Iteratively reweighted least squares for robust regression via SVM and ELM

Hongwei Dong, Liming Yang*

College of Science, China Agricultural University, Beijing, 100083, China

Abstract

Most machine learning methods implement robust learning by reweighted strategy. To overcome the optimization difficulty of the implicitly reweighted robust methods (including modifying loss functions and objectives), we try to use a more direct method: explicitly iteratively reweighted method to handle noise (even heavy-tailed noise and outlier) robustness. In this paper, an explicitly iterative reweighted framework based on two kinds of kernel based regression algorithm (LS-SVR and ELM) is established, and a novel weight selection strategy is proposed at the same time. Combining the proposed weight function with the iteratively reweighted framework, we propose two models iteratively reweighted least squares support vector machine (IRLS-SVR) and iteratively reweighted extreme learning machine (IRLS-ELM) to implement robust regression. Different from the traditional explicitly reweighted robust methods, we carry out multiple reweighted operations in our work to further improve robustness. The convergence and approximability of the proposed algorithms are proved theoretically. Moreover, the robustness of the algorithm is analyzed in detail from many angles. Experiments on both artificial data and benchmark datasets confirm the validity of the proposed methods.

Keywords: Robust methods, Kernel based regression, Iteratively reweighted least squares, Support vector machines, Extreme learning machines

*Corresponding author

Email addresses: donghongwei1994@163.com (Hongwei Dong), cauyanglm@163.com (Liming Yang)

1. Introduction

Machine learning algorithms can be divided into pattern classification and regression, according to different task background. For pattern classification, some deep representation learning based algorithms are quite mature and have been proved to be effective [1, 2]. However, it seems to be not effective enough of solving the regression tasks. For a general regression problem, linear regression or kernel based regression (KBR) still dominates [3]. For a dataset $Z = \{(\boldsymbol{x}_i, y_i)|i = 1, 2, \cdots, N\} \in \mathcal{R}^n \times \mathcal{R}$ and a nonlinear prediction function $f(\cdot)$. A classical and effective framework of KBR is pursuing structural risk minimization plus empirical risk [4]-[5] minimization (ERM) as following

$$\arg\min_{f} \lambda||f||^p + \frac{1}{N}\sum_{i=1}^{N} \ell(y_i - f(\boldsymbol{x}_i)) \tag{1}$$

In this objective, structural risk term controls complexity and empirical risk term controls the size of the modeling error through a loss function $\ell(\cdot)$. The most common and effective of these is least squares regression which focus on minimizing the mean square error. The reason lies on the empirical mean has an optimal minimax mean square error among all mean estimators in all models including Gaussian distributions [6]. This means that in general, least squares regression has the optimal mean of estimation. However, samples are often contaminated with noise and outliers because of erroneous samplings and measurements in practical applications and it is meaningless to estimate the mean value under the influence of noise especially heavy-tailed noise. Thus, some sub-optimal estimators are widely studied to alleviate the problem of noise robustness [6–9]. In the author's opinion, the central idea of these biased ERM estimators is to assign corresponding weights to samples, which can be divided into explicitly and implicitly ones according to the way of giving weights. For the explicit methods, different weights are given to the loss of each sample through a weight generation strategy. Various loss functions [8, 10, 11] and some modifications to the objective function [12, 13] are all means to achieve implicitly reweighted. Because of the same core idea, we think these three kinds of methods can be transformed into each other.

The establishment of robust methods is mostly based on the change of loss function, such as Hampel, Tukey, Bisquare, Welsch, Weibull et al [7], and the form of them are plotted in Figure 1. Some machine learning algorithms based on new robust loss functions have also been extensively studied [14, 15]. Recently, to handle robust regression, especially under the interference of heavy-tailed distribution noise, some truncated minimization methods have been proved to be effective [6, 16] and these methods belongs to the modifications to objective. However, the novel robust loss functions are often non-convex and the objectives of these truncated minimization methods are often complex. Compared with ordinary mean square error minimization, they are much more difficult to optimize. In contrast, the explicitly reweighted methods do not have this problem.

To achieve robust regression under noise disturbance, especially heavy-tailed noise, and to give a fast and simple solution for the implicitly reweighted methods. In this paper, we focus on the explicitly reweighted methods under the classical least squares based KBR framework, which directly assigning different weights to samples to achieve noise robust regression. Considering some good properties of sigmoid function, we propose a novel weight design method and combine it with two mature least squares based kernel regression methods, (LS-SVR) [17] and extreme learning machine (ELM) [18]. SVMs is based on the idea of maximum margin and ELM is a BP neural network with random hidden layer weights. Robust SVR [19–22] or ELM [15, 23] based on implicitly weighting methods have also been extensively studied. Unlike the implicitly reweighted methods, the explicit reweighted methods are more convenient to implement and suitable for large-scale problems. Because they have the same inherent meaning, we prove that they can also be transformed into each other to some extent, which means that the explicit reweighted method is equivalent to corresponding implicitly reweighted method in certain cases and it also has satisfactory noise robustness.

The main contributions of this paper can be summarized as follows:

1. The explicit reweighted method is redesigned and a new weighting strategy based on sigmoid function is proposed and applied to SVR and ELM to adapt to the outlier robust regression.
2. We prove that explicitly reweighted and implicitly reweighted can be transformed into each other under certain conditions. On this basis, convergence and robustness of the proposed method can be theoretically proven.
3. The robustness of proposed methods for outliers is demonstrated in artificial and benchmark datasets.

The rest of this paper is organized as follows: Some relevant technical literatures are reviewed in section 2. The proposed strategies and relevant analysis are listed in section 3. Robustness is discussed from two angles in section 4. In section 5, experimental results are exhibited. Conclusion and possible future directions for further development are given in section 6.

2. Background

2.1. Analysis of robust methods

For the regression problem without noise interference, we know that a ERM term in Eq. (1) with least square loss ($\ell(u) = u^2$) is an unbiased estimate of the expected risk. This is also the reason why the least squares method is effective, which can achieve the optimal mean estimation and is easy to solve. However, it is another story when the data is contaminated by noise. When the data contains noise with outliers or heavy-tailed distributions, the following formula holds

$$E_{X,Y}[\ell(Y, f(X))] \neq \frac{1}{N}\sum_{i=1}^{N}(y_i - f(\boldsymbol{x}_i))^2 \to \infty \tag{2}$$

therefore, the mean estimation often fails in the case of noise [24] and algorithms with robustness to noise is needed. To handle this problem, some robust loss functions are studied to replace least square loss. A effective way is to use M-estimator loss functions [7]

$$\min_{f} \frac{1}{N}\sum_{i=1}^{N}\rho(y_i - f(\boldsymbol{x}_i)) \tag{3}$$

where $\rho(\cdot)$ is quadratic near the origin, and the other parts are linear. In this way, the loss of large residuals will not be too big. Similar outlier robust losses have been extensively studied. Most of them truncate the non-robust losses to achieve non-incremental losses with large residual values [25, 26]. In fact, this can be seen as an implicitly reweighted. By modifying the loss function, a small loss weight is given to the sample which would cause great loss and the original large loss is reduced. On the other hand, giving weights to each sample can also be considered as a special loss function. Moreover, Some typical methods for solving the non-convex loss functions such as difference of convex algorithm (DCA) [27, 28], concave-convex procedure (CCCP)

[29] and half-quadratic optimization [30, 31], have been proved to be an iterative variant of the explicitly reweighted method [32–34]. Later, we will give the correspondence between changing loss implicitly reweighted method and explicit weighting method.

Some novel objective functions are also studied to solve robust regression. In [9], they call similar methods truncated minimization problems. A min-max objective of ridge regression are proposed in [6] as follows

$$\min_{\mathbf{w}} \max_{\mathbf{u}} \lambda(||\mathbf{w}||^2 - ||\mathbf{u}||^2) + \frac{1}{\alpha N} \sum_{i=1}^{N} \psi_C[\alpha(y_i - \boldsymbol{x}_i^T \mathbf{w})^2 - \alpha(y_i - \boldsymbol{x}_i^T \mathbf{u})^2] \tag{4}$$

where $\lambda > 0$ controls trade-off and $\psi_C(\cdot)$ is a sigmoidal truncation function

$$\psi_C(u) = \begin{cases} -\log(1 - u + u^2/2), & 0 \leq u \leq 1 \\ \log(2), & u \geq 1 \\ -\psi_C(-u), & u \leq 0 \end{cases} \tag{5}$$

This can be seen as an effective approximation of ridge regression and it has certain robustness to outliers. In Eq. (4), we notice that α acts as a scaling parameter in the formula, and this new objective function is actually to measure the loss instead of residuals by M-estimator robust loss functions Eq.(5). On basis of Catoni's work, Zhang [9] proposed a truncated minimization problem for linear regression:

$$\min_{\mathbf{w}} \frac{1}{\alpha N} \sum_{i=1}^{N} \psi(\alpha |y_i - \boldsymbol{x}_i^T \mathbf{w}|) \tag{6}$$

it can be seen as a truncated median minimization problem and stochastic normalized gradient descent optimization method is used to handle the non-convexity of the objective [35]. In [12], a novel objective for heavy-tailed noise robust regression was proposed:

$$\begin{aligned} &\min_{f} \quad \mu_f \\ &s.t. \ \ \frac{1}{\alpha N} \sum_{i=1}^{N} \psi[\alpha(f(\boldsymbol{x}_i) - \mu_f)] = 0 \end{aligned} \tag{7}$$

where $\psi(\cdot)$. It is a extension of Eq. (4) to use a M-estimator loss function to obtain a robust location estimation for loss and an elegant proof of error bound was given. Similar to our work, a robust biased objective was given in [16]:

$$\min_{f}\{\min_{\theta}\frac{1}{N}\sum_{i=1}^{N}\rho(\frac{\ell(y_i-f(\boldsymbol{x}_i))-\theta}{s})\} \tag{8}$$

it can be seen that this is a two-stage optimization with the idea of obtaining a robust location estimation θ for loss as the objective to approximate the expected risk instead of ERM. In addition, a fast solution method based on iterative least squares is given, and their proof also reflects the relationship between the implicitly reweighted method by modifying the objective and the explicitly reweighted method.

It is worth to note that the above two robust methods based on loss modification and objective modification have following similarities.

- Both methods change the original objective function of KBR by different means, and the complexity of the new objectives is relatively heavy, so it is difficult to optimize it.

- The fundamental purpose of both is to reduce the impact of outliers on the model, that is, to reduce their losses. That is completely consistent with the origin of the explicitly reweighted robust methods and can be regarded as an implicitly reweighted method.

- To some extent, these two methods are equivalent to the explicitly reweighted methods. Whether modifying the loss function [32, 33] or modifying the objective [16], some studies have found that they can be transformed into an iterative reweighted pipeline to solve.

- Moreover, there is an inseparable relationship between the two kinds of robust methods. Compound function as observed in Eq. (6), $\psi(|\cdot|)$ can be regarded as a single non-convex function. This means that the two can also be transformed under certain conditions.

