approach is how to set appropriately weight for each observation. Suykens *et al.* (2002) gave small weights to large residuals which is computed by a robust standardized statistic. Wen *et al.* (2010) proposed the weight scheme that samples far from others based on the least trimmed sum of absolute deviations have smaller weights.

Some authors studied robust loss functions instead of the weight setting approaches to resolve an outlier issue. For example, Yang *et al.* (2014) adopted a truncated least squares loss function. Wang *et al.* (2014) used non-convex least squares loss function which can be solved by utilizing the concave-convex procedure. These loss functions are not convex, the corresponding optimization problems are hard to be solved and consequently they requires much computation time.

Motivated by the aforementioned research, in this paper, we propose robust LS-SVR methods with the least absolute deviation (LAD) loss function and weights to reduce the influence of both regression outliers. We call this weighted least absolute deviation support vector regression (WLAD-SVR). Since the proposed loss function is not differentiable, the closed solution like LS-SVR can not be derived. However, we adopt an approximation approach to obtain the influence of observations and so we can obtain the influence information by solving linear equations like LS-SVR. Therefore, we can investigate the influence of observations on the estimator without burden of computation time, unlike SVR methods with non-convex loss functions. Wang *et al.* (2014) used a Newton method for constructing a smooth function of the LAD loss function.

This paper is organized as follows. In Section 2 we review the LS-SVR estimator and we propose LAD-SVR and WLAD-SVR algorithms. Section 3 gives an algorithm to implement the proposed methods. Wang *et al.* (2014) and Chen *et al.* (2017) proposed an approximation method of the LAD loss function. We also use an approximation algorithm of the LAD loss function, which can be written by a linear equation system to minimize the non-differentiable function. Also, we consider the weight for each case, which reduces the effect of outlying observations. Section 4 provides the result for the artificial and benchmark datasets. It shows that the performance of the proposed method is superior to the traditional SVM in the view of the exactness and robustness. Section 5 gives concluding remarks.

2. Weighted least absolute deviation support vector regression

Consider a regression problem with a training data $(\mathbf{x}_i, y_i), i = 1, \dots, n$, where $\mathbf{x}_i \in \mathbb{R}^p$ is the input data and $y_i \in \mathbb{R}$ is the corresponding output (Suykens *et al.*, 2002). The LS-SVR

algorithm finds the solution of the following objective function

$$min_{\mathbf{w},b}\frac{1}{2}\mathbf{w}^{T}\mathbf{w} + \frac{\gamma}{2}\sum_{i=1}^{n}\epsilon_{i}^{2}$$

$$(2.1)$$

s.t.
$$y_i = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) + b + \epsilon_i, i = 1, \cdots, n,$$
 (2.2)

where $\mathbf{w} \in \mathbb{R}^h$ is the model complexity, b is the bias, ϵ is the error variable, $\phi(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \dots, \phi_h(\mathbf{x}_i))$ is the function which maps the input space into a feature space, and γ is the regularization parameter which controls the balance between the fitting error and the model complexity.

Lagrange multipliers and Karush-Kuhn-Tucker (KKT) conditions yields the solution of LS-SVR as solving the linear system

$$\begin{pmatrix} \mathbf{K} + \frac{1}{\gamma} \mathbf{I}_n & \mathbf{1}_n \\ \mathbf{1}_n^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix},$$
(2.3)

where $\mathbf{1}_n$ is the $n \times 1$ vector of ones, $\boldsymbol{\alpha}$ is the Lagrange parameter of length n, $\mathbf{K} = (K_{ij})_{n \times n}$ is the kernel matrix whose element is $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$. In this paper we use the radial bases function (RBF) $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/\sigma^2)$, where σ is the bandwidth.

From the solution α , b of (2.3), we get the nonlinear regression estimate as

$$f(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b.$$

Instead of \mathbf{w} , b in (2.1), we estimate the function using α , b and kernel matrix in (2.3). It implies that the regression function can be represented by a linear combination of kernel functions at the input data. This result is called the representer theorem (Argyriou *et al.*, 2009). By the kernel reproducing property (Chapelle, 2007) the problem (2.1) becomes

$$min_{\boldsymbol{\alpha},b}\frac{1}{2}\boldsymbol{\alpha}^{T}\mathbf{K}\boldsymbol{\alpha} + \frac{\gamma}{2}\sum_{i=1}^{n}(y_{i}-\mathbf{K}_{i}\boldsymbol{\alpha}-b)^{2},$$
(2.4)

where \mathbf{K}_i is the *i*th row vector of kernel matrix \mathbf{K} .

