Final Accepted Version, and this manuscript will be published in PAKDD 2018, Springer.

ε-Distance Weighted Support Vector Regression

Ge Ou[1], Yan Wang[1*], Lan Huang[1], Wei Pang[2*], and George Macleod Coghill[2]

[1] Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of
Education, College of Computer Science and Technology,
Jilin University, Changchun, 130012, China.
wy6868@jlu.edu.cn,
[2] Department of Computing Science,
University of Aberdeen, Aberdeen AB24 3UE, UK
pang.wei@abdn.ac.uk

Abstract. We propose a novel support vector regression approach called ε-Distance Weighted Support Vector Regression (ε-DWSVR). ε-DWSVR specifically addresses a challenging issue in support vector regression: how to deal with the situation when the distribution of the internal data in the ε-tube is different from that of the boundary data containing support vectors. The proposed ε-DWSVR optimizes the minimum margin and the mean of functional margin simultaneously to tackle this issue. To solve the new optimization problem arising from ε-DWSVR, we adopt dual coordinate descent (DCD) with kernel functions for medium-scale problems and also employ averaged stochastic gradient descent (ASGD) to make ε-DWSVR scalable to larger problems. We report promising results obtained by ε-DWSVR in comparison with five popular regression methods on sixteen UCI benchmark datasets.

Keywords: Regression Analysis, Support Vector Regression, Distance Weighted Support Vector Regression, Dual Coordinate Descent, Averaged Stochastic Gradient Descent

1 Introduction

Support Vector Regression (SVR) has recently received a significant amount of attention due to its competitive performance [1] compared with other regression approaches, including the method of least squares [2], Neural Networks (NN) [3], logistic regression [4], and ridge regression [5]. However, the performance of existing SVR systems tends to be sensitive to parameter values and easily affected by the distribution of data on the boundary. In this research, the internal data indicates the data which are densely distributed together in the ε-tube, and the boundary data indicates the data which are distributed on the boundary of the ε-tube, which generally contain many support vectors.

In this paper, we present a novel SVR approach by considering recent progress in support vector (SV) theory and addressing the above limitations.

In general, SVR constructs decision functions in high-dimensional space for linear regression while the training data are mapped to a kernel Hilbert feature space. ε-SVR [6] was the first popular SVR strategy. It aims to find a function whose deviation from the actually observed values for all the training data is not more than ε, thus forming the so-called ε-tube, to fit training data. To find the best fitting hyperplane, ε-SVR tries to maximize the minimum margin containing data in the ε-tube as much as possible, which is similar to Support Vector Machines (SVMs) [7]. However, ε-SVR is susceptible to the distribution of those boundary data. In fact, the optimization objective greatly depends on the margin between support vectors, and this makes the final fitting function heavily reliant on the distribution of the boundary data: if the distribution of the internal data is very different from that of the boundary data, the final fitting function may not be reliable.

Recent progress in SV theory [8, 9] suggests that maximizing the minimum margin, that is, the shortest distance from the instances to the separating hyperplane, is not the only optimization goal in order to achieve better learning performance. Unlike traditional SVMs, Distance-weighted Discrimination (D-WD) [8] maximize the mean of the functional margin (i.e. the harmonic mean of the distances of all data to the separating hyperplane), thus greatly improving the classification performance. Inspired by the idea of DWD, we can also improve the original optimization objective for our regression problems by introducing the concept of the mean of the functional margin in regression.

Considering the above limitations of existing SVR systems and recent progress in SV theory, we propose a novel SVR approach called ε-Distance Weighted Support Vector Regression (ε-DWSVR), which optimizes the minimum margin and the mean of functional margin simultaneously. To solve the optimization problem, ε-DWSVR adopts the dual coordinate descent (DCD) [10] strategy with kernel functions on medium-scale problems, and it also employs the averaged stochastic gradient descent (ASGD) [11] strategy to improve its scalability. A comparison of ε-DWSVR with five popular regression methods (i.e. ε-SVR, linear regression, NN, logistic regression, and ridge regression) on sixteen UCI benchmark datasets indicates ε-DWSVR outperforms these algorithms: ε-DWSVR fits better the distribution of the internal data in most cases, especially for those datasets with strong interference noise.

2 Background

Let $S = (X, Y)$ be a training set of n instances. $X = [x_1, ..., x_n]$ are the input instances where $x_i \in R^m$, and $Y = [y_1, ..., y_n]$ are the output instances where $y_i \in R$. For classification problems, $Y = \{+1, -1\}$ is the label set. For regression problems, Y is the corresponding target values, where $y_i \in \{-\infty, +\infty\}$. The objective function is $f(x) = w \cdot \phi(x_i) + b$, where $x \in R^m$, $w \in R^m$, and $\phi(\cdot)$ is the mapping function induced by a kernel K, i.e., $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$.