Based on the above analysis, explicitly reweighted robust methods are easy to implement, so we study a direct expansion of the explicitly reweighted methods: iteratively reweighted least squares (IRLS) based robust method in this paper. IRLS has been studied to a certain extent, and it is widely used in non-convex optimization [8, 36, 37], sparse representation [38, 39] and so on. We concisely present the basic principles of the IRLS strategy and relevant theoretical exposition can be found in [40]. For a kernel based regression problem, take LS-SVR as an example, a sequence of successive minimizers of a weighted least squares theoretical regularized risk is defined as follows

$$f_{k+1} = \arg\min_{f\in\mathcal{H}} \lambda||f||^2_{\mathcal{H}} + E(v(Y - f_k(X)(Y - f(X))^2)) \tag{9}$$

where k represents the times of iteration, $v(\cdot)$ is a weight function. In fact, implicitly reweighted method by modifying loss function can be directly transformed to IRLS. For an arbitrarily loss $\rho(\cdot)$, its gradient function $\psi(\cdot)$ and weight function $v(\cdot)$ are defined as follows:

$$\psi(x) = \partial\rho(x)/\partial x \tag{10}$$

$$v(x) = \begin{cases} \psi(x)/2x, & x \neq 0 \\ \psi'(0), & x = 0 \end{cases} \tag{11}$$

although not mentioned in this article, a robust scale estimation can be considered, similar to M-estimators. However, improper selection of the scale may result in inability to converge.

To let the sequence $\{f_k\}$ converge, the following conditions for weight function $v(\cdot)$ have been proved to be necessary in [40]:

v1 $v(x)$ a non-negative bounded Borel measurable function.

v2 $v(x)$ is an even function.

v3 $v(x)$ continuous and differentiable with $v'(x) \leq 0$ for $x > 0$.

Moreover, the solution of KBR with a convex loss $\rho(\cdot)$ instead of L_2-loss can be obtained as the limit of a sequence of IRLS minimization with arbitrary initial fit. If $\rho(\cdot)$ non-convex, the $\{f_k\}$ can be a local minimum depending on the initial start. The convergence and approximability of this sequence will be proved latter.

2.2. Representation of LS-SVR and ELM

In this section, we give a concisely review of LS-SVR [17]. The training set $Z = \{(x_i, y_i)|i = 1, 2, \cdots, N\} \in \mathcal{R}^n \times \mathcal{R}$. LS-SVR solves the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N} e_i^2 \\ s.t. \quad & y_i = \boldsymbol{w^T}\phi(\boldsymbol{x}_i) + b + e_i, \quad i = 1, 2, \cdots, N \end{aligned} \tag{12}$$

where $\phi(\cdot) : \mathcal{R}^n \rightarrow \mathcal{R}$ a function which maps the input space into a higher dimensional space, $e_i \in \mathcal{R}$ represents the error variables, $b \in \mathcal{R}$ represents the bias, and $C > 0$ is the regularization parameter which balances the structural risk and empirical risk. The optimization problem Eq. (12) usually convert into its dual problem by introducing Lagrangian multiplier $\boldsymbol{\alpha}$.

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{e}, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N} \boldsymbol{\alpha}_i(\boldsymbol{w}^{\boldsymbol{T}}\phi(\boldsymbol{x}_i) + b + e_i - y_i) \quad (13)$$

from KKT conditions, we derive

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N} \boldsymbol{\alpha}_i \phi(\boldsymbol{x}_i) = 0 \quad (14)$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{N} \boldsymbol{\alpha}_i = 0 \quad (15)$$

$$\frac{\partial \mathcal{L}}{\partial e_i} = Ce_i - \boldsymbol{\alpha}_i = 0 \quad (16)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_i} = \boldsymbol{w}^{\boldsymbol{T}}\phi(\boldsymbol{x}_i) + b + e_i - y_i = 0 \quad (17)$$

After elimination of $\boldsymbol{w}, \boldsymbol{\varepsilon}$ one obtains the solution

$$\begin{bmatrix} 0 & \boldsymbol{1}^{\boldsymbol{T}} \\ \boldsymbol{1} & K + \frac{1}{C}E \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{Y} \end{bmatrix} \quad (18)$$

where $\boldsymbol{Y} = [y_1, y_2, \cdots, y_N]^T$, $\boldsymbol{1} = (1, 1, \cdots, 1)^T \in \mathcal{R}^N$, E denotes $N \times N$ identity matrix, K is the kernel matrix with $K_{ij} = \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$. The resulting function for prediction as

$$y(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \quad (19)$$

where $\boldsymbol{\alpha}, b$ are the solutions of Eq. (18).

ELM is proposed for training the generalized single-hidden layer feedforward neural networks (SLFNs). Training samples are nonlinearly mapped to the feature space through stochastic initialization of hidden layer parameters, and the output function of ELM with L hidden nodes can be represented as

$$f(\boldsymbol{x}) = \sum_{i=1}^{L} h_i(\boldsymbol{x})\beta_i = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta} \tag{20}$$

where $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \cdots, h_L(\boldsymbol{x}))$, $h_i(\boldsymbol{x})$ is the hidden layer function $g(\boldsymbol{\alpha}_i, b_i, \boldsymbol{x})$ between the input layer and the ith hidden node($\boldsymbol{\alpha}_i$,b_i randomly pick) and $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_L)^T$ is the output weight between the hidden node and the output node. ELM minimizes the least square error as following

$$\min_{\beta} ||H\boldsymbol{\beta} - \boldsymbol{Y}||_2^2 \tag{21}$$

where $H = (\boldsymbol{h}(\boldsymbol{x}_1), \boldsymbol{h}(\boldsymbol{x}_2), \cdots, \boldsymbol{h}(\boldsymbol{x}_N))^T$ and $Y = (y_1, y_2, \cdots, y_N)^T$. The solution can be obtained by

$$\boldsymbol{\beta}^* = H^{\dagger}\boldsymbol{Y} \tag{22}$$

In order to maintain the complexity of the model and avoid over-fitting, a regularized ELM can be expresses as

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \\ s.t. \quad & y_i = h(\boldsymbol{x}_i)\boldsymbol{\beta} + \xi_i, \quad i = 1, 2, \cdots, N \end{aligned} \tag{23}$$

and the solution is

$$\boldsymbol{\beta}^* = \begin{cases} H^T(\frac{1}{C} + HH^T)^{-1}Y, & N < L \\ (\frac{1}{C} + H^TH)^{-1}H^TY, & N \geq L \end{cases} \tag{24}$$

In this paper, we choose the above two least squares based KBR methods as the basis of achieving robust regression, and apply the explicitly reweighted strategy to the both.

3. Main results

3.1. Iteratively reweighted least squares support vector regression(IRLS-SVR)

Put the IRLS strategy into the framework of LS-SVR, for the regression function $f = \boldsymbol{w}^T\phi(\boldsymbol{x}) + b$, a sequence of minimizers of weighted SVR can be written as follows:

$$\begin{aligned}&(\boldsymbol{w}_{k+1}, b_{k+1}) = \\ &\arg\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N} v(y_i - (\boldsymbol{w}_k^T\phi(\boldsymbol{x}_i) + b_k))(y_i - (\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b))^2\end{aligned} \tag{25}$$

where $\boldsymbol{w}_k, b_k$ are obtained from iteration, $v(\cdot)$ is the weight function which satisfies the requirements of iterative convergence conditions **v1** to **v3**, and the $k+1$th iteration as follows:

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N} v(y_i - (\boldsymbol{w}_k^T\phi(\boldsymbol{x}_i) + b_k))(y_i - (\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b))^2 \tag{26}$$

Eq. (26) can be rewritten as follows

$$\begin{aligned}(\boldsymbol{w}_{k+1}, b_{k+1}) = \arg\min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N} v(\xi_i^{(k)})\xi_i^2 \\ s.t. \quad & y_i = \boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b + \xi_i, \quad i = 1, 2, \cdots, N.\end{aligned} \tag{27}$$

where $v(\xi_i^{(k)})$ represents the weight of ith sample based on kth residual $\xi_i^{(k)}$. By introducing the Lagrangian multiplier $\boldsymbol{\alpha}$, we have the Lagrangian function

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N} v(\xi_i^{(k)})\xi_i^2 - \sum_{i=1}^{N}\boldsymbol{\alpha}_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b + \xi_i - y_i) \tag{28}$$

from KKT conditions, we derive

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N}\boldsymbol{\alpha}_i\phi(\boldsymbol{x}_i) = 0 \tag{29}$$

$$\frac{\partial\mathcal{L}}{\partial b} = -\sum_{i=1}^{N}\boldsymbol{\alpha}_i = 0 \tag{30}$$

$$\frac{\partial\mathcal{L}}{\partial\xi_i} = Cv(\xi_i^{(k)})\xi_i - \boldsymbol{\alpha}_i = 0 \tag{31}$$

$$\frac{\partial\mathcal{L}}{\partial\alpha_i} = \boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b + \xi_i - y_i = 0 \tag{32}$$

eliminating the variable $\boldsymbol{w}$ and $\boldsymbol{\xi}$, Eq. (29) to Eq. (32) can be transformed as

$$\begin{bmatrix} 0 & \mathbf{1^T} \\ \mathbf{1} & K+V \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{Y} \end{bmatrix} \tag{33}$$

where $\boldsymbol{Y} = (y_1, y_2, \cdots, y_N)^T$. $K_{ij} = \phi^T(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$ and V denotes the weight matrix:

$$V = diag(\frac{1}{Cv(\xi_1^{(k)})}, \frac{1}{Cv(\xi_2^{(k)})}, \cdots, \frac{1}{Cv(\xi_N^{(k)})}) \tag{34}$$

which can be calculated after every iteration. By comparison, we can see that the form of Eq. (27) is very similar to Eq. (12) and the difference is whether the loss is affected by the weight function. The prediction function can be expressed as:

$$y = \sum_{i=1}^{N} \boldsymbol{\alpha}_i^* K(\boldsymbol{x}, \boldsymbol{x}_i) + b^* \tag{35}$$

where $\boldsymbol{\alpha}^*, \boldsymbol{b}$ represents the final result of the iteration process.

Algorithm 1 Iterative reweighted robust method with LS-SVR (IRLS-SVR)

```
Input:
    Training set $T = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, $\boldsymbol{x}_i \in \mathcal{R}^n$.
    Gaussian kernel matrix $K$ and a suitable weight function $v(\cdot)$.
    A positive integer $M$ as the maximum number of iterations and a small real $\varepsilon > 0$.
    Initialize $k = 0,\ \boldsymbol{\alpha}_0,\ b_0$.
Output: $\boldsymbol{\alpha}^*, b^*$;
1: Calculate $V(\xi^{(k)})$ by weight function $v(\cdot)$ and residual $\xi^{(k)}$;
2: Solve the optimization problem Eq. (33) to get $(\boldsymbol{\alpha}_{k+1}, b_{k+1})$ and calculate corresponding residual $\xi^{(k+1)}$;
3: If $k > M$ or $||\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1}||_2^2 < \varepsilon$, stop; Else, go to next step;
4: Let $k = k + 1$ and go to step 1;
```

3.2. Extend IRLS strategy to ELM

In order to widely study the universality of the explicitly iterative reweighted robust methods, we integrate the IRLS with the framework of ELM in this subsection. Ordinary ELM can easily lead to overfitting with the only ERM term and a Tikhonov regularization term is added to improve generalization performance:

$$\min_{\boldsymbol{\beta}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N} ||H\boldsymbol{\beta} - \boldsymbol{Y}||_2^2 \tag{36}$$