The LS-SVR converts the inequality constraints of SVR into equality constraints, and so the computation is very fast. However, the LS-SVR has the drawback of the sensitivity to outliers or noises with the square loss function. In order to overcome the non-robustness of LS-SVR Suykens *et al.* (2002) proposed the weighted LS-SVR (WLS-SVR) by putting small weights on outliers to reduce their influence to the model. They considered the objective function instead of (2.4)

$$min_{\boldsymbol{\alpha},b}\frac{1}{2}\boldsymbol{\alpha}^{T}\mathbf{K}\boldsymbol{\alpha} + \frac{\gamma}{2}\sum_{i=1}^{n}\nu_{i}(y_{i} - \mathbf{K}_{i}\boldsymbol{\alpha} - b)^{2}, \qquad (2.5)$$

where ν_i is the weighting factor given by Rousseeuw and Leroy (1987)

$$\nu_i = \begin{cases} 1 & \text{if } |\hat{\epsilon}_i/\hat{s}| \leq \eta_1 \\ \frac{\eta_2 - |\eta_i/\hat{s}|}{\eta_2 - \eta_1} & \text{if } \eta_1 \leq |\hat{\epsilon}_i/\hat{s}| \leq \eta_2, \\ 10^{-4} & \text{otherwise,} \end{cases}$$

1709

Kang-Mo Jung

where $\eta_1 = 2.5, \eta_2 = 3, \hat{\epsilon}_i = \hat{\alpha}_i / \gamma, \hat{s}$ is a robust estimate of the standard deviation of the LS-SVR error variable ϵ_i , for example, $\hat{s} = IQR/(2 \times 0.6745)$ or $MAD(\hat{\epsilon}_i)$. The interquartile range IQR is the difference between the 75th percentile and the 25th percentile. The MAD stands for the median absolute deviation, defined as $MAD = median(|\hat{\epsilon}_i - median(\hat{\epsilon}_i)|)$.

Similar to (2.4), we can obtain the optimal solution for (2.5) using the Lagrangian multipliers. By setting the derivatives of (2.5) with respect to α and b to be zero, similar to (2.3) we obtain the linear equations

$$\begin{pmatrix} \mathbf{K} + \boldsymbol{\nu}/\gamma & \mathbf{1}_n \\ \mathbf{1}_n^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix},$$
(2.6)

where $\boldsymbol{\nu} = diag(1/\nu_1, \cdots, 1/\nu_n).$

De Brabanter *et al.* (2009) further developed an iteratively reweighted LS-SVR by improving the WLS-SVR. To further improve the robustness of WLS-SVR, Chen *et al.* (2015) and Liu *et al.* (2016) used trimmed squares function and penalized trimmed squares function instead of squares loss function, respectively.

Insipred by the robustness of LAD to outliers, we propose the weighted LAD loss function instead of the LS loss function (2.5), and thus obtain the objective function

$$min_{\boldsymbol{\alpha},b}\frac{1}{2}\boldsymbol{\alpha}^{T}\mathbf{K}\boldsymbol{\alpha} + \gamma \sum_{i=1}^{n} \nu_{i}|y_{i} - \mathbf{K}_{i}\boldsymbol{\alpha} - b|.$$
(2.7)

The LAD loss function is convex, but non-differentiable. Then it is not easy to directly get the solution of (2.7). We call the solution of (2.7) the WLAD-SVR estimator for a nonlinear regression model. In case $\boldsymbol{\nu} = \mathbf{1}_n$, that is when the weights are not considered, we call the solution of (2.7) LAD-SVR. Without the consideration of weights Wang *et al.* (2014) used the Newton algorithm of the Huber loss function which is similar to the LAD function to solve the objective function (2.7), and some authors tried to use approximations (Chen *et al.*, 2014).