2.1 Recent Progress in SV Theory

Recently, SV theory has made great progress. SVM aims to maximize the minimum margin, which denotes the smallest distances of all instances to the separating hyperplane [7]. The optimization problem is represented as follows:

$$\begin{array}{l}\min\limits_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{n}\xi_i \\ s.t.\ y_i\left(w\cdot\phi(x_i)+b\right) \geq 1-\xi_i,\quad \xi_i \geq 0,\quad i=1,2,...,n.\end{array}$$

where C is the regularization parameter and ξ measures the losses of instances. DWD is proposed to solve data pilling problems [8], which uses a new criterion, that is, maximizing the mean of the functional margin, to replace the criterion of maximizing the minimum margin in SVM for solving the optimization problem [12]. DWD denotes the functional margin as $u_i = y_i(w\cdot\phi(x_i)+b)$ and let $r_i = u_i + \xi_i$ be the adjusted distance of the i-th data to the separating hyperplane, and the optimization problem is then given below:

$$\begin{array}{l}\min\limits_{w,b,\xi}\sum\limits_{i=1}^{n}\left(\frac{1}{r_i}+C\xi_i\right) \\ s.t.\ r_i = y_i\left(w\cdot\phi(x_i)+b\right)+\xi_i,\quad r_i \geq 0,\quad \xi_i \geq 0,\quad \|w\|^2 \leq 1,\quad i=1,2,...,n,\end{array}$$

Since SVR is the application of SV theory to regression problems, the fitting hyperplane is also affected by the distribution of the boundary data. When the distribution of the internal data is different from that of the boundary data, the fitting hyperplane produced by SVR may not be consistent with the actual data distribution, which is similar to the data piling problems. Therefore, we introduce recent progress in SV theory into the original optimization objective of SVR and hope that it will lead to better regression performance.

3 The Proposed ε-DWSVR

In this section, we propose the novel ε-DWSVR method, which applies the idea of the mean of the functional margin, and we adopt the DCD method to handle general conditions and employ the ASGD method to deal with larger problems.

3.1 The Formulation of ε-DWSVR

To simplify the complexity, we enlarge the dimension of the vectors w and $\phi(x_i)$ to handle the bias term b as in [13], i.e.,$w \leftarrow [w,b]^T$, $\phi(x_i) \leftarrow [\phi(x_i),\mathbf{1}]$. Thus the regression function becomes $f(x) = w\cdot\phi(x)$. Then the margin in regression will be the distance of the data to the fitting hyperplane, i.e., $|w\cdot\phi(x_i)-y_i| \,/\, \|w\|$. Based on the concept of margin, we define the functional margin in regression.

Definition 1. *The functional margin in regression is defined as follows:* $\gamma = (w\cdot\phi(x_i)-y_i)^2,\quad i=1,2,...,n.$

The functional margin in regression can describe the difference between the real values and the estimated ones. It also has a significant connection with the geometrical distance. If the value of w is determined, the ranking of all data to the fitting hyperplane with respect to the margin can be decided by the functional margin. Next, we define the mean of the functional margin in regression.

Definition 2. *The mean of the functional margin in regression is as follows:*

$$\bar{\gamma} = \frac{1}{n}\sum_{i=1}^{n}\left(w^T\phi(x_i) - y_i\right)^2 = \frac{1}{n}\left(w^T\phi(X)\phi(X)^T w - 2(\phi(X)Y)^T w + YY^T\right),$$

where $\phi(X) = [\phi(x_1), ..., \phi(x_n)]$ *and* $\phi(X)\phi(X)^T = \sum_{i=1}^{n}\phi(x_i)\phi(x_i)$.

Based on Definitions 1 and 2, we add the mean of the functional margin to original ε-SVR objective problems. As in the soft-margin of ε-SVR [6] we also consider the soft-margin in our problem. So the final optimal function is as follows:

$$\begin{aligned} &\min_{w,\xi,\xi^*} \frac{1}{2}\|w\|^2 + \lambda_1\bar{\gamma} + C\sum_{i=1}^{n}(\xi_i + \xi_i{}^*) \\ &s.t.\ y_i - w\cdot\phi(x_i) \le \varepsilon + \xi_i, \\ &\quad\ w\cdot\phi(x_i) - y_i \le \varepsilon + \xi_i{}^*,\quad \xi_i, \xi_i{}^* \ge 0,\quad i = 1, 2, ..., n, \end{aligned} \tag{1}$$

where λ_1 is the parameter for achieving the trade-off between the mean of functional margin and the model complexity.

In our ε-DWSVR, we maximize the minimum margin and minimize the mean of the functional margin at the same time, to obtain a better tradeoff between the distribution of the internal data and that of the boundary data. ε-DWSVR considers the influence of all data to the fitting hyperplane, as this is closer to the actual distribution of the internal data, and it is more robust to noise.