$C > 0$ is a parameters defined later, which is used to trade-off the regularization term and the risk term. Similar work has been done in [41] and they focus on the modification of regularization term and the application of various weights. Based on the previous conclusion, the $k+1$th iteration of IRLS-ELM can be rewritten as follows:

$$\begin{aligned} \boldsymbol{\beta}_{k+1} = \arg\min_{\boldsymbol{\beta}} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \frac{C}{2}\sum_{i=1}^{N} v(\xi_i^{(k)})\xi_i^2 \\ s.t. \quad & y_i = \boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta} + \xi_i, \quad i = 1, 2, \cdots, N \end{aligned} \tag{37}$$

where $v(\xi_i^{(k)})$ represents the weight of ith sample based on kth residual $\xi_i^{(k)}$. The output weights $\boldsymbol{\beta}_{k+1}$ of Eq. (37) can be solved by:

$$\boldsymbol{\beta}_{k+1} = \left\{ \begin{array}{ll} H^T(\frac{1}{C} + VHH^T)^{-1}VY, & N < L \\ (\frac{1}{C} + H^TVH)^{-1}H^TVY, & N \geq L \end{array} \right. \tag{38}$$

where H is the hidden layer output matrix and V denotes the weight matrix. Its expression is consistent with Eq. (34), which can be calculated after every iteration. Then the prediction function can be expressed as

$$y = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta}^* \tag{39}$$

where $\boldsymbol{\beta}^*$ represents the final result of the iteration process. Similar to IRLS-SVR, the difference between IRLS-ELM and ELM with regularization term is whether the loss is affected by the weight function. The workflow of IRLS-ELM can be summarized as **Algorithm** 2

3.3. Weight selection strategy

For the explicitly reweighted based robust methods, the assignment strategy of weights is the core issue to be considered. Many pioneers have done a lot of research on this issue [42–44]. In most studies, weight selection

Algorithm 2 Iterative reweighted robust method with ELM (IRLS-ELM)

```
Input:
    Training set $T = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, $\boldsymbol{x}_i \in \mathcal{R}^n$.
    Hidden layer function $g(\boldsymbol{a}_i, b_i, \boldsymbol{x})$, hidden nodes $L$ and a suitable weight function $v(\cdot)$.
    A positive integer $M$ as the maximum number of iterations and a small real $\varepsilon > 0$.
    Initialize $k = 0$, $\boldsymbol{\beta}_0$.
Output: $\boldsymbol{\beta}^*$;
1: Randomly generate $L$ hidden node parameters $(\boldsymbol{a}_i, b_i)$ and calculate the hidden layer output matrix $H$;
2: Calculate $V(\xi^{(k)})$ by weight function and residual $\xi^{(k)}$;
3: Solve the optimization problem Eq. (38) to get $\boldsymbol{\beta}_{k+1}$ and calculate corresponding residual $\xi^{(k+1)}$;
4: If $k > M$ or $||\boldsymbol{\beta}_k - \boldsymbol{\beta}_{k+1}||_2^2 < \varepsilon$, stop; Else, go to next step;
5: Let $k = k + 1$ and go to step 2;
```

strategies are heuristic, but we hope to find a more effective weight selection strategy from the theoretical point of view. A quite detail theoretical analysis of the weight selection strategy is elaborated in [40] and our work is highly inspired by them. Naturally, weight function should satisfy the condition **v1** to **v3** to ensure the convergence of the algorithm. Moreover, we consider that the weight function should give a small weight value for outliers and ordinary weights should be given to the clear samples. For IRLS framework, the following guidelines for gradient function were given in [40], its proof is through two special cases (just consider the empirical risk or the discriminant function is equal to zero). Although the proof does not extend to the general situation, it still has some significance for the selection of our weight selection.

c1 $\psi(x)$ is a measurable, real, odd function.

c2 $\psi(x)$ continuous and differentiable.

c3 $\psi(x)$ bounded.

c4 $\psi(x)$ strictly increasing.

where **c4** can be relaxed as $\psi(x)$ increasing. Inspired by the theoretical requirements of IRLS method, we consider the following sigmoid function $(1/(1+e^{-x}))$ induced gradient function for explicitly reweighted based noise (even heavy-tailed) robust regreesion:

$$\psi_s(x) = \frac{\lambda}{1+e^{-\lambda x}} - \frac{\lambda}{1+e^{0}} = \frac{\lambda}{1+e^{-\lambda x}} - \frac{\lambda}{2} \tag{40}$$

where λ a parameter whose value will be determined later. On basis of $\psi_s(\cdot)$, we can obtain the corresponding weight function as:

$$w_s(x) = \frac{\lambda}{2x + 2x\exp(-\lambda x)} - \frac{\lambda}{4x} \tag{41}$$

It can be easily prove that the $\psi_s(\cdot)$ satisfies the **c1** to **c4** conditions and its weight function $v_s(\cdot)$ also satisfies the **v1-v3** conditions, simultaneously. As mentioned in [40], for IRLS based KBR, the selection of kernel function is also important. A bounded kernel function is quite recommended, so the Gaussian kernel function is selected in the proposed IRLS-SVR:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2}{2\sigma^2}) \tag{42}$$

There are two main reasons why we choose this weight function. Firstly, the proposed weight selection strategy can make the IRLS algorithm converge, and we will give the proof of convergence next. In addition, $v_s(\cdot)$ weight based IRLS algorithm has a certain degree of robustness both theoretically and experimentally, and we will focus on the analysis of robustness in the next subsection.

Now, we discuss the convergence of IRLS algorithm with the proposed weight selection strategy. Convergence of iterative algorithms is usually essential. According to [40], since the sigmoid function induced weight $v_s(\cdot)$ satisfies the conditions **v1-v3**, for the kth iteratively optimization result f_k, there exists f_∞ such that $f_k \to f_\infty$ as $k \to \infty$. Moreover, the convergence solution f^* can approximate the optimal solution to the following optimization problem

$$\min_{f \in \mathcal{H}} \quad \lambda \|f\|_{\mathcal{H}}^2 + E(L(Y, f(X))) \tag{43}$$

and then we prove that similar conclusions hold when we minimize the empirical risk instead of expected risk. Without losing generality, the proof is based on LS-SVR. We give the $k+1$th iteration of IRLS-SVR again and the bias b is deliberately omitted here for writing convenient.

$$\boldsymbol{w}_{k+1} = \arg\min_{w} \quad \lambda\|\boldsymbol{w}\|_2^2 + \frac{1}{N}\sum_{i=1}^{N} v_s(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))(y_i - \boldsymbol{w}^T\phi(\boldsymbol{x}_i))^2 \tag{44}$$

and note the following implicitly reweighted objective by modifying loss function:

$$R(\boldsymbol{w}_k) = \lambda\|\boldsymbol{w}_k\|_2^2 + \frac{1}{N}\sum_{i=1}^{N} L_s(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i)) \tag{45}$$

where $L_s(\cdot)$ represents the corresponding loss function under the premise of using $v_s(\cdot)$ weight selection strategy and we will give the concrete expression of it later, and now we declare that $L_s(\cdot)$ is a convex, continuous and differentiable loss function. $R \geq 0$ is obviously true. We mainly focus on whether R is strictly decreasing. Before we give the convergence proof, we give a following representation lemma:

Lemma 1. *For the $\boldsymbol{w}$ of the $k+1$th iteration, it holds that*

$$\boldsymbol{w}_{k+1} = \frac{1}{\lambda}\frac{1}{N}\sum_{i=1}^{N}[v_s(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))h_{k+1}(\boldsymbol{x}_i, y_i)\phi(\boldsymbol{x}_i)] \tag{46}$$

with $h_{k+1}(\boldsymbol{x}_i, y_i) = y_i - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i)$.

Proof. For the optimization problem Eq. (44), Fermat lemma gives the necessary conditions for the objective function to be extreme at some point. Due to $\boldsymbol{w}_{k+1}$ is the optimal solution of the kth iteration, we have

$$\frac{\partial}{\partial \boldsymbol{w}_{k+1}}[\lambda\|\boldsymbol{w}_{k+1}\|_2^2 + \frac{1}{N}\sum_{i=1}^{N} v_s(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))(y_i - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i))^2] = 0 \tag{47}$$

expand the derivative we have

$$2\lambda\boldsymbol{w}_{k+1} - \frac{2}{N}\sum_{i=1}^{N} v_s(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))(y_i - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i))\phi(\boldsymbol{x}_i) = 0 \tag{48}$$

transpose and replace $y_i - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i)$ by $h_{k+1}(\boldsymbol{x}_i, y_i)$.□

On the basis of this lemma, we can prove the convergence and approximability of IRLS method. Firstly, we give the proof of convergence. It can be proven that the sequence of iteratively optimization result $\{\boldsymbol{w}_k\}$ converges using this representation.

Theorem 1. *Let $\boldsymbol{w}_0 \in \mathcal{R}^n$ be any initial fit. The weight function $v_s(\cdot)$ of its loss function $L_s(\cdot)$ satisfying **v1-v3**. Thus, it holds that $w_k \to w_\infty$, as $k \to \infty$.*

Proof. Define a real function $U(\cdot)$ satisfies $U'(z) = \psi_s(z) = 2zv_s(z)$ and define $g(z^2) = U(z)$, so we have $2zv_s(z) = (g(z^2))' = g'(z^2) \cdot 2z$. Thus we have $g'(z^2) = v_s(z)$. Because of **v1** and **v3**, it holds that $U'(z) \geq 0$ for $z \geq 0$ and $U'(z) \leq 0$ for $z < 0$. According to **v3** the weight $v_s(\cdot)$ is decreasing, so the function $g(\cdot)$ is concave. Thus the inequality $g(a) - g(b) \leq (a-b)g'(b)$ holds.

$$\begin{aligned}
&R(\boldsymbol{w}_{k+1}) - R(\boldsymbol{w}_k)\\
&= \lambda||\boldsymbol{w}_{k+1}||_2^2 - \lambda||\boldsymbol{w}_k||_2^2 + \frac{1}{N}\sum_{i=1}^{N}[U(y_i - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i)) - U(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))]\\
&\leq \underbrace{\frac{1}{N}\sum_{i=1}^{N}[((y_i - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i))^2 - (y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))^2)g'((y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))^2)]}_{R_1}\\
&+ \underbrace{\lambda||\boldsymbol{w}_{k+1}||_2^2 - \lambda||\boldsymbol{w}_k||_2^2}_{R_2}
\end{aligned} \tag{49}$$

$g'(z^2) = v_s(z)$ and use the formula for the difference of squares, R_1 can be written as:

$$\frac{1}{N}\sum_{i=1}^{N}[v_s(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))(2y_i - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i) - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))(\boldsymbol{w}_k^T\phi(\boldsymbol{x}_i) - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i))] \tag{50}$$

substitute $(2y_i - 2\boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i)) + (\boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i) - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))$ for $(2y_i - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i) - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))$ and replace $y_i - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i)$ with $h_{k+1}(\boldsymbol{x}_i, y_i)$, R_1 can be divided into two parts: R_{11} and R_{12}

$$R_{11} = -\frac{1}{N}\sum_{i=1}^{N}[v_s(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))(\boldsymbol{w}_k^T\phi(\boldsymbol{x}_i) - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i))^2] \tag{51}$$

$$R_{12} = \frac{1}{N}\sum_{i=1}^{N}[v_s(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))(\boldsymbol{w}_k^T\phi(\boldsymbol{x}_i) - \boldsymbol{w}_{k+1}^T\phi(\boldsymbol{x}_i))2h_{k+1}(\boldsymbol{x}_i, y_i)] \tag{52}$$

it can be written as

$$R_{12} = (\boldsymbol{w}_k - \boldsymbol{w}_{k+1})^T \frac{1}{N}\sum_{i=1}^{N}[v_s(y_i - \boldsymbol{w}_k^T\phi(\boldsymbol{x}_i))2h_{k+1}(\boldsymbol{x}_i, y_i)\phi(\boldsymbol{x}_i)] \tag{53}$$

using **Lemma 1** it can be transformed as

$$R_{12} = (\boldsymbol{w}_k - \boldsymbol{w}_{k+1})^T 2\lambda\boldsymbol{w}_{k+1} = -2\lambda||\boldsymbol{w}_{k+1}||_2^2 + 2\lambda\boldsymbol{w}_{k+1}^T\boldsymbol{w}_k \tag{54}$$

thus we have

$$\begin{aligned} &R(\boldsymbol{w}_{k+1}) - R(\boldsymbol{w}_k) = R_{11} + R_{12} + R2 \\ &= R_{11} - \lambda||\boldsymbol{w}_{k+1}||_2^2 + 2\lambda\boldsymbol{w}_{k+1}^T\boldsymbol{w}_k - \lambda||\boldsymbol{w}_k||_2^2 \\ &= R_{11} - \lambda||\boldsymbol{w}_{k+1} - \boldsymbol{w}_k||_2^2 \end{aligned} \tag{55}$$

R_{11} is negative, so **Theorem 1** is proved.□

Then, we discuss the approximability of IRLS algorithm with the proposed weight selection strategy. We are concerned about whether the explicitly reweighted IRLS method can be transformed into the implicitly reweighted pipeline. The answer is yes and we call this property approximability.