3. Computing Algorithm

The LS-SVR in (2.4) can be solved by linear equation (2.1), because of the quadratic form of the second term in (2.4). We need to use the quadratic term instead of the absolute term in (2.7). Note that

$$|u| \approx \frac{u^2}{2|u_0|} + \frac{|u_0|}{2} \tag{3.1}$$

for nonzero u_0 near u (Jung, 2013). Then (2.7) at the (t + 1) step can be approximated by the following model up to constant terms

$$\mathcal{L}_{t+1}(\boldsymbol{\alpha}, b) = \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} + \frac{\gamma}{2} \sum_{i=1}^n \nu_i \frac{(\mathbf{K}_i \boldsymbol{\alpha} + b - y_i)^2}{|\mathbf{K}_i \boldsymbol{\alpha}_t + b_t - y_i|}.$$
(3.2)

Letting the derivative of \mathcal{L}_{t+1} with respect to $\boldsymbol{\alpha}$ and b be zero, we obtain the linear equations

$$\begin{cases} (\mathbf{K} + \mathbf{V}_t / \gamma) \boldsymbol{\alpha} + b \mathbf{1}_n &= \mathbf{y} \\ \mathbf{1}_n^T \boldsymbol{\alpha} &= 0 \end{cases}$$

or

$$\begin{pmatrix} \mathbf{K} + \mathbf{V}_t / \gamma & \mathbf{1}_n \\ \mathbf{1}_n^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix},$$
(3.3)

1711

where $\mathbf{V}_t = diag(|\mathbf{K}_i \boldsymbol{\alpha}_t + b_t - y_i|/\nu_i).$ Let

$$\mathbf{J}_t = \begin{pmatrix} \mathbf{K} + \mathbf{V}_t / \gamma & \mathbf{1}_n \\ \mathbf{1}_n^T & 0 \end{pmatrix}^{-1}.$$

Based on matrix inversion formula (Graybill, 1983), \mathbf{J}_t can be rewritten by

$$\mathbf{J}_t = \frac{1}{h_t} \begin{pmatrix} \mathbf{A}_t^{-1} - \mathbf{A}_t^{-1} \mathbf{1}_n \mathbf{1}_n^T \mathbf{A}_t^{-1} & \mathbf{A}_t^{-1} \mathbf{1}_n \\ \mathbf{1}_n^T \mathbf{A}_t^{-1} & -1 \end{pmatrix},$$

where $\mathbf{A}_t = \mathbf{K} + \mathbf{V}_t / \gamma$ and $h_t = \mathbf{1}_n^T \mathbf{A}_t^{-1} \mathbf{1}_n$. Then we get the solution $\boldsymbol{\alpha}_{t+1}$ and b_{t+1} at the (t+1)-th step by the linear equation

$$\begin{pmatrix} \boldsymbol{\alpha}_{t+1} \\ b_{t+1} \end{pmatrix} = \mathbf{J}_t \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix} = \begin{pmatrix} \mathbf{A}_t^{-1} - \mathbf{A}_t^{-1} \mathbf{1}_n \mathbf{1}_n^T \mathbf{A}_t^{-1} \\ \mathbf{1}_n^T \mathbf{A}_t^{-1} \end{pmatrix} \mathbf{y}/h_t.$$
(3.4)

Similar to (2.1) the solution α_{t+1} , b_{t+1} at the (t+1)-th step in (3.4) can be reformulated as the primal form

$$min_{\mathbf{w},b}\frac{1}{2}\mathbf{w}^{T}\mathbf{w} + \gamma \sum_{i=1}^{n} \nu_{i}|\epsilon_{i}|.$$
(3.5)

The objective function (3.5) can be approximated up to constant terms by the identity (3.1) as following

$$\frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{\gamma}{2}\sum_{i=1}^n\nu_i\epsilon_i^2/|\epsilon_{i,t}|,$$

where $\epsilon_{i,t}$ denotes the error evaluated at the *t*-th step. Then the Lagrangian objective function becomes

$$\mathcal{L}^* = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\gamma}{2} \sum_{i=1}^n \nu_i \frac{\epsilon_i^2}{|\epsilon_{i,t}|} - \sum_{i=1}^n \alpha_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b + \epsilon_i - y_i)$$

for Lagrangian multipliers α_i . The KKT condition (Suykens *et al.*, 2002) can be given by

$$\begin{cases} \frac{\partial \mathcal{L}^*}{\partial \mathbf{w}} = 0 \to \mathbf{w} = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i) \\ \frac{\partial \mathcal{L}^*}{\partial b} = 0 \to \sum_{i=1}^n \alpha_i = 0 \\ \frac{\partial \mathcal{L}^*}{\partial \epsilon_i} = 0 \to \alpha_i = \gamma \frac{\nu_i \epsilon_i}{|\epsilon_{i,i}|}, \quad i = 1, \cdots, n \\ \frac{\partial \mathcal{L}^*}{\partial \alpha_i} = 0 \to \mathbf{w}^T \phi(\mathbf{x}_i) + b + \epsilon_i = y_i, \quad i = 1, \cdots, n. \end{cases}$$