To illustrate the robustness of ε-DWSVR to noise and the differences between ε-SVR and that of ε-DWSVR, we use an example for comparison among linear regression, ε-SVR, and ε-DWSVR on an artificial dataset. In Fig. 1, the green points represent the data in which the distribution of the internal data is different from that of the boundary data, and the purple points represent noise. The cyan dashed curve, the grey dashed curve, and the red solid curve are produced by linear regression, ε-SVR, and ε-DWSVR, respectively.

Obviously, the curve produced by linear regression largely deviates from the actual distribution of the dataset, which indicates the linear regression is more sensitive to noise. ε-SVR and ε-DWSVR are more robust with the presence of noise, so the grey dashed curve and the red solid curve are within the area of non-noisy data. However, ε-SVR is controlled by boundary data containing many support vectors. Once the distribution of the internal data is different from that of the boundary data (which is the case in Fig. 1), ε-SVR may not achieve good performance. The grey dashed curve produced by ε-SVR is different from the curve produced by ε-DWSVR. Because ε-DWSVR considers the influence of all data to the fitting hyperplane, it is obvious that the red solid curve produced by ε-DWSVR is closer to the actual distribution of the internal data.

It is obvious that the optimization problem of (1) is more complicated than that of the original SVR. Thus, as mentioned before, to solve (1) and improve the scalability, we implement different methods for ε-DWSVR, that is, we adopt the DCD method with kernel functions for small and medium problems and the ASGD method for larger problems. These will be presented in the following sections.

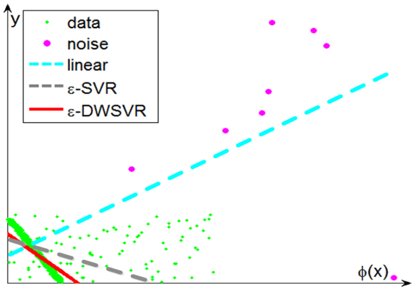

Fig. 1. The fitting curves produced by linear regression, ε-SVR, and ε-DWSVR in the original space. The data (green points) are composed of (1) 86.3% of all data which are evenly distributed across the line with a slope being -2 and $y \in [0,+\infty), x \in [0,10]$, and (2) 12.5% of all data which are evenly distributed on the line with a slope of 0 and $y \in [0,+\infty), x \in [0,40]$. This means the distribution of the internal data is different from that of the boundary data (those 12.5% of data). The rest 1.2% of data are noise (purple points). Due to noise, the cyan dashed curve produced by linear regression is very different from the rational one. The grey dashed curve produced by ε-SVR is adversely influenced by the distribution of the boundary data, while the red solid curve produced by ε-DWSVR better reflects the distribution of the internal data.

3.2 The Regression of Medium Problems with Kernel Functions

Considering the mean of the functional margin $\bar{\gamma}$ in (1) and Definition 2, we can obtain the following form:

$$\begin{array}{l}\min\limits_{w,\xi,\xi^*} \frac{1}{2}\|w\|^2 + \frac{\lambda_1}{n}\left(w^T\phi(X)\phi(X)^Tw - 2(\phi(X)Y)^Tw\right) + C\sum\limits_{i=1}^{n}(\xi_i + {\xi_i}^*)\\ s.t.\ y_i - w\cdot\phi(x_i) \le \varepsilon + \xi_i,\\ \qquad w\cdot\phi(x_i) - y_i \le \varepsilon + {\xi_i}^*,\quad \xi_i, {\xi_i}^* \ge 0,\quad i = 1,2,...,n.\end{array} \tag{2}$$

Here we omit the term YY^T in $\bar{\gamma}$ (Definition 2) because it is regarded as a constant in an optimization problem. Obviously, the high dimensionality of $\phi(\cdot)$ and its complicated form makes (2) intractable. To simplify (2), we take the suggestion from [14] and the optimal solution w in [9]. We first give the following theorem which can be proved.

Theorem 1. *The optimal solution w for (2) can be represented as follows: $w = \sum\limits_{i=1}^{n}(\alpha_i - \alpha_i^*)\cdot\phi(x_i) = \phi(X)(\alpha - \alpha^*)$, where $\alpha = [\alpha_1,...,\alpha_n]^T$ and $\alpha^* = [{\alpha_1}^*,...,{\alpha_n}^*]^T$ are the parameters of ε-DWSVR.*

According to Theorem 1, (2) can be cast as

$$\begin{array}{l}\min\limits_{\alpha,\alpha^*,\xi,\xi^*} \frac{1}{2}(\alpha-\alpha^*)^TQ(\alpha-\alpha^*) + p^T(\alpha-\alpha^*) + C\sum\limits_{i=1}^{n}(\xi_i + {\xi_i}^*)\\ s.t.\ y_i - (\alpha-\alpha^*)^TG_i \le \varepsilon + \xi_i,\\ \qquad (\alpha-\alpha^*)^TG_i - y_i \le \varepsilon + {\xi_i}^*,\ \ \xi_i, {\xi_i}^* \ge 0,\ \ i = 1,2,...,n,\end{array} \tag{3}$$

where $G = \phi(X)^T\phi(X)$, G_i denotes the i-th column of G, $Q = 2\lambda_1G^TG/n + G$, and $p = -2\lambda_1GY/n$. Thus (3) can be transformed into a dual formulation with