Proposition 1. *The solution of LS-SVR with the convex $L_s(\cdot)$ loss can be obtained as the limit of a sequence of IRLS-SVR with the $v_s(\cdot)$ weight selection strategy with arbitrary initial fit.*

Proof. For a sequence $\{\boldsymbol{w}\}$ satisfies the convergence, based on **Lemma 1**, the limit $\boldsymbol{w}_\infty$ satisfies:

$$\boldsymbol{w}_\infty = \frac{1}{\lambda}\frac{1}{N}\sum_{i=1}^{N}[v_s(y_i - \boldsymbol{w}_\infty^T\phi(\boldsymbol{x}_i))(y_i - \boldsymbol{w}_\infty^T\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_i)] \tag{56}$$

A quantitative representation theorem for the optimization problem of Eq. (45) with arbitrary convex loss proposed in [45] as follows

$$\boldsymbol{w} = -\frac{1}{2\lambda}\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\delta}_i\phi(\boldsymbol{x}_i) \tag{57}$$

where $\boldsymbol{\delta}_i = L'(y_i - \boldsymbol{w}^{\boldsymbol{T}}\phi(\boldsymbol{x}_i))$, $L'(\cdot)$ denotes the derivative with respect to the second variable. Due to $L'(x) = \psi(x) = 2xv(x)$, the above formula can be written as

$$
\begin{aligned}
\boldsymbol{w} =& -\frac{1}{2\lambda}\frac{1}{N}\sum_{i=1}^{N}(\psi(y_i - \boldsymbol{w}^T\phi(\boldsymbol{x}_i))(-1)\phi(\boldsymbol{x}_i)) \\
=&\frac{1}{2\lambda}\frac{1}{N}\sum_{i=1}^{N}(v(y_i - \boldsymbol{w}^T\phi(\boldsymbol{x}_i))2(y_i - \boldsymbol{w}^T\phi(\boldsymbol{x}_i))\phi(\boldsymbol{x}_i))
\end{aligned}
\tag{58}
$$

Compare Eq. (58) with Eq. (56), it can be found that the final solution of $v_s(\cdot)$ weight based IRLS-SVR $\boldsymbol{w}_\infty$ satisfies the quantitative representation theorem for Eq. (45) with the $L_s(\cdot)$ loss. Due to the local optimal solution is certainly the global optimal one of convex optimization and $L_s(\cdot)$ is a convex, continuous and differentiable loss, we can draw a conclusion: the solution of LS-SVR with the convex $L_s(\cdot)$ loss can be obtained as the limit of a sequence of IRLS-SVR with the $v_s(\cdot)$ weight selection strategy with arbitrary initial fit.□

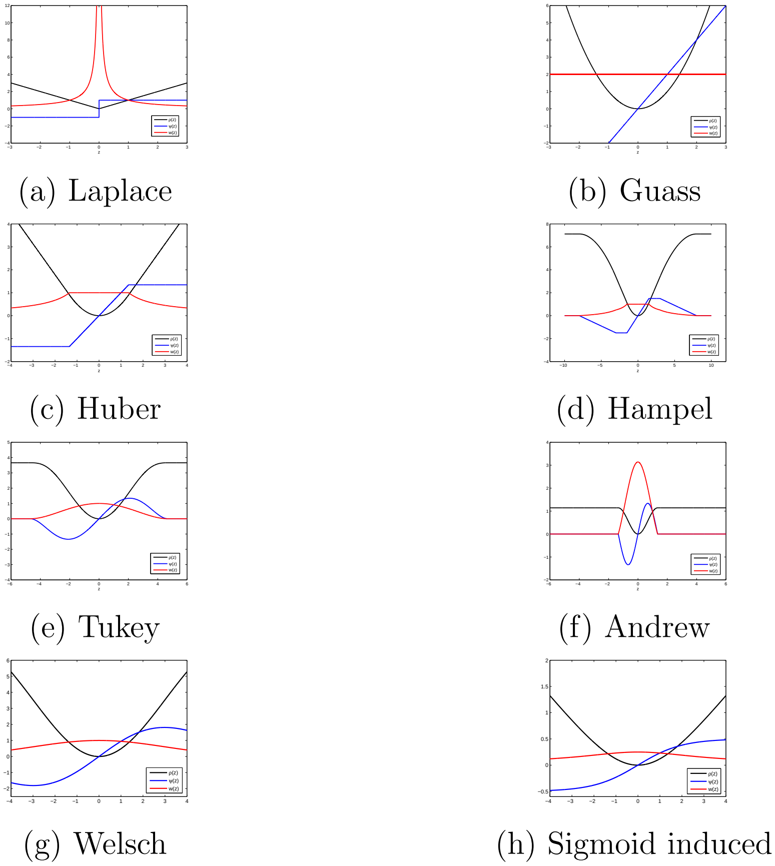
So far we theoretically prove the convergence and approximability of the proposed $v_s(\cdot)$ weight selection strategy based IRLS-KBR. Convergence is the basic condition of the algorithm. Approximability property can better analyze the potential relationship between explicitly reweighted robust method and implicitly reweighted one, and help us discuss the robustness of the proposed method in the next section.

4. Robust analysis

In this section, we will illustrate the robustness of the proposed sigmoid function induced weight selection strategy based explicitly reweighted KBR from two different angles: Theoretical perspective and numerical perspective.

4.1. Theoretical perspective

From a theoretical point of view, by observing the weight selection strategy, we can find that the proposed sigmoid function induced weight $v_s(cdot)$ is a non-negative even function. When the independent variable of $v_s(cdot)$ is greater than 0, the derivative is less than 0, which means that for the samples with large residual, the corresponding weight tends to decrease, and eventually tends to 0. This property is the basis of robustness, and outliers which are difficult to predict will be gradually neglected as the iteration proceeds.



(a) Laplace

(b) Guass

(c) Huber

(d) Hampel

(e) Tukey

(f) Andrew

(g) Welsch

(h) Sigmoid induced

Fig. 1. Comparison of commonly-used loss functions and L_s-loss

Further, based on the analysis in subsection 3.3, We know that the $v_s(\cdot)$ weight selection strategy based robust IRLS-KBR can be equivalent to the implicitly reweighted KBR which uses a exclusive loss function as follows:

$$L_s(x) = ln(1 + e^{\lambda x}) - \frac{\lambda}{2}x + l_0 \tag{59}$$

where $\lambda, l_0 \in \mathcal{R}$, λ is a variable used to change to the amplitude and l_0 is a constant to guarantee function through the origin. Comparison with L_2-loss, L_1-loss, Huber loss and some other commonly used loss functions can be seen in Figure 1 and expression of the gradient functions and weight functions can be seen in Table 1. Because of the equivalence between the two methods proved before, the criterion of robust loss function can be applied to the discrimination of our explicitly reweighted method.

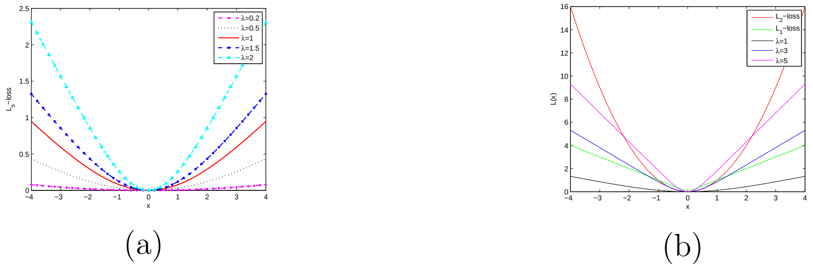


Fig. 2. Form of L_s-loss function under different parameters values.

Remarks. The robustness is analyzed from the implicitly reweighted method by modifying loss function. We list advantages of the L_s-loss as follows to confirm the robustness and superiority of the proposed $v_s(\cdot)$ weight selection:

1. L_s-loss is a convex, continuous and differentiable loss function, due to the smoothness and convexity, it can be optimized efficiently.
2. The gradient function of L_s-loss is a bounded, continuous, differentiable and strictly monotone increasing odd function. It is well known that a bounded gradient function theoretically has good robustness.
3. Theoretically, the variation of L_s-loss, L_1-loss and L_2-loss with large residuals can be compared by:

$$\lim_{x\to\infty}\frac{L_s(x)}{L_1(x)}=\frac{\lambda}{2} \tag{60}$$

$$\lim_{x\to\infty}\frac{L_s(x)}{L_2(x)}=0. \tag{61}$$

   which means: the proposed L_s-loss treat large residual samples similar to L_1-loss (also affected by parameter λ), both are more robust than L_2-loss, cause when $x\to\infty$, $L_s(x)$ and $L_1(x)$ is of the same order and $L_2(x)$ is of higher order.

From Figure 1, gradient function is linear with residuals for L_2-loss, thus L_2-loss changes enormously for large residuals, and the function graph is steep. For other robust losses, a bounded derivative is owned, so using the these losses instead of L_2-loss is more robust. As shown in Figure 2(a), with the increase of parameter λ, the height of L_s-loss will synchronously raise. Theoretically, it can approach any convex loss function. From Figure 2(b), we can scrutinize some of the properties of L_s-loss. When $\lambda=1$ (black solid line), it means that the form of L_s-loss is not adjusted. The shape of the L_s-loss is relatively smooth, and the height is lower than L_1-loss. When $\lambda = 3$ (blue solid line), L_s-loss approximate to L_1-loss, and when $\lambda = 5$ (magenta solid line), L_s-loss is situated between L_1-loss and L_2-loss. The bigger value of λ, the steeper shape and the higher altitude of the L_s-loss.

In addition to the above shallow contrast, it is well known that the derivative of a function at a certain point describes the the function's change rate around this point. That is, the size of the first derivative of a function at a point represents how fast the function changes at this point, and it is often called influence function [46] in robust methods. Obviously, a bounded influence function means that there is a limited change of function caused by noise. Influence function of a estimator T is defined as

$$IF = \lim_{\epsilon \to 0} \frac{T((1-\epsilon)F + \epsilon H) - T(F)}{\epsilon} \tag{62}$$

where F is the main distribution, H is the pollution distribution and ϵ is pollution rate. For regression parameter β of linear regression, above formula can be written as

$$IF = M^{-1} L'(z_y - f(z_x)) z_x \tag{63}$$

where $z = (z_x, z_y)$ is polluted point, f is decision function and $M = \frac{1}{n}\sum_{i=1}^{n} L''(y_i - f(x_i)) x_i^T x_i$. Similar conclusions were presented in [47]

$$IF = S^{-1}(E_F(L'(Y, f(X))\psi(X))) - L'(z_y, f(z_x)) S^{-1}\psi(z_x) \tag{64}$$

where $S(f) = 4f/C + E_F((L''(Y, f(X)) < \psi(X), f > \psi(X)))$. By comparing the above two formulas, it can be seen that the bounded gradient function and bounded kernel such as Guassian kernel function can cause a bounded influence function. According to [40] and Eq. (40), we can easily prove that the gradient function of L_s-loss meets **c1** to **c4** and it can converge to a bounded influence function under special circumstances. The other robust losses in the Table 1 are going against the conditions.