Eliminating $\mathbf{w}, \boldsymbol{\epsilon}$ yields

$$\sum_{j=1}^{n} \alpha_j \phi(\mathbf{x}_j) \phi(\mathbf{x}_i) + b + \alpha_i |\epsilon_{i,t}| / \gamma \nu_i = y_i$$

and it can be rewritten by

$$\begin{pmatrix} \mathbf{K} + \mathbf{V}_t^* / \gamma & \mathbf{1}_n \\ \mathbf{1}_n^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix},$$
(3.6)

where $\mathbf{V}_{t}^{*} = diag(|\epsilon_{i,t}|/\nu_{i})$. Since $\epsilon_{i,t}$ can be estimated by $\mathbf{K}_{i}\boldsymbol{\alpha}_{t} + b_{t} - y_{i}$, (3.6) is the same as (3.4). That is, we obtain the WLAD-SVR solution such as

$$\begin{pmatrix} \boldsymbol{\alpha} \\ b \end{pmatrix} = \begin{pmatrix} \mathbf{A}_t^{*-1} - \mathbf{A}_t^{*-1} \mathbf{1}_n \mathbf{1}_n^T \mathbf{A}_t^{*-1} \\ \mathbf{1}_n^T \mathbf{A}_t^{*-1} \end{pmatrix} \mathbf{y} / h_t^*,$$
(3.7)

where $\mathbf{A}_{t}^{*} = \mathbf{K} + \mathbf{V}_{t}^{*}/\gamma$ and $h_{t}^{*} = \mathbf{1}_{n}^{T} \mathbf{A}_{t}^{*-1} \mathbf{1}_{n}$. The unweighted LAD-SVR estimate can be obtained by $\mathbf{V}_{t}^{*} = diag(|\epsilon_{i,t}|)$ in (3.7). It is well known that the least square loss function is highly sensitive to unusual cases in linear regression. LAD is a robust alternative to least squares (Giloni *et al.*, 2006). Furthermore, the weighted LAD estimate is more robust than unweighted LAD in least absolute shrinkage and selection operator (Jung, 2011).

To summarize the proposed algorithm, it is as following procedure.

Step 1. For training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, determine the optimal values (γ, σ) using the 5 folds cross-validation method based on a grid search, where γ and σ are selected from the range of 0.01 to 2 with the interval of 0.1 in this paper.

Step 2. Set t = 1 and $\nu_i = 1$.

Step 3. Solve the LAD-SVR α_t, b_t by (3.4)

Step 4. Update ν_i by (2.5) with $\hat{\epsilon}_i = \alpha_{i,t} |\epsilon_{i,t-1}| / \gamma$ and \mathbf{V}_t^* . Obtain α_{t+1}, b_{t+1} by (3.6).

Step 5 Compute $tol = ||\boldsymbol{\alpha}_{t+1} - \boldsymbol{\alpha}_t||_2^2 / ||\boldsymbol{\alpha}_t||_2^2$. If $tol > 1e^{-3}$, set t = t+1 and goto Step 3, otherwise exit.

4. Numerical Experiments

In numerical experiments the proposed algorithms LAD-SVR and WLAD-SVR are compared with LS-SVR and WLS-SVR to show the robustness of the proposed estimates for simulated and benchmark datasets. All the experiments were run on a personal computer with 3.6 GHz Intel Core i7-7700 CPU, 16.0 GB RAM, and Windows 10 64 bit operation system using R 3.4.0. The selection of parameters σ and γ is crucial to SVR methods based on kernel methods. There are so many approaches, such as grid search (Cristianini and Shawe-Taylor, 2000), particle swarm optimization (Li *et al.*, 2010; Baoi *et al.*, 2013), firefly algorithm (Xiong *et al.*, 2014), and memetic algorithm (*Hu et al.*, 2013) for optimization of the SVM parameters. Among these approaches, grid search is one of the most popular choice, which uses an exhaustive search on parameter space. In this paper we used a grid search to choose the optimal bandwidth σ and the regularization parameter γ by 5-folds cross validation method.