Lagrange multipliers, so the Lagrange function of (3) leads to

$$\begin{aligned} L = {} & \tfrac{1}{2}(\alpha-\alpha^*)^T Q\,(\alpha-\alpha^*) + p^T\,(\alpha-\alpha^*) + C\sum_{i=1}^{n}(\xi_i+\xi_i^*) - \sum_{i=1}^{n}(\eta_i\xi_i+\eta_i^*\xi_i^*) \\ & - \sum_{i=1}^{n}\beta_i\left(\varepsilon+\xi_i-y_i+(\alpha_i-\alpha_i^*)^T G\right) - \sum_{i=1}^{n}\beta_i\left(\varepsilon+\xi_i^*+y_i-(\alpha_i-\alpha_i^*)^T G\right), \end{aligned} \tag{4}$$

where $\eta,\eta^*,\beta,\beta^*$ are Lagrange multipliers. To satisfy the KKT conditions [15], we set the partial derivatives of $(\alpha-\alpha^*)$ and $\xi^{(*)}$ to zero and thus obtain the following equations:

$$\frac{\partial L}{\partial(\alpha-\alpha^*)} = Q\,(\alpha-\alpha^*) + p - \sum_{i=1}^{n}(\beta_i-\beta_i^*)G_i = 0, \tag{5}$$

$$\frac{\partial L}{\partial {\xi_i}^{(*)}} = C - \beta_i^{(*)} - \eta_i^{(*)} = 0, \quad i=1,2,...,n. \tag{6}$$

By substituting (5) and (6) into (4), and inspired by the work of [16], (4) can be written as follows to compute the values of $\begin{bmatrix}\beta \\ \beta^*\end{bmatrix}$ separately:

$$\begin{aligned} \min_{\beta,\beta^*} f(\beta,\beta^*) = {} & \tfrac{1}{2}\left[\beta^T,(\beta^*)^T\right]\begin{bmatrix} H & -H \\ -H & H\end{bmatrix}\begin{bmatrix}\beta \\ \beta^*\end{bmatrix} \\ & + \left[\varepsilon e^T + \left(\tfrac{2\lambda_1}{n}HY-Y\right)^T, \varepsilon e^T - \left(\tfrac{2\lambda_1}{n}HY-Y\right)^T\right]\begin{bmatrix}\beta \\ \beta^*\end{bmatrix} \\ s.t.\ & 0 \le \beta_i, {\beta_i}^* \le C, \quad i=1,2,...,n. \end{aligned} \tag{7}$$

where $H = GQ^{-1}G$, and e means the all-one vector.

We adopt the DCD method as in [10] to solve (7). This method continuously selects one variable for minimization and keeps others as constants at each iteration. In our situation, we minimize the variation of $f(\beta^{'})$ by adjusting the value of $\beta_k^{'}$ with a step size of t while fixing other $\beta_{l\neq k}^{'}$, where $\beta^{'} = (\beta,\beta^*)^T$, and the following equation needs to be solved: $\min\limits_t f(\beta^{'}+tb_k)\ \ s.t.\ 0 \le \beta_k^{'}+t \le C,\ \ k=1,2,...,2n$, where b_k means the vector with 1 in the k-th element and 0's elsewhere. Then, we have the form of this sub-problem as follows:

$$f(\beta^{'}+tb_k) = \frac{1}{2}h_{kk}t^2 + \nabla f(\beta^{'})_k t + f(\beta^{'}), \tag{8}$$

where h_{kk} is the diagonal entry of $\begin{bmatrix} H - H \\ -H\ H\end{bmatrix}$. It can be seen that $f(\beta^{'})$ is independent of t, so we omit this term in (8).

Hence $f(\beta^{'}+tb_k)$ is transformed into a simple quadratic function of t. Assume that $\beta_k^{'iter}$ is the value of $\beta_k^{'}$ at the $iter$-th iteration, then the value of $\beta_k^{'}$ at the $(iter+1)$-th iteration is $\beta_k^{'(iter+1)} = \beta_k^{'iter} + tb_k$. According to (8), the minimization of t which satisfies (8) is $t = -\frac{\nabla f(\beta^{'iter})_k}{h_{kk}}$. Considering the box constraint

$0 \le \beta_k^{'} \le C$, the minimization for $\beta_k^{'(iter+1)}$ has the following form:$\beta_k^{'(iter+1)} \leftarrow \min(\max(\beta_k^{'iter} - \frac{\nabla f(\beta^{'iter})_k}{h_{kk}}, 0), C)$. After $\beta^{'}$ converges, we can obtain $(\alpha - \alpha^*)$ according to (5) as follows: $(\alpha - \alpha^*) = Q^{-1}G\left(\frac{\lambda_1}{n}Y + (\beta - \beta^*)\right)$.