The above analysis can prove the robustness of L_s-loss based KBR, which can also reflect the robustness of the explicitly reweighted IRLS based KBR with the proposed $v_s(\cdot)$ weight on the other hand, because they are equivalent.

4.2. Numerical perspective

Even with the above analysis, it is still not easy to measure the robustness of an algorithm. In this subsection, we use regression curves to evaluate the

Table 1

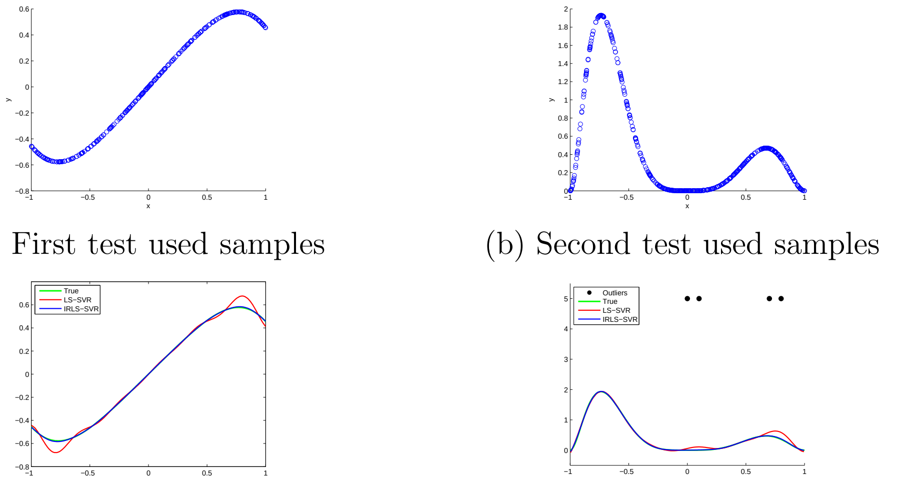
Commonly-used loss functions and corresponding weight functions

Function case	**Gradient function**	**Weight function**
	$\psi(z)$	$w(z)$
Gauss(L_2-loss)	$2z$	1
Laplace(L_1-loss)	$sign(z)$	$\frac{1}{2\lvert z\rvert}$
Huber	$\begin{cases} z, & \lvert z\rvert \le k \\ ksign(z), & \lvert z\rvert > k \end{cases}$	$\begin{cases} \frac{1}{2}, & \lvert z\rvert \le k \\ \frac{k}{2\lvert z\rvert}, & \lvert z\rvert > k \end{cases}$
Hampel	$\begin{cases} z, & \lvert z\rvert \le a \\ asign(z), & a < \lvert z\rvert \le b \\ asign(z)\frac{c-\lvert z\rvert}{c-b}, & b < \lvert z\rvert \le c \\ 0, & \lvert z\rvert > c \end{cases}$	$\begin{cases} \frac{1}{2}, & \lvert z\rvert \le a \\ \frac{a}{2\lvert z\rvert}, & a < \lvert z\rvert \le b \\ \frac{ac-a\lvert z\rvert}{2(c-b)\lvert z\rvert}, & b < \lvert z\rvert \le c \\ 0, & \lvert z\rvert > c \end{cases}$
Tukey	$\begin{cases} z(1-\frac{z^2}{k^2})^2, & \lvert z\rvert \le k \\ 0, & \lvert z\rvert > k \end{cases}$	$\begin{cases} \frac{1}{2}(1-\frac{z^2}{k^2})^2, & \lvert z\rvert \le k \\ 0, & \lvert z\rvert > k \end{cases}$
Andrew	$\begin{cases} ksign(z)sin(\frac{\pi z}{k}), & \lvert z\rvert \le k \\ 0, & \lvert z\rvert > k \end{cases}$	$\begin{cases} \frac{ksin(\frac{\pi z}{k})}{2\lvert z\rvert}, & \lvert z\rvert \le k \\ 0, & \lvert z\rvert > k \end{cases}$
Welsch	$zexp(-\frac{1}{2}(\frac{z}{k})^2)$	$\frac{1}{2}exp(-\frac{1}{2}(\frac{z}{k})^2)$
Sigmoid induced(L_s-loss)	$\frac{\lambda}{1+exp(-\lambda z)} - \frac{\lambda}{2}$	$\frac{\lambda}{2z+2zexp(-\lambda z)} - \frac{\lambda}{4z}$

performance visually. Sensitivity curves can be seen as a finite version of the influence function of the decision function. We generate artificial data points and deliberately add several outliers. For optimal parameters C, λ and RBF kernel parameter γ, we observe the effect of outliers by two different settings and record the changes of the weight of outlier during the iteration. From [40, 45], the sensitivity curve at an additional point $z_i(\boldsymbol{x}_i, y_i)$ can be defined as

$$SC(z_i; f) = \frac{(f(P) - f(P^i))}{1/n} \tag{65}$$

where P is the training set with n samples and $P^i = P \backslash z_i$. $f(P)$ and $f(P^i)$ respectively means the decision made after adding z_i or without z_i. Generally, the additional point z_i may be an errant point so that we can see how it affects the model. Obviously, a smaller SC value means better robustness. The simulated data from $y = sin(z)cos(z^2)$,$z \in [-1, 1]$ and $y = 15(z^2 - 1)^2 z^4 exp(-z)$, $z \in [-1, 1]$ are shown in Figure 3. We deliberately add a few artificial outliers for both simulation: $\boldsymbol{x}_1(-0.8, -5)$, $\boldsymbol{x}_2(0.8, 5)$ for first and $\boldsymbol{x}_1(0, 5)$, $\boldsymbol{x}_2(0.1, 5)$, $\boldsymbol{x}_3(0.7, 5)$ and $\boldsymbol{x}_4(0.8, 5)$ for second. From regression curves, the proposed algorithm is less affected by outliers than LS-SVR. The red solid line (LS-SVR) deviates from green dotted line (Real) and tends to outliers. The magenta solid line (IRLS-SVR) almost coincides with the green dotted line. From the Figure 1 we can see that weights of the proposed $v_s(\cdot)$



(a) First test used samples

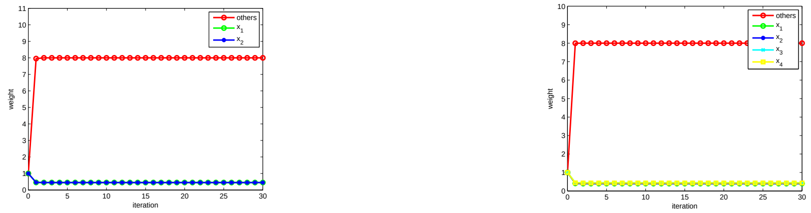
(b) Second test used samples

(c) First test's regression curve

(d) Second test's regression curve

Fig. 3. Numerical simulation for adding outliers

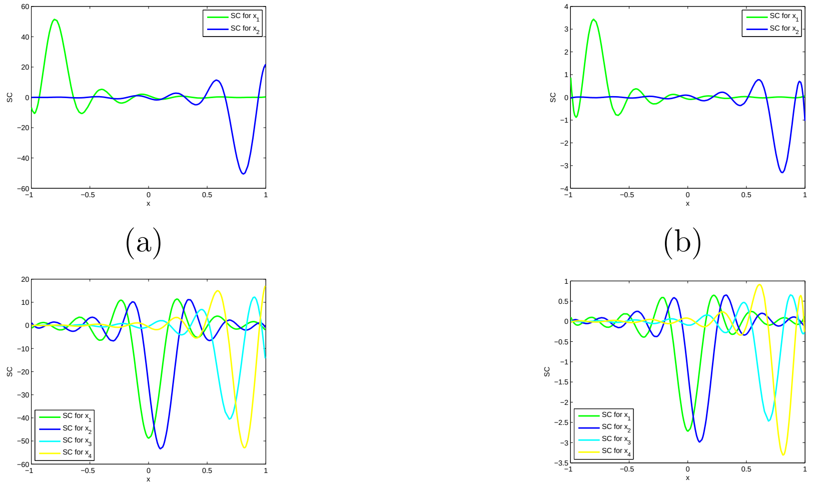
function are tiny for the big residuals instead of equal weight of LS-SVR, which means that the proposed method can reduce the impact of potential outliers on decision making. This is also confirmed by numerical simulation Figure 4, from which we can see that the weight of the outlier is indeed lower than normal in the iteration. For each manual outlier, we plot its sensitivity curve in Figure 5. These figures shows that the sensitivity curve of proposed algorithm significantly better than LS-SVR.



(a)

(b)

Fig. 4. Weight's change with iterating. (a) shows outliers' weights change of the first test and (b) shows the second test's result



(a)

(b)

(c)

(d)

Fig. 5. Sensitivity curve of LS-SVR and IRLS-SVR, (a) and (b) show the sensitivity of LS-SVR and IRLS-SVR to outliers in the first set of test, respectively. (c) (d) is for the second test.

5. Experiments

In order to evaluate the effectiveness of proposed IRLS-SVR and IRLS-ELM, experiments on simulated and benchmark datasets are carried out. We choose LS-SVR [17], weighted LS-SVR (WLS-SVR) [43], ELM and weighted ELM (WELM) [48] as contrast methods. In addition, all the experiments are completed on a personal computer with Intel Core i5-3230M CPU, 4.0 GB RAM, and Windows 7 64 bit operation system in MATLAB R2014a environment. Root mean square error (RMSE) and mean absolute error (MAE) were selected as the evaluation criterions, which are defined as follows

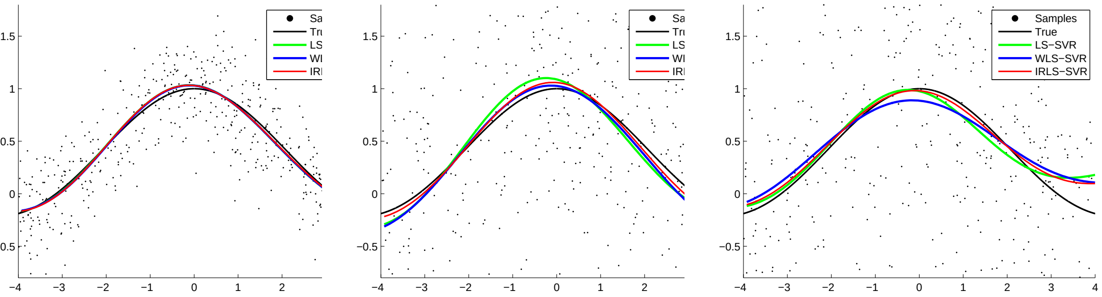
$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{66}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \tag{67}$$

$$MRE = \frac{1}{m}\sum_{i=1}^{m}|\frac{y_i - \hat{y}_i}{y_i}| \tag{68}$$

Table 2
Accuracy comparison on $y = sin(x)/x$ under different noise distributions.