Root mean square error (RMSE), mean absolute error (MAE) and ratio between the sum squared error (SSE) and the sum squared total (SST) for testing samples are employed to evaluate the effectiveness of four estimates LS-SVR, WLS-SVR, LAD-SVR and WLAD-SVR (Wang *et al.*, 2014). These measures are defined by

RMSE =
$$\sqrt{\sum_{i=1}^{m} (y_i - \hat{y}_i)^2 / m},$$

MAE = $\sum_{i=1}^{m} |y_i - \hat{y}_i| / m,$
SSE/SST = $\sum_{i=1}^{m} (\hat{y}_i - y_i)^2 / \sum_{i=1}^{m} (y_i - \bar{y})^2,$

where *m* is the size of test samples, y_i and \hat{y}_i denotes the *i*-th response data and predicted data, respectively, $\bar{y} = \sum_{i=1}^{m} y_i/m$.

RMSE and MAE are commonly adopted fitting measures to assess the performance of a regression model. Usually, the smaller RMSE or MAE is, the better the model is (Chen *et al.*, 2017). The small SSE/SST value indicates the coincidence between true and predicted values (Zhao *et al.*, 2012).

4.1. Simulation

In this section we conducted SVR methods for two simulated functions with 6 error models to evaluate the robustness of the algorithms. For simulation study we generated the artificial datasets (x_i, y_i) as followings (Wang and Zhong, 2014)

$$\begin{aligned} y_i &= sin(3x_i)/(3x_i) + \epsilon_i, x_i \in [-4, 4], \\ y_i &= (x_i^2 - 1)^2 x_i^3 exp(-x_i) + \epsilon_i, x_i \in [-1, 1], \end{aligned}$$

where ϵ_i is the sample noise. We consider various noise distributions as followings

Case 1. Normal distribution $\epsilon_i \sim N(0, 0.2^2)$.

Case 2. Uniform distribution $\epsilon_i \sim U(-0.3, 0.3)$.

Case 3. t distribution ϵ_i with degrees of freedom $3 \sim t(3)$.

Case 4. Contaminated normal distribution $\epsilon_i \sim \rho N(0, 2^2) + (1 - \rho)N(0, 0.2^2)$, where ρ is the contaminated proportion. We set ρ as 0.1, 0.2 and 0.4.

Under the situation of Cases 1 and 2 it assumes that there is no outliers, while Cases 3 and 4 may contain outliers. We generated 200 training samples and 500 testing samples for simulation study. The initial search range may decide parameters $\sigma, \gamma \in [0.01, 2]$, refer to Suykens *et al.* (2002).

We conducted 100 simulation replications to evaluate RMSE, MAE, SSE/SST on the testing data. The simulation results are summarized in Tables 1-2. Results indicate that LS-SVR and WLS-SVR has a comparative performance to LAD-SVR and WLAD-SVR under Cases 1 and 2, because the data has no outliers. However, the former has large RMSE, MAE under Cases 3 and 4, that is, under the non-Gaussian noise or outliers. Especially

when the data is contaminated up to 40% outliers, the proposed estimate WLAD-SVR has lowest measures for two simulated functions. It implies that the proposed estimate LAD-SVR or WLAD-SVR is robust to outliers, the latter generally outperforms the former on the simulated datasets. Also WLAD-SVR has low SSE/SST values for most cases, and it indicates that the proposed algorithm is superior to LS-SVR and WLS-SVR, especially WLAD-SVR is robust to outliers or noise data.

For the function $\sin(3x)/(3x)$ the fitted lines for LS-SVR and WLAD-SVR are plotted in Figure 1, because the results of WLS-SVR and LAD-SVR are similar to LS-SVR and WLAD-SVR, respectively. The results show that under normal and uniform errors the approximation performances of LS-SVR is similar to that of WLAD-SVR. However, WLAD-SVR is more approximate to the true than LS-SVR for the data with outliers (Figure 1 c-f). Overall, the proposed method performs better than the others under the contaminated errors. For the 20% and 40% contaminated normal errors, the fitted line by LS-SVR has large deviation from the true lines. Figure 1 show that the generalization performance is further improved by applying LAD-SVR or WLAD-SVR.