Therefore, the final fitting function becomes: $f(x) = \sum\limits_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x)$.

Algorithm 1 presents the steps of the DCD method for updating $\beta^{'}$.

Algorithm 1 ε-DWSVR with Kernel Functions

Input: Dataset X, Y, λ_1, C, ε, K; **Output:** $\alpha - \alpha^*$; **Initialization:** $\beta^{'} = \mathbf{0}, (\alpha - \alpha^*) = \frac{2\lambda_1}{n}Q^{-1}GY, A = Q^{-1}G, h_{kk} = b_k^T G Q^{-1} G b_k$;

```
while $\beta^{'}$ not converges do
    for $k = 1, 2, ..., 2n$ do
        $\nabla f(\beta^{'})_k \leftarrow \varepsilon + (G(\alpha - \alpha^*)b_k - y_k);\quad if \quad k = 1, 2, ..., n$
        $\nabla f(\beta^{'})_k \leftarrow \varepsilon - (G(\alpha - \alpha^*)b_k - y_{k-n});\quad if \quad k = n+1, n+2, ..., 2n$
        $\beta_k^{'temp} \leftarrow \beta_k^{'}$; $\beta_k^{'} \leftarrow \min(\max(\beta_k^{'} - \frac{\nabla f(\beta^{'})_k}{h_{kk}}, 0), C)$;
        for $i = 1, 2, ..., n$ do
            $(\alpha_i - \alpha_i^*) \leftarrow (\alpha_i - \alpha_i^*) + \left(\beta_k^{'} - \beta_k^{'temp}\right)Ab_k;\quad if \quad k = 1, 2, ..., n$
            $(\alpha_i - \alpha_i^*) \leftarrow (\alpha_i - \alpha_i^*) - \left(\beta_k^{'} - \beta_k^{'temp}\right)Ab_k;\quad if \quad k = n+1, n+2, ..., 2n$
        end for
    end for
end while
```

3.3 The Regression of Larger Problems

In regression analysis, processing larger datasets may increase the time complexity. Although the DCD method can solve ε-DWSVR efficiently for small and medium problems, it is not the best strategy for larger problems. To improve the scalability of ε-DWSVR, we adjust the ASGD method to ε-DWSVR, which can effectively deal with larger regression problems. ASGD solves the optimization problem by computing a noisy unbiased estimate of the gradient, and it randomly samples a subset of the training instances rather than using all data.

Considering the constraints in (2), we reformulate (2) as follows:

$$\begin{aligned} \min_{w} g(w) = \tfrac{1}{2}\|w\|^2 + \tfrac{\lambda_1}{n}\left(w^T X^T X w - 2(XY)^T w\right) \\ + C\sum_{i=1}^{n}\max\{0, y_i - w \cdot x_i - \varepsilon, w \cdot x_i - y_i - \varepsilon\} \end{aligned} \tag{9}$$

Computing the gradient of w in (9) is time consuming because we need all the training instances for computation, especially when the size of datasets is large. Considering this issue, we use Stochastic Gradient Descent (SGD) [17] to reduce the computational time for larger problems. The SGD method is a drastic simplification [17]: instead of calculating the gradient exactly, it computes a noisy unbiased estimation of the gradient at each iteration, which is done by randomly sampling part of the training instances. According to [17], the SGD method is expected to converge to the global optimal solution when the objective is convex.

Therefore, we give an unbiased estimation of the gradient $\nabla g(w)$ in our case. For representing the last term of (9) formally, we define a function $s(w)$ that has different values under different constraint conditions, as shown below:

$$s(w) = \begin{cases} -x_i, & i \in I_1 \\ x_i, & i \in I_2 \\ 0, & otherwise \end{cases}, \quad i = 1, 2, ..., n,$$

where $I_1 \equiv \{i\,|y_i - w \cdot x_i \leq \varepsilon\}$, and $I_2 \equiv \{i\,|w \cdot x_i - y_i \leq \varepsilon\}$. In order to obtain an unbiased estimation of the gradient $\nabla g(w)$, we first present the following theorem which can be proved for computing $\nabla g(w)$.

Theorem 2. *An unbiased estimate of the gradient $\nabla g(w)$ in (9) has the following form: $\nabla g(w, x_i) = 2\lambda_1 x_i x_i^T w + w - 2\lambda_1 y_i x_i + nC \cdot s(w)$, where (x_i, y_i) is an randomly sampled instance from the training set.*

Based on Theorem 2, the stochastic gradient can be updated iteratively as follows:

$$w_{t+1} = w_t - \varphi_t \nabla g_t\left(w_t, x_i\right), \tag{10}$$

where φ_t is the learning rate at the t-th iteration.