Distribution	**Method**	(C, γ, λ)	**RMSE**	**MAE**
$N(0, 0.3^2)$	LS-SVR	$(1, 0.125, \backslash)$	0.0289 ± 0.0085	0.0239 ± 0.0075
	WLS-SVR	$\backslash$	0.0292 ± 0.0091	0.0241 ± 0.0078
	IRLS-SVR	$(1, 0.125, 4)$	$\mathbf{0.0285 \pm 0.0088}$	$\mathbf{0.0235 \pm 0.0075}$
$L(0, 1)$	LS-SVR	$(0.1, 0.125, \backslash)$	0.1151 ± 0.0405	0.0961 ± 0.0355
	WLS-SVR	$\backslash$	0.1109 ± 0.0381	0.0929 ± 0.0332
	IRLS-SVR	$(0.1, 0.125, 8)$	$\mathbf{0.0893 \pm 0.0320}$	$\mathbf{0.0747 \pm 0.0274}$
$\chi(4)$	LS-SVR	$(0.1, 0.125, \backslash)$	0.1152 ± 0.0428	0.0967 ± 0.0373
	WLS-SVR	$\backslash$	0.1059 ± 0.0368	0.0891 ± 0.0322
	IRLS-SVR	$(0.1, 0.125, 8)$	$\mathbf{0.1017 \pm 0.0387}$	$\mathbf{0.0853 \pm 0.0336}$



(a) Guassian noise (b) Laplacian noise (c) $\chi^2(4)$ noise

Fig. 6. Test with different kinds of noise

where y_i and $\hat{y}_i$ represent the ith true and predicted values. m is the number of samples.

5.1. Simulation on synthetic data

For synthetic data, we generate the data by the widely used *Sinc* function and factitiously add noise as follows: Guassian noise $N(0, 0.3^2)$, Laplacian noise $L(0, 1)$ and χ^2 noise with 4 degree of freedom. We generate 500 training samples and 300 testing samples. Noise is only added in training. For optimal parameters, the process 500 times to reduce randomness.

From the results of Table 2, robustness of the proposed method can be verified. Testing accuracy is improved in all noise conditions. Each method's regression curve is plotted as Figure 6. It can be obviously seen that the IRLS-SVR's regression curve is the most closest to the true *Sinc* curve.

Table 3
SVR comparison on benchmark datasets without outliers.

Dataset	Method	(C, γ, λ)	RMSE	MAE	MRE
Diabetes	LS-SVR	$(2^7, 2^{-1}, \backslash)$	0.1493	0.1260	0.3219
(43 × 2)	WLS-SVR	$(2^6, 2^{-1}, 2^{-3})$	0.1497	0.1257	0.3217
	IRLS-SVR	$(2^8, 2^{-1}, 2^1)$	0.1493	0.1260	0.3216
Pyrim	LS-SVR	$(2^5, 2^{-2}, \backslash)$	0.0795	0.0590	0.0735
(74 × 27)	WLS-SVR	$(2^5, 2^{-2}, \backslash)$	0.0810	0.0615	0.0781
	IRLS-SVR	$(2^8, 2^{-2}, 2^0)$	0.0795	0.0590	0.0735
Triazines	LS-SVR	$(2^5, 2^{-3}, \backslash)$	0.1636	0.1226	0.9146
(186 × 60)	WLS-SVR	$(2^5, 2^{-3}, \backslash)$	0.1730	0.1246	0.9411
	IRLS-SVR	$(2^8, 2^{-3}, 2^0)$	0.1636	0.1226	0.9148
Boston Housing	LS-SVR	$(2^5, 2^{-3}, \backslash)$	0.0870	0.0642	0.2350
(506 × 14)	WLS-SVR	$(2^5, 2^{-3}, \backslash)$	0.0870	0.0629	0.2270
	IRLS-SVR	$(2^2, 2^{-3}, 2^3)$	0.0858	0.0631	0.2302
AutoMPG	LS-SVR	$(2^4, 2^0, \backslash)$	0.0671	0.0505	0.1817
(392 × 7)	WLS-SVR	$(2^4, 2^0, \backslash)$	0.0662	0.0503	0.1774
	IRLS-SVR	$(2^1, 2^0, 2^3)$	0.0668	0.0502	0.1804
Concrete	LS-SVR	$(2^7, 2^{-1}, \backslash)$	0.0998	0.0760	0.2404
(1030 × 8)	WLS-SVR	$(2^7, 2^{-1}, \backslash)$	0.1022	0.0782	0.2518
	IRLS-SVR	$(2^8, 2^{-1}, 2^1)$	0.0998	0.0760	0.2405
Slumptest	LS-SVR	$(2^8, 2^{-2}, \backslash)$	0.0239	0.0190	0.0692
(103 × 10)	WLS-SVR	$(2^8, 2^{-2}, \backslash)$	0.0260	0.0201	0.0698
	IRLS-SVR	$(2^8, 2^{-2}, 2^3)$	0.0194	0.0149	0.0610
MachineCPU	LS-SVR	$(2^6, 2^{-3}, \backslash)$	0.0488	0.0294	0.5912
(209 × 7)	WLS-SVR	$(2^6, 2^{-3}, \backslash)$	0.0533	0.0315	0.6432
	IRLS-SVR	$(2^3, 2^{-3}, 2^3)$	0.0486	0.0293	0.5909

5.2. Simulation on real word benchmark data

In this subsection, we test eight benchmark datasets to illustrate the effectiveness of the proposed methods. The experiments used datasets can be found from UCI machine learning repository [49] and they are widely used for testing machine learning algorithms. All data are scaled such that features locate in the interval $[0, 1]$ before training. The testing accuracies of all experiments are computed using standard 10-fold cross validation. For LS-SVR and its variants, the model parameter C is selected from $\{2^i | i = -4, \cdots, 8\}$ and γ, λ are selected from $\{2^i | i = -3, \cdots, 3\}$ by grid search. For ELMs, the method of selecting parameters C, λ is the same as former, and for the optimal number of hidden nodes L, we choose from $\{m \cdot N | m = 5\%, 10\%, 20\%, \cdots, 50\%\}$,where N is the number of training samples. Noise-free and noisy experiments are carried out simultaneously. For the noisy experiment, we randomly selected 20% of the samples of the training set, multiplied their labels by 10 to simulate outliers. And the final accuracy is obtained by averaging five times 10-fold cross validation experiment. The experimental results are summarized in Table 3 for SVRs and Table 5 for ELMs.

From Table 3 and Table 4, we can see that when training without outliers,

Table 4
SVR comparison on benchmark datasets with outliers.

Dataset	Method	(C, γ, λ)	RMSE	MAE	MRE
Diabetes	LS-SVR	$(2^7, 2^{-1}, \backslash)$	1.0575 ± 0.1329	0.9345 ± 0.1354	2.3504 ± 0.3941
	WLS-SVR	$(2^6, 2^{-1}, 2^{-3})$	0.4451 ± 0.1618	0.3555 ± 0.1355	0.9990 ± 0.4146
	IRLS-SVR	$(2^8, 2^{-1}, 2^1)$	0.4255 ± 0.0860	0.3497 ± 0.0672	0.9942 ± 0.2268
Pyrim	LS-SVR	$(2^5, 2^{-2}, \backslash)$	2.2108 ± 0.2332	1.6877 ± 0.1533	2.4412 ± 0.2104
	WLS-SVR	$(2^5, 2^{-2}, \backslash)$	1.5159 ± 0.2730	1.0721 ± 0.1825	1.5490 ± 0.2498
	IRLS-SVR	$(2^8, 2^{-2}, 2^0)$	1.7703 ± 0.2797	1.3309 ± 0.2447	1.9356 ± 0.3610
Triazines	LS-SVR	$(2^5, 2^{-3}, \backslash)$	1.9586 ± 0.1943	1.4591 ± 0.1460	5.4298 ± 1.6316
	WLS-SVR	$(2^5, 2^{-3}, \backslash)$	1.4187 ± 0.2357	0.8559 ± 0.1321	2.5385 ± 0.8062
	IRLS-SVR	$(2^8, 2^{-3}, 2^0)$	1.6620 ± 0.2024	1.1235 ± 0.1169	3.0085 ± 0.5551
Boston Housing	LS-SVR	$(2^5, 2^{-3}, \backslash)$	0.8322 ± 0.0738	0.7147 ± 0.0740	2.1248 ± 0.2220
	WLS-SVR	$(2^5, 2^{-3}, \backslash)$	0.2133 ± 0.0331	0.1639 ± 0.0254	0.7627 ± 0.1705
	IRLS-SVR	$(2^2, 2^{-3}, 2^3)$	0.1339 ± 0.0130	0.1095 ± 0.0114	0.4614 ± 0.0730
AutoMPG	LS-SVR	$(2^4, 2^0, \backslash)$	0.9354 ± 0.0393	0.7451 ± 0.0525	2.1356 ± 0.1314
	WLS-SVR	$(2^4, 2^0, \backslash)$	0.2842 ± 0.0473	0.1946 ± 0.0275	0.8632 ± 0.0983
	IRLS-SVR	$(2^1, 2^0, 2^3)$	0.1421 ± 0.0100	0.1092 ± 0.0086	0.4560 ± 0.0224
Concrete	LS-SVR	$(2^7, 2^{-1}, \backslash)$	1.1424 ± 0.1852	0.9135 ± 0.1595	2.5063 ± 0.4019
	WLS-SVR	$(2^7, 2^{-1}, \backslash)$	0.3400 ± 0.0452	0.2269 ± 0.0271	0.8807 ± 0.1089
	IRLS-SVR	$(2^8, 2^{-1}, 2^1)$	0.5072 ± 0.0253	0.3845 ± 0.0189	1.2530 ± 0.1112
Slumptest	LS-SVR	$(2^8, 2^{-2}, \backslash)$	1.6551 ± 0.2270	1.3237 ± 0.1675	4.3510 ± 0.9127
	WLS-SVR	$(2^8, 2^{-2}, \backslash)$	1.2711 ± 0.3077	1.0014 ± 0.2654	3.1439 ± 0.5351
	IRLS-SVR	$(2^8, 2^{-2}, 2^3)$	1.3975 ± 0.1901	1.1127 ± 0.1646	3.5497 ± 0.5067
MachineCPU	LS-SVR	$(2^6, 2^{-3}, \backslash)$	0.2697 ± 0.0499	0.1862 ± 0.0315	3.4087 ± 0.5773
	WLS-SVR	$(2^6, 2^{-3}, \backslash)$	0.0872 ± 0.0048	0.0613 ± 0.0044	1.6631 ± 0.0629
	IRLS-SVR	$(2^3, 2^{-3}, 2^3)$	0.0851 ± 0.0142	0.0643 ± 0.0090	1.8202 ± 0.1500

the accuracy of the proposed algorithm is comparable to that of LS-SVR. Both methods are slightly better that WLS-SVR. When artificially adding outliers to the training set, LS-SVR results are unsatisfactory in all datasets, which also reflects its sensitivity to outliers. The difference between the proposed IRLS-SVR and the traditional WLS-SVR is the weight function and the frequency of weighted operation. These two points results in a difference in their robustness. Both methods achieved superior performance relative to LS-SVR.

The Table 5 and Table 6 show the comparison of the results based on the ELM algorithm. The results are similar to those based on SVR. When no artificially add outliers, the generalization performance of the three methods is similar, and the proposed method had no obvious advantage. When the training samples are added to the outliers, the proposed IRLS-ELM exhibit comparable or better generalization performance on the remaining data sets except the Pyrim and Concrete datasets. The Triazines, Boston Housing, and AutoMPG datasets have greatly improved accuracy. Since our method does not filter the optimal parameters again under noisy conditions, but uses the optimal parameters of noise-free experiments, it is acceptable to obtain such results.

Table 5
ELM comparison on benchmark datasets without outliers.