4.2. Real dataset

In this section we conduct the experiments on eight benchmark regression datasets to test the robustness of the proposed algorithms, where the datasets AutoMPG, Concrete Compressive Strength, Machine CPU (MCPU), Pyrimidines (Pyrim), Servo, Yacht are downloaded from the UCI datasets (http://archive.ics.uci.edu/ml/datasets.html), Bodyfat, Pollution are downloaded from the StatLib database (http://lib.stat.cmu.edu/datasets/), and Diabetes is downloaded from the webpage (http://www.dcc.fc.up.pt/~ltorgo/Regression/DataStes.html). These datasets are widely used to evaluate regression algorithms. Each attribute of the sample including the response is normalized into the interval [0, 1]. Each datasets was randomly divided into 70% training and 30% testing samples.

Similar to the previous simulation study, we adopt LS-SVR, WLS-SVR, LAD-SVR, WLAD-SVR to the benchmark datasets and obtain the results in Table 3. It can be seen that the three performance measures (RMSE, MAE, and SSE/SST) of LAD-SVR or WLAD-SVR are the best ones in most datasets, which shows that the proposed algorithms are better than LS-SVR and WLS-SVR in the points of generalization and stability. These results reveal that the proposed methods have a higher accuracy than that of LS-SVR and WLS-SVR. The real data experimental results demonstrate that the approximation methods by LAD-SVR and WLAD-SVR and WLAD-SVR.

Noise	Method	RMSE	MAE	SSE/SST
$N(0, 0.2^2)$	LS-SVR	0.153	0.126	0.193
	WLS-SVR	0.155	0.128	0.198
	LAD-SVR	0.079	0.065	0.056
	WLAD-SVR	0.083	0.068	0.061
U(-0.3, 0.3)	LS-SVR	0.152	0.125	0.191
	WLS-SVR	0.152	0.125	0.192
	LAD-SVR	0.085	0.069	0.063
	WLAD-SVR	0.087	0.071	0.067
t(3)	LS-SVR	0.596	0.438	3.301
	WLS-SVR	0.464	0.360	1.885
	LAD-SVR	0.377	0.301	1.265
	WLAD-SVR	0.384	0.306	1.334
$0.9N(0, 0.2^2) + 0.1N(0, 2^2)$	LS-SVR	0.185	0.138	0.308
	WLS-SVR	0.097	0.077	0.090
	LAD-SVR	0.100	0.080	0.092
	WLAD-SVR	0.099	0.079	0.091
$0.8N(0, 0.2^2) + 0.2N(0, 2^2)$	LS-SVR	0.282	0.209	0.704
	WLS-SVR	0.130	0.098	0.157
	LAD-SVR	0.121	0.095	0.144
	WLAD-SVR	0.111	0.088	0.122
$0.6N(0, 0.2^2) + 0.4N(0, 2^2)$	LS-SVR	0.435	0.329	1.654
	WLS-SVR	0.287	0.214	0.730
	LAD-SVR	0.158	0.120	0.260
	WLAD-SVR	0.132	0.102	0.168

Table 4.1 Simulation results of $\sin(3x)/(3x)$ for various noise distributions

Table 4.2 Simulation results of $(x^2 - 1)^2 x^3 e^{-x}$ for various noise distributions

Noise	Method	RMSE	MAE	SSE/SST
$N(0, 0.2^2)$	LS-SVR	0.042	0.033	0.395
	WLS-SVR	0.042	0.033	0.405
	LAD-SVR	0.062	0.050	0.943
	WLAD-SVR	0.065	0.052	1.049
U(-0.3, 0.3)	LS-SVR	0.038	0.030	0.339
	WLS-SVR	0.038	0.030	0.339
	LAD-SVR	0.078	0.064	1.455
	WLAD-SVR	0.081	0.066	1.573
t(3)	LS-SVR	0.446	0.342	50.036
	WLS-SVR	0.342	0.272	28.453
	LAD-SVR	0.283	0.230	20.620
	WLAD-SVR	0.289	0.235	22.146
$0.9N(0, 0.2^2) + 0.1N(0, 2^2)$	LS-SVR	0.171	0.126	6.960
	WLS-SVR	0.063	0.050	0.935
	LAD-SVR	0.066	0.052	1.111
	WLAD-SVR	0.065	0.051	1.063
$0.8N(0, 0.2^2) + 0.2N(0, 2^2)$	LS-SVR	0.238	0.185	13.225
	WLS-SVR	0.081	0.063	1.617
	LAD-SVR	0.081	0.064	1.646
	WLAD-SVR	0.075	0.059	1.412
$0.6N(0, 0.2^2) + 0.4N(0, 2^2)$	LS-SVR	0.341	0.270	27.022
	WLS-SVR	0.187	0.145	8.889
	LAD-SVR	0.082	0.065	1.806
	WLAD-SVR	0.071	0.056	1.301