To make the solution to (9) more robust, we can adopt the ASGD method to solve the optimization problem in (9), which outperforms the SGD method [11]. In ASGD [11], a good choice for φ_t can be obtained by the form $\varphi_t = \varphi_0(1 + a\varphi_0 t)^{-c}$ to compute (10), where a, φ_0, and c are set by constant values as in [9]. In addition to updating the ordinary stochastic gradient in (10), we also compute $\bar{w}_t$ at each iteration as follows: $\bar{w}_t = \frac{1}{t-t_0}\sum\limits_{i=t_0+1}^{t} w_i$, where t_0 is used to decide when we apply the averaging process. This average value can also be calculated in a recursive manner as follows: $\bar{w}_{t+1} = \delta_t w_{t+1} + (1 - \delta_t)\bar{w}_t$.

Finally, Algorithm 2 presents the detailed steps of the ASGD method for larger problems, where $T * n'$ determines the number of iterations. T is a coefficient for adjusting the number of iterations with a default value of 5; n' is the sampling number from n instances with a value between 1000 and 1% of the training instances. The settings of these two variable values follow those in [9].

Algorithm 2 ε-DWSVR for Larger Problems

```
Input: Dataset $X$, $Y$, $\lambda_1$, $C$, $\varepsilon$; Output: $\bar{w}$
Initialization: $w_0 = 0, \nabla g_0 = 0, t = 1$
while $t \leq T * n'$ do
    Randomly select one instance $(x_i, y_i)$ from the training set;
    Compute $\nabla g_t(w_t, x_i)$;$w_{t+1} \leftarrow w_t - \varphi_t \nabla g_t(w_t, x_i)$;$\bar{w}_{t+1} \leftarrow \delta_t w_{t+1} + (1 - \delta_t)\bar{w}_t$;
    $t \leftarrow t + 1$;
end while
```

4 Experiments

In this section, we compare the fitting performance between ε-DWSVR and other regression methods on several real datasets to assess whether our method has better fitting performance.

4.1 Experimental Setup

We select sixteen datasets from UCI [18] to perform the evaluations on ε-DWSVR, ε-SVR, linear regression, NN, logistic regression, and ridge regression. This includes eight medium-scale datasets and eight larger datasets. The characteristics of all datasets are in Table 1. All the features of the datasets and target set are normalized into [0,1] to balance the influence of each feature. After normalization, for preprocessing the data, we use PCA with 95% for feature extraction to reduce the interference of irrelevant attributes. During the construction of the model, we divide the datasets into training sets and test sets by 5-fold cross validation. Parameters selections are processed on the test sets to obtain better experimental results.

Table 1. The characteristics of benchmark datasets.

Scale	Datasets	Instances	Features	Datasets	Instances	Features
medium	Slump	103	7	Housing	506	14
	Automobile	205	26	Stock	536	9
	Yacht	308	7	Concrete	1030	8
	Auto MPG	398	8	Music	1059	68
larger	Crime	1994	128	Bike	17389	16
	SkillCraft	3338	18	ONP	39797	61
	CCPP	9568	4	CASP	45730	9
	Drift	13910	129	Buzz	140000	77

Finally, we use mean square error (MSE) [19] as the evaluation metric, and evaluations are also processed on the test sets. The experiments are repeated 30 times, and the average values of the evaluation metric are recorded. For medium-scale datasets, we evaluate both the linear and RBF kernels [7]. In addition, we record the computational time for larger datasets.

4.2 Results and Discussion

For medium-scale datasets, Table 2 summarizes the results of MSE on all methods, including linear kernel function and RBF kernel function for ε-DWSVR and ε-SVR. As one can see from Table 2, the fitting performance of ε-DWSVR is much better than ε-SVR, which indicates that ε-DWSVR is more competitive than ε-SVR. Besides, the Housing dataset is ideal with less noise and a consistent distribution of overall data; thus linear regression works better on this dataset. The average MSE values on all datasets are shown in Table 2 and the best ones are indicated in bold.

Table 2. The evaluation of average MSE on medium-scale datasets.