Dataset	Method	(C, L, λ)	RMSE	MAE	MRE
Diabetes	ELM	$(2^{6}, 8, \backslash)$	0.1620 ± 0.0129	0.1346 ± 0.0091	0.3618 ± 0.0355
	W-ELM	$(2^{6}, 8, \backslash)$	0.1558 ± 0.0041	0.1306 ± 0.0020	0.3309 ± 0.0135
	IRLS-ELM	$(2^{-4}, 8, 2^{1})$	0.1770 ± 0.0185	0.1444 ± 0.0095	0.3762 ± 0.0177
Pyrim	ELM	$(2^{6}, 14, \backslash)$	0.1357 ± 0.0150	0.0984 ± 0.0092	0.1293 ± 0.0175
	W-ELM	$(2^{6}, 14, \backslash)$	0.1179 ± 0.0084	0.0906 ± 0.0067	0.1218 ± 0.0109
	IRLS-ELM	$(2^{-1}, 22, 2^{0})$	0.1431 ± 0.0167	0.1028 ± 0.0126	0.1357 ± 0.0154
Triazines	ELM	$(2^{6}, 18, \backslash)$	0.1887 ± 0.0071	0.1440 ± 0.0044	1.0280 ± 0.0597
	W-ELM	$(2^{6}, 18, \backslash)$	0.1871 ± 0.0052	0.1379 ± 0.0045	1.0370 ± 0.1340
	IRLS-ELM	$(2^{1}, 18, 2^{2})$	0.1867 ± 0.0058	0.1410 ± 0.0050	0.9628 ± 0.0615
Boston Housing	ELM	$(2^{8}, 25, \backslash)$	0.1114 ± 0.0053	0.0832 ± 0.0036	0.3135 ± 0.0126
	W-ELM	$(2^{8}, 25, \backslash)$	0.1125 ± 0.0023	0.0843 ± 0.0019	0.3349 ± 0.0109
	IRLS-ELM	$(2^{0}, 25, 2^{3})$	0.1077 ± 0.0010	0.0810 ± 0.0016	0.2973 ± 0.0169
AutoMPG	ELM	$(2^{3}, 39, \backslash)$	0.0743 ± 0.0021	0.0565 ± 0.0014	0.2039 ± 0.0138
	W-ELM	$(2^{3}, 39, \backslash)$	0.0753 ± 0.0019	0.0573 ± 0.0013	0.2160 ± 0.0129
	IRLS-ELM	$(2^{7}, 39, 2^{3})$	0.0760 ± 0.0030	0.0578 ± 0.0024	0.2141 ± 0.0230
Concrete	ELM	$(2^{-1}, 51, \backslash)$	0.1265 ± 0.0040	0.0979 ± 0.0020	0.3209 ± 0.0143
	W-ELM	$(2^{-1}, 51, \backslash)$	0.1242 ± 0.0067	0.0952 ± 0.0053	0.3038 ± 0.0097
	IRLS-ELM	$(2^{5}, 51, 2^{-3})$	0.1249 ± 0.0073	0.0956 ± 0.0042	0.3204 ± 0.0144
Slumptest	ELM	$(2^{-1}, 51, \backslash)$	0.0598 ± 0.0084	0.0475 ± 0.0077	0.1804 ± 0.0371
	W-ELM	$(2^{-1}, 51, \backslash)$	0.0634 ± 0.0195	0.0481 ± 0.0128	0.1757 ± 0.0523
	IRLS-ELM	$(2^{-4}, 51, 2^{1})$	0.0569 ± 0.0122	0.0437 ± 0.0084	0.1473 ± 0.0318
MachineCPU	ELM	$(2^{-1}, 10, \backslash)$	0.0633 ± 0.0049	0.0395 ± 0.0023	0.9238 ± 0.0873
	W-ELM	$(2^{-1}, 10, \backslash)$	0.0637 ± 0.0094	0.0375 ± 0.0043	0.8256 ± 0.1219
	IRLS-ELM	$(2^{3}, 10, 2^{2})$	0.0612 ± 0.0049	0.0385 ± 0.0034	0.9386 ± 0.0761

Table 6
ELM comparison on benchmark datasets with outliers.

Dataset	Method	(C, L, λ)	RMSE	MAE	MRE
Diabetes	ELM	$(2^{6}, 8, \backslash)$	1.1969 ± 0.1523	1.0704 ± 0.1222	2.6609 ± 0.5126
	W-ELM	$(2^{6}, 8, \backslash)$	0.5886 ± 0.0787	0.4793 ± 0.0693	1.3178 ± 0.2761
	IRLS-ELM	$(2^{-4}, 8, 2^{1})$	0.4753 ± 0.0861	0.3955 ± 0.0725	1.1265 ± 0.1497
Pyrim	ELM	$(2^{6}, 14, \backslash)$	1.6947 ± 0.1604	1.4153 ± 0.1253	2.0716 ± 0.1573
	W-ELM	$(2^{6}, 14, \backslash)$	1.2127 ± 0.1898	0.9667 ± 0.1281	1.4319 ± 0.2113
	IRLS-ELM	$(2^{-1}, 22, 2^{0})$	1.7292 ± 0.2647	1.3349 ± 0.2043	1.9657 ± 0.3377
Triazines	ELM	$(2^{6}, 18, \backslash)$	1.5228 ± 0.1205	1.3004 ± 0.0858	4.1854 ± 0.9463
	W-ELM	$(2^{6}, 18, \backslash)$	0.6436 ± 0.1223	0.4613 ± 0.0776	2.0766 ± 0.5883
	IRLS-ELM	$(2^{1}, 18, 2^{2})$	0.2984 ± 0.0083	0.2259 ± 0.0050	1.4069 ± 0.2056
Boston Housing	ELM	$(2^{8}, 25, \backslash)$	0.9395 ± 0.0763	0.8181 ± 0.0765	2.6935 ± 0.2944
	W-ELM	$(2^{8}, 25, \backslash)$	0.2830 ± 0.0519	0.2165 ± 0.0426	0.9177 ± 0.2181
	IRLS-ELM	$(2^{0}, 25, 2^{3})$	0.1513 ± 0.0086	0.1223 ± 0.0066	0.5047 ± 0.0507
AutoMPG	ELM	$(2^{3}, 39, \backslash)$	1.0052 ± 0.0421	0.8109 ± 0.0422	2.6366 ± 0.2897
	W-ELM	$(2^{3}, 39, \backslash)$	0.3833 ± 0.0288	0.2801 ± 0.0192	1.2204 ± 0.2182
	IRLS-ELM	$(2^{7}, 39, 2^{3})$	0.1736 ± 0.0224	0.1317 ± 0.0133	0.5436 ± 0.0345
Concrete	ELM	$(2^{-1}, 51, \backslash)$	1.0682 ± 0.1414	0.8603 ± 0.0799	2.4029 ± 0.1653
	W-ELM	$(2^{-1}, 51, \backslash)$	0.3367 ± 0.0514	0.2433 ± 0.0264	0.9509 ± 0.0605
	IRLS-ELM	$(2^{5}, 51, 2^{-3})$	1.0288 ± 0.1064	0.8423 ± 0.0848	2.4539 ± 0.1701
Slumptest	ELM	$(2^{-1}, 51, \backslash)$	2.5533 ± 0.2901	2.0344 ± 0.2063	8.1882 ± 1.3991
	W-ELM	$(2^{-1}, 51, \backslash)$	2.8802 ± 0.5165	2.2698 ± 0.4157	7.9072 ± 1.6204
	IRLS-ELM	$(2^{-4}, 51, 2^{1})$	2.5214 ± 0.1040	1.9153 ± 0.0608	7.0224 ± 0.5167
MachineCPU	ELM	$(2^{-1}, 10, \backslash)$	0.3381 ± 0.0633	0.2233 ± 0.0229	5.8377 ± 1.3531
	W-ELM	$(2^{-1}, 10, \backslash)$	0.1305 ± 0.0183	0.0866 ± 0.0062	2.0914 ± 0.2821
	IRLS-ELM	$(2^{3}, 10, 2^{2})$	0.1262 ± 0.0176	0.0959 ± 0.0090	2.8142 ± 0.4807

Next, we use two large-scale datasets, Abalone and Winequality, to test the accuracy of the proposed methods. We use grid research and cross validation to filter the optimal parameters. Using the optimal parameters to do five times 10-fold cross validation and record the error of each fold of the experiment. A total of fifty experiment results are summarized in the Table 7 and Table 8. Meanwhile, Figure 7 shows boxplots comparison of fifty experiments' RMSE results. It can be seen from the above that the proposed method effectively reduces the error of LS-SVR and ELM in the presence of noise. Compared with the weighted methods, the proposed reweighted methods also have certain improvements.

Table 7
Comparison on Abalone(4177×7) dataset with outliers.

Method	RMSE	MAE	MRE
LS-SVR	0.5829 ± 0.0739	0.5736 ± 0.0755	2.0557 ± 0.4048
WLS-SVR	0.1299 ± 0.0196	0.1167 ± 0.0255	0.4435 ± 0.1357
IRLS-SVR	0.0958 ± 0.0036	0.0804 ± 0.0058	0.2936 ± 0.0572
ELM	0.6417 ± 0.0908	0.5781 ± 0.0365	1.9428 ± 0.2492
W-ELM	0.1345 ± 0.0600	0.0768 ± 0.0176	0.2524 ± 0.0348
IRLS-ELM	0.1172 ± 0.0647	0.0828 ± 0.0068	0.2992 ± 0.0567

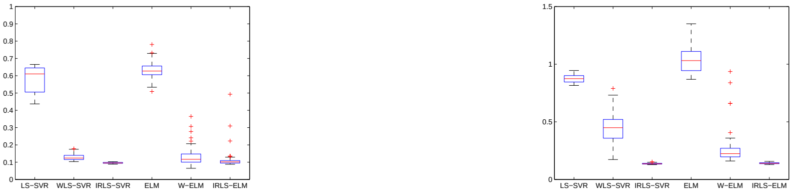
Table 8
Comparison on Winequality(4898×11) dataset with outliers.

Method	RMSE	MAE	MRE
LS-SVR	0.8750 ± 0.0344	0.8621 ± 0.0347	2.0461 ± 0.0980
WLS-SVR	0.4503 ± 0.1514	0.4271 ± 0.1568	1.0906 ± 0.3841
IRLS-SVR	0.1377 ± 0.0064	0.1101 ± 0.0063	0.2889 ± 0.0289
ELM	1.0481 ± 0.1553	0.9003 ± 0.0754	2.0788 ± 0.1408
W-ELM	0.2784 ± 0.1601	0.1532 ± 0.0215	0.3756 ± 0.0561
IRLS-ELM	0.1409 ± 0.0067	0.1122 ± 0.0064	0.2929 ± 0.0302

5.3. Effect of parameters

In this subsection, in the case of with noise, we conduct the experiments to reveal the influence of parameters on test error RMSE. All experiments are conducted in the AutoMPG dataset. For parameters of IRLS-SVR, results are shown in Figure 8 as a three-dimensional bar chart. And the similar results of IRLS-ELM can be seen from Figure 9.