Kang-Mo Jung



Figure 4.1 The regression lines for the true function sin(3x)/(3x) under 6 error distributions The true (solid line : —), LS-SVR (dotted line …), WLAD-SVR (dashed line – – –)

No	Dataset	#explanatory	#train	#test	Method	RMSE	MAE	SSE/SST
1	AutoMPG	7	275	117	LS-SVR	0.091	0.064	0.184
					WLS-SVR	0.097	0.067	0.210
					LAD-SVR	0.077	0.058	0.134
					WLAD-SVR	0.075	0.055	0.125
2	Bodyfat	14	177	75	LS-SVR	0.053	0.043	0.099
					WLS-SVR	0.055	0.044	0.105
					LAD-SVR	0.026	0.009	0.023
					WLAD-SVR	0.026	0.009	0.023
3	Concrete	8	721	309	LS-SVR	0.124	0.099	0.368
					WLS-SVR	0.125	0.099	0.370
					LAD-SVR	0.100	0.076	0.239
					WLAD-SVR	0.119	0.092	0.337
4	Diabetes	2	31	12	LS-SVR	0.170	0.110	0.967
					WLS-SVR	0.167	0.111	0.925
					LAD-SVR	0.163	0.137	0.886
					WLAD-SVR	0.159	0.125	0.839
5	MCPU	6	147	62	LS-SVR	0.071	0.034	0.253
					WLS-SVR	0.104	0.038	0.535
					LAD-SVR	0.051	0.029	0.132
					WLAD-SVR	0.058	0.032	0.167
6	Pollution	16	42	18	LS-SVR	0.122	0.085	0.562
					WLS-SVR	0.126	0.092	0.599
					LAD-SVR	0.127	0.099	0.608
					WLAD-SVR	0.108	0.077	0.443
7	Pyrim	27	52	22	LS-SVR	0.168	0.094	0.717
					WLS-SVR	0.168	0.094	0.718
					LAD-SVR	0.189	0.120	0.906
					WLAD-SVR	0.162	0.084	0.662
8	Yacht	6	216	92	LS-SVR	0.154	0.107	0.323
					WLS-SVR	0.178	0.109	0.434
					LAD-SVR	0.139	0.076	0.264
					WLAD-SVR	0.138	0.075	0.258

 Table 4.3 Experimental results on the benchmark datasets

5. Conclusion

In this paper LAD-SVR and WLAD-SVR algorithms are presented based on the robustness of least absolute deviation for regression. The proposed methods provide robust SVR algorithms to reduce the influence of outliers. To solve the derived optimization problem using the least absolute deviation loss function and the squared regularization function we use an approximation iterative method. The experiments have been conducted on two artificial datasets and eight benchmark datasets. Experiments for both simulated and benchmark datasets demonstrate that the proposed methods have better performance than classical LS-SVR and WLS-SVR in points of generalization and stability. Further study on this topic will adopt WLAD-SVR to many applications in real world regression problems.

References

Amayri, O. and Bouguila, N. (2010). A study of spam filtering using support vector machines. Artificial Intelligence Review, 34, 73-108.