Datasets	ε-DWSVR (RBF)	ε-SVR (RBF)	ε-DWSVR (Linear)	ε-SVR (Linear)	LINEAR	NN	Logistic	Ridge
Slump	**0.0036**	0.0037	0.0047	0.0050	0.0063	0.0055	0.0054	0.0215
Automobile	**0.0057**	0.0063	0.0092	0.0102	0.0094	0.0129	0.0136	0.0232
Yacht	**0.0101**	0.0166	0.0154	0.0171	0.0180	0.0175	0.0171	0.0434
Auto MPG	**0.0133**	0.0137	0.0135	0.0136	0.0140	0.0148	0.0152	0.0380
Housing	0.0142	0.0170	0.0169	0.0176	**0.0117**	0.0199	0.0182	0.0178
Stock	**0.0080**	0.0083	0.0087	0.0088	0.0101	0.0111	0.0093	0.0148
Concrete	**0.0227**	0.0251	0.0256	0.0257	0.0262	0.0267	0.0261	0.0362
Music	**0.0306**	0.0348	0.0359	0.0360	0.0368	0.0388	0.0408	0.0594

For larger datasets, Fig. 2 shows the results of MSE on all methods. We can see that ε-DWSVR performs better than other methods on most datasets. In addition, the Drift dataset contains less noise, and there exists a consistent distribution of all data. So linear regression works better on this dataset. Besides, linear regression did not return the results on some datasets after 48 hours.

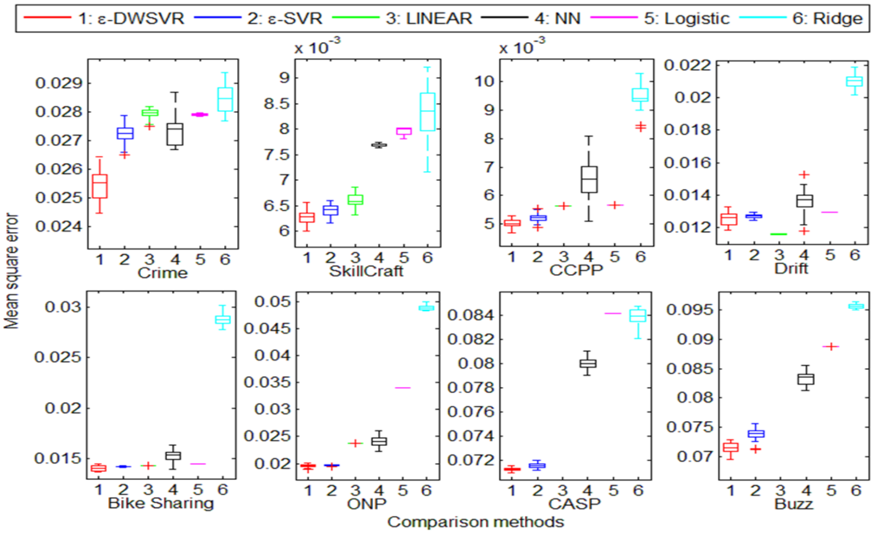

Fig. 2. The evaluation of MSE on larger datasets.

4.3 Parameter Effects

ε-DWSVR has three main parameters: λ_1, C, and ε. To further investigate the influence of these three parameters, we evaluate the MSE value by changing one of them on the medium-scale datasets and larger datasets, while fixing other parameters. Fig. 3 and Fig. 4 show that the MSE on the medium-scale and larger datasets does not change significantly with the change of the parameters. This indicates that the performance of ε-DWSVR is not sensitive to parameter values, which demonstrates the robustness of ε-DWSVR.

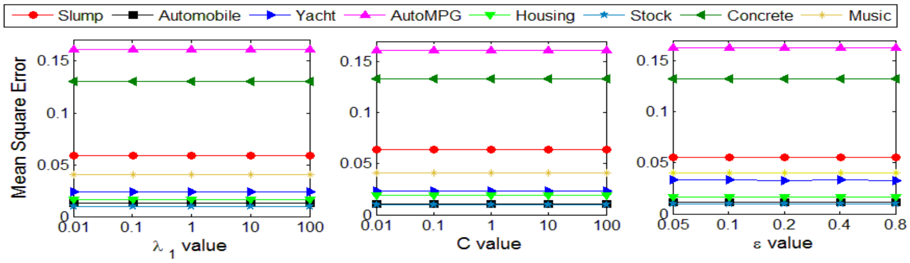

Fig. 3. Parameter influence on medium-scale datasets.

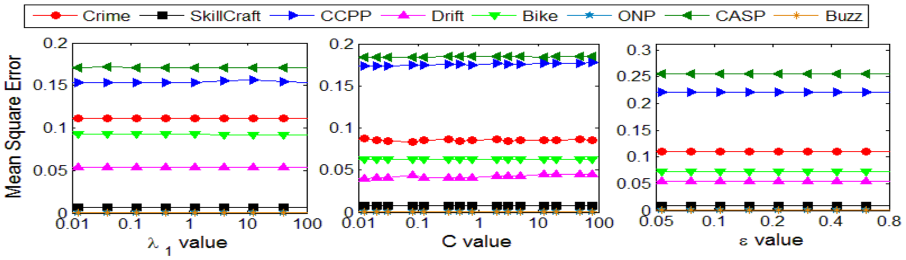

Fig. 4. Parameter influence on larger datasets.