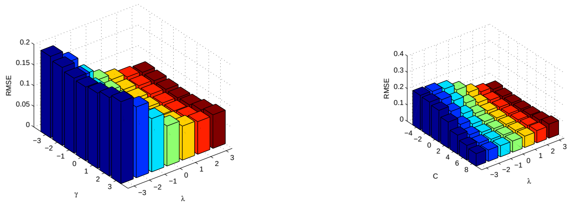
Figure 8(a)-(c) shows the experimental results of the proposed IRLS-SVR. It can be seen from the figure that for IRLS-SVR, each parameter



(a) Comparison on Abalone

(b) Comparison on Winequality

Fig. 7. Accuary Comparison

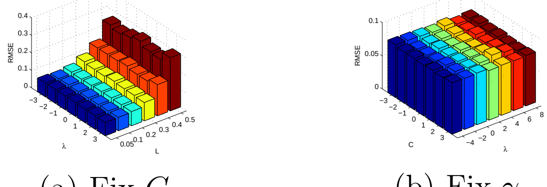


(a) Fix C

(b) Fix γ

(c) Fix τ

Fig. 8. Parameters' influence diagram of the proposed IRLS-SVR



(a) Fix C

(b) Fix γ

(c) Fix τ

Fig. 9. Parameters' influence diagram of the proposed IRLS-ELM

has a certain effect on the model. When C is fixed, the change of kernel parameter γ has little effect on the model, while a larger λ will make the model performance better. When γ is fixed, the larger C and λ are more efficient. When λ is fixed, the other two parameters have little effect on the model. The medium size C and the smaller γ are slightly better. Summarizing the information of these three graphs: λ should be larger and γ should be smaller, which will make the model better and the C parameter has no obvious rules. Figure 9(a)-(c) shows the experimental results of the proposed IRLS-ELM model. It is concluded from the diagram that the parameter L has a greater impact on the performance of the model and the remaining parameters have little effect. Specifically, when C is fixed, the change in λ has little effect on accuracy, and a smaller L will significantly enhance the performance of the model. Similarly, the same happens for fixing optimal λ. Figure 9(b) also shows that the parameters C and λ have little effect on the model. Therefore, when selecting parameters, we should focus more efforts on searching for a good number of hidden layer nodes L. The number of hidden layer nodes L is not as large as possible, while the number of hidden layer nodes slows down the calculation speed, may not improve the accuracy of the algorithm.

6. Conclusion

Based on the theoretical analysis of the weighted robust methods, we research on the explicitly reweighted method, and combine it with two mature KBR algorithms (LS-SVR and ELM) to structure a iteratively reweighted regression framework. Further, in order to solve the problem of how to assign weights, a novel sigmoid function induced weight selection strategy is proposed and the corresponding weight function ($v_s(\cdot)$), gradient function ($\psi_s(\cdot)$) and loss function ($L_s(\cdot)$) are carefully analyzed. On basis of the proposed weight selection strategy, two explicitly iterative reweighted algorithms (IRLS-SVR and IRLS-ELM) with $v_s(\cdot)$ weight function are proposed. We not only theoretically prove the convergence of the algorithm, but also prove that this $v_s(\cdot)$ weight based explicitly reweighted method is equivalent to the L_s-loss based implicitly reweighted robust method. Moreover, the robustness of the proposed method is analyzed theoretically and numerically. Thanks to multiple reweighted operation, the proposed methods achieve good results in the experiments under the noise interference. The effect of hyperparameters on the model is also discussed. Nevertheless, the proposed models have one major shortcoming: more model parameters, which will lead to the increase of training time. How to effectively select the optimal parameters and fast matrix inverse method are our research directions in the future.

Acknowledgements
This work is supported by National Nature Science Foundation of China(11471010).

References

[1] G. E. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets, Neural Computation 18 (7) (2006) 1527–1554. doi:10.1162/neco.2006.18.7.1527.

[2] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Computation 1 (4) (1989) 541–551. doi:10.1162/neco.1989.1.4.541.

[3] J.-Y. Audibert, O. Catoni, Robust linear least squares regression, Annals of Statistics 39 (5) (2011) 2766–2794. doi:10.1214/11-AOS918.

[4] P. Bartlett, S. Mendelson, Empirical minimization, Probability Theory and Related Fields 135 (3) (2006) 311–334. doi:10.1007/s00440-005-0462-3.

[5] F. Fama, D. MacBeth, L. D. Jackel, Risk, Return, and Equilibrium: Empirical tests, Journal of Political Economy 81 (3) (1973) 607–636. doi:10.1086/260061.

[6] O. Catoni, Challenging the empirical mean and empirical variance: a deviation study, Annales de l'Institut Henri Poincaré, Probabilités et Statistiques 48 (4) (2010) 1148–1185. doi:10.1214/11-AIHP454.

[7] P. J. Huber, Robust Statistics, Springer New York, 2014. doi:10.1007/978-3-642-04898-2_594.

[8] P. J. Huber, Robust estimation of a location parameter, Annals of Mathematical Statistics 35 (1) (1964) 73–101. doi:10.1214/aoms/1177703732.

[9] L. Zhang, Z. H. Zhou, ℓ_1-regression with heavy-tailed distributions, in: Advances in Neural Information Processing Systems, 2018.

[10] O. Karal, Maximum likelihood optimal and robust support vector regression with lncosh loss function, Neural Networks 94 (10) (2017) 1–12. doi:10.1016/j.neunet.2017.06.008.

[11] Q. Yao, H. Tong, Asymmetric least squares regression estimation: A nonparametric approach, Journal of Nonparametric Statistics 6 (4) (2007) 273–292. doi:10.1080/10485259608832675].

[12] C. Brownlees, E. Joly, G. Lugosi, Empirical risk minimization for heavy-tailed losses, Annals of Statistics 43 (6) (2015) 2507–2536. doi:10.1214/15-AOS1350.

[13] G. Lugosi, S. Mendelson, Risk minimization by median-of-means tournaments, arXiv preprint arXiv:1608.00757.

[14] L. Yang, H. Dong, Support vector machine with truncated pinball loss and its application in pattern recognition, Chemometrics and Intelligent Laboratory Systems 177 (6) (2018) 89–99. doi:10.1016/j.chemolab.2018.04.003.

[15] Z. Ren, L. Yang, Correntropy-based robust extreme learning machine for classification, Neurocomputing 313 (11) (2018) 74–84. doi:10.1016/j.neucom.2018.05.100.

[16] M. J. Holland, K. Ikeda, Robust regression using biased objectives, Machine Learning 106 (4) (2017) 1–37. doi:10.1007/s10994-017-5653-5.

[17] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, J. Vandewalle, Least squares support vector machines, International Journal of Circuit Theory and Applications 27 (6) (2002) 605–615. doi:10.1002/(SICI)1097-007X(199911/12)27:6<605::AID-CTA86>3.0.CO;2-Z.

[18] G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (1) (2006) 489–501. doi:10.1016/j.neucom.2005.12.126.

[19] C. Chen, C. Yan, Y. Li, A robust weighted least squares support vector regression based on least trimmed squares, Neurocomputing 168 (2015) 941–946. doi:10.1016/j.neucom.2015.05.031.

[20] C. Chen, Y. Li, C. Yan, J. Guo, G. Liu, Least absolute deviation-based robust support vector regression, Knowledge-Based Systems (2017) S0950705117302824doi:10.1016/j.knosys.2017.06.009.

[21] O. L. Mangasarian, D. R. Musicant, Robust linear and support vector regression, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (9) (2002) 950–955. doi:10.1109/34.877518.

[22] A. Christmann, I. Steinwart, How svms can estimate quantiles and the median, in: Advances in Neural Information Processing Systems, Vol. 20, 2007, pp. 305–312.

[23] Z. Kai, M. Luo, Outlier-robust extreme learning machine for regression problems, Neurocomputing 151 (2015) 1519–1527. doi:10.1016/j.neucom.2014.09.022.

[24] O. Catoni, High confidence estimates of the mean of heavy-tailed real random variables, arXiv preprint arXiv:0909.5366.

[25] K. Wang, Z. Ping, Robust non-convex least squares loss function for regression with outliers, Knowledge-Based Systems 71 (2014) 290–302. doi:10.1016/j.knosys.2014.08.003.

[26] Y. P. Zhao, J. G. Sun, Robust truncated support vector regression, Expert Systems with Applications 37 (7) (2010) 5126–5133. doi:10.1016/j.eswa.2009.12.082.

[27] D. Pham Dinh, H. A. Le Thi, F. Akoa, Combining DCA (DC Algorithms) and interior point techniques for large-scale nonconvex quadratic programming, Optimization Methods and Software 23 (4) (2008) 609–629. doi:10.1080/10556780802263990.

[28] L. Yang, Y. Qian, A sparse logistic regression framework by difference of convex functions programming, Applied Intelligence 45 (2) (2016) 241–254. doi:10.1007/s10489-016-0758-2.

[29] Yuille, A. L., CCCP algorithms to minimize the Bethe and Kikuchi free energies : Convergent alternatives to belief propagation, Neural Computation 14 (7) (2002) 1691–1722. doi:10.1162/08997660260028674.

[30] Y. Zhang, Y. Sun, R. He, Robust subspace clustering via half-quadratic minimization, in: IEEE International Conference on Computer Vision, 2013.

[31] R. He, W. Zheng, T. Tan, Z. Sun, Half-quadratic-based iterative minimization for robust sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2) (2014) 261. `doi:10.1109/TPAMI.2013.102`.

[32] Y. Feng, Y. Yang, X. Huang, S. Mehrkanoon, J. A. K. Suykens, Robust support vector machines for classification with nonconvex and smooth losses, Neural Computation 28 (6) (2016) 1217–1247. `doi:10.1162/NECO_a_00837`.

[33] C. Li, S. Zhou, Sparse algorithm for robust LSSVM in primal space, Neurocomputing 275 (2017) 2880–2891. `doi:10.1016/j.neucom.2017.10.011`.

[34] G. Xu, B. G. Hu, J. C. Principe, Robust C-loss kernel classifiers, IEEE Transactions on Neural Networks and Learning Systems 29 (3) (2016) 510–522. `doi:10.1109/TNNLS.2016.2637351`.

[35] E. Hazan, K. Levy, S. Shalev-Shwartz, Beyond convexity: Stochastic quasi-convex optimization, in: Advances in Neural Information Processing Systems, Vol. 28, 2015, pp. 1594–1602.

[36] M. J. Lai, Y. Xu, W. Yin, Improved iteratively reweighted least squares for unconstrained smoothed ℓq minimization, Siam Journal on Numerical Analysis 51 (2) (2013) 927–957. `doi:10.1137/110840364`.

[37] P. J. Green, Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives, Journal of the Royal Statistical Society 46 (2) (1984) 149–192. `doi:doi:10.2307/2981697`.

[38] R. Von Borries, C. J. Miosso, Compressive sensing reconstruction with prior information by iteratively reweighted least-squares, IEEE Transactions on Signal Processing 57 (6) (2009) 2424–2431. `doi:10.1109/TSP.2009.2016889`.

[39] H. Peng, Y. Fan, A general framework for sparsity regularized feature selection via iteratively reweighted least square minimization, in: AAAI Conference on Artificial Intelligence, 2017.

[40] M. Debruyne, A. Christmann, M. Hubert, J. A. K. Suykens, Robustness of reweighted least squares kernel based regression, Journal of Multivariate Analysis 101 (2010) 447–463. doi:10.1016/j.jmva.2009.09.007.

[41] C. Kai, L. Qi, L. Yao, D. Yong, Robust regularized extreme learning machine for regression using iteratively reweighted least squares, Neurocomputing 230 (2016) 345–358. doi:10.1016/j.neucom.2016.12.029.

[42] Q. Xu, J. Zhang, C. Jiang, X. Huang, Y. He, Weighted quantile regression via support vector machine, Expert Systems with Applications 42 (2015) 5441–5451. doi:10.1016/j.eswa.2015.03.003.

[43] J. A. K. Suykens, J. D. Brabanter