- Argyriou, A., Micchelli, C. A. and Pontil, M. (2009). When is there a representer theorem? Vector versus matrix regularizers. *Journal of Machine Learning Research*, 10, 2507-2529.
- Bao, Y., Hu, Z. and Xiong, T. (2013). A PSO and pattern search based memetic algorithm for SVMs parameters optimization. *Neurocomputing*, **117**, 98-106.
- Cao, L. J. and Tay, F. E. H. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, 14, 1506-1518.
- Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Computation*, **19**, 1155-1178. Chen, C., Li, Y., Yan, C., Guo, J. and Liu, G. (2017). Least absolute deviation-based robust support vector
- regression. Knowledge-Based Systems, 131, 183-194.
 Chen, C. F., Yan, C. Q. and Li, Y. Y. (2015). A robust weighted least squares support vector regression based on least trimmed squares. Neurocomputing, 168, 941-946.
- Cristianini, N. and Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernelbased learning methods, Cambridge University Press.
- De Brabanter, K., Pelckmans, K., De Brabanter, J., De Brabanter, M. Suykens, J. A. K., Hubert, M. and De Moor, B. (2009). Robustness of kernel based regression: a comparison of iterative weighting schemes. *International Conference on Artificial Neural Networks* (Eds. C. Alippi, M. Polycarpou, C. Panayiotou, G. Ellinas), 100-110, Springer.
- Giloni, A., Simonoff, J. S. and Sengupta, B. (2006). Robust weighted LAD regression. Computational Statistics & Data Analysis, 50 3124-3140.
- Graybill, F. A. (1983). Matrices with Applications in Statistics, 2nd Ed., Wadworth, Belmont.
- Hu, Z., Bao, Y. and Xiong, T. (2013). Electricity load forecasting using support vector regression with memetic algorithms. *Science World Journal*, Article ID 292575.
- Hwang, C. H. (2011). Asymmetric least squares regression estimation using weighted least squares support vector machine. Journal of the Korean Data & Information Science Society, 22, 999-1005.
- Isa, D., Lee, L. H., Kallimani, V. P. and RajKumar, R. (2008). Text Document preprocessing with the Bayes formula for classification using the support vector machine. *IEEE Transactions on Knowledge* and Data Engineering, 20, 1264-1272.
- Jung, K.-M. (2011). Weighted least absolute deviation lasso estimator. Communications for Statistical Applications and Methods, 18, 733-739.
- Jung, K.-M. (2013). Weighted support vector machine with the SCAD penalty. Communications for Statistical Applications and Methods, 20, 481-490.
- Keerthi, S. S., Sheavade, S. K., Bhattacharyya, C. and Murthy, K. R. K. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13, 637-649.
- Lee, D., Kim, G. and Lee, K. E. (2013). Soil moisture prediction using a support vector regression. Journal of the Korean Data & Information Science Society, 24, 401-408.
- Li, S. and Tan, M. (2010). Tuning SVM parameters by using a hybrid CLPSO-BFGS algorithm. Neurocomputing, 73, 2089-2096.
- Liu, J., Yang, Y., Fu, C., Guo, J. and Yu, Q. (2016). A robust regression based on weighted LSSVM and penalized trimmed squares. *Chaos Soliton Fractals*, 89, 328-334.
- Mitra, V., Wang, C. J. and Banerjee, S. (2007). Text classification: a least square support vector machine approach. Applied Soft Computing, 7, 908-914.

Rousseeuw, P. J. and Leroy, A. (1987). Robust regression and outlier detection, Wiley, New York.

- Seok, K. (2014). Semi-supervised regression based on support vector machine. Journal of the Korean Data & Information Science Society, 25, 447-454.
- Shen, K. Q., Ong, C. J. and Li, X. P. (2008). Feature selection via sensitivity analysis of SVM probabilistic outputs. *Machine Learning*, **70**, 1-20.
- Suykens, J. A. K., De Brabanter, J., Lukas, L. and Vandewalle, J. (2002). Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, **48**, 85-105.
- Suykens, J. A. K., Gestel, T. V., De Barbanter, J., De Moor, B. and Vandelwalle, J. (2002). Least squares support vector machines, World Scientific, Singapore.

Suykens, J. A. K. and Vandewalle, J. (1999). Least squares support vector machine classifiers. Neural Processing Letters, 9, 293-300.

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

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

- Wang, K., Zhang, J., Chen, Y. and Zhong, P. (2014). Least absolute deviation support vector regression. Mathematical Problems in Engineering, Article ID 169575.
- Wang, K. and Zhong, P. (2014). Robust non-convex least squares loss function for regression with outliers. *Knowledge-Based Systems*, **71**, 290-302.
- Wen, W., Hao, Z. F. and Yang, X. W. (2010). Robust least squares support vector machine based on recursive outlier elimination. *Soft Computing*, 14, 1241-1251.
- Xiong, T., Bao, Y. and Hu, Z. (2014). Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting. *Knowledge-Based Systems*, **55**, 87-100.
- Yang, X. W., Tan, L. and He, L. (2014). A robust least squares support vector machine for regression and classification with noise. *Neurocomputing*, 140, 41-52.
- Zhao, Y., Sun, J., Du, Z.-H., Zhang, Z., Zhang, Y. and Zhang, H. (2012). An improved recursive reduced least squares support vector regression. *Neurocomputing*, 87, 1-9.