4.4 Time Cost

We present a comparison of CPU time taken between ε-SVR and ε-DWSVR on each larger dataset in Fig. 5. For ε-SVR, C is set to 1; ε is set to 0.1. For ε-DWSVR, λ_1 is set to 1; C is set to 10; ε is set to 0.1. ε-SVR for larger problems was implemented by the LIBLINEAR [13] package and ε-DWSVR was implemented by ASGD. Fig. 5 shows that ε-DWSVR cost less time than ε-SVR on most datasets, and it is only slightly slower than ε-SVR on two datasets.

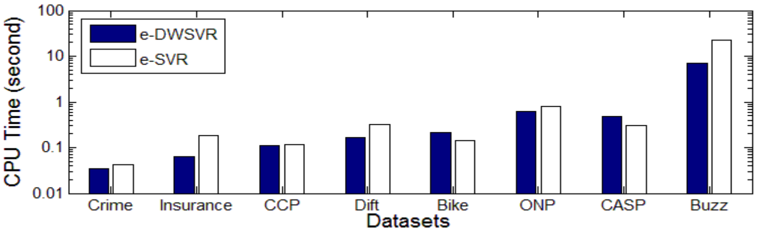

Fig. 5. The CPU time on larger datasets.

Acknowledgement

We gratefully thank Dr Teng Zhang and Prof Zhi-Hua Zhou for providing the source code of "LDM", and their kind technical assistance. We also thank Prof Chih-Jen Lins team for providing the LIBSVM and LIBLINEAR packages and their support. This work is supported by the National Natural Science Foundation of China (Grant Nos.61472159, 61572227) and Development Project of Jilin Province of China (Grant Nos. 20140101180JC, 20160204022GX, 20180414012G H). This work is also partially supported by the 2015 Scottish Crucible Award funded by the Royal Society of Edinburgh and the 2016 PECE bursary provided by the Scottish Informatics & Computer Science Alliance (SICSA).

References

1. Brown, J.D., Summers, M.F., Johnson, B.A.: Prediction of hydrogen and carbon chemical shifts from rna using database mining and support vector regression. Journal of Biomolecular NMR **63**(1) (2015) 1–14
2. Rajaraman, P.K., Manteuffel, T.A., Belohlavek, M., Mcmahon, E., Heys, J.J.: Echocardiographic particle imaging velocimetry data assimilation with least square finite element methods. Computers and Mathematics with Applications **68**(11) (2016) 1569–1580
3. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, L., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1) (2014) 1929–1958
4. Ke, Y., Fu, B., Zhang, W.: Semi-varying coefficient multinomial logistic regression for disease progression risk prediction. Statistics in Medicine **35**(26) (2016) 4764–4778
5. Dicker, L.H.: Ridge regression and asymptotic minimax estimation over spheres of growing dimension. Bernoulli **22**(1) (2016) 1–37
6. Demir, B., Bruzzone, L.: A multiple criteria active learning method for support vector regression. Pattern Recognition **47**(7) (2014) 2558–2567
7. Vapnik, V.: The nature of statistical learning theory. Springer-Verlag (1995)
8. Marron, J.S.: Distance-weighted discrimination. Journal of the American Statistical Association **102**(December) (2007) 1267–1271
9. Zhang, T., Zhou, Z.H.: Large margin distribution machine. In: Proceedings of the Twenthieth ACM SIGKDD international conference on Knowledge discovery and data mining, Banff, Alberta, Canada, ACM Press (2014) 313–322
10. Yuan, G.X., Ho, C.H., Lin, C.J.: Recent advances of large-scale linear classification. Proceedings of the IEEE **100**(9) (2012) 2584–2603
11. Xu, W.: Towards optimal one pass large scale learning with averaged stochastic gradient descent. Computer Science (2011)
12. Qiao, X.Y., Zhang, L.S.: Distance-weighted support vector machine. Statistics and Its Interface **8**(3) (2015) 331–345
13. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of Machine Learning Research **9**(12) (2010) 1871–1874
14. Scholkopf, B., Smola, A.: Learning with kernels: Support vector machines, regularization, optimization, and beyond. Journal of the American Statistical Association **16**(3) (2011) 781–781
15. Izmailov, A.F., Solodov, M.V.: Karush-kuhn-tucker systems: regularity conditions, error bounds and a class of newton-type methods. Mathematical Programming **95**(3) (2003) 631–650
16. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. Acm Transactions on Intelligent Systems and Technology **2**(3) (2011) 389–396
17. Bottou, L.: Large-Scale Machine Learning with Stochastic Gradient Descent. Physica-Verlag HD (2010)
18. Lichman, M.: UCI machine learning repository. http://archive.ics.uci.edu/ml (2013) last accessed: 01 August 2016.
19. Guo, D.N., Shamai, S., Verdu, S.: Mutual information and minimum mean-square error in gaussian channels. IEEE Transactions on Information Theory **51**(4) (2005) 1261–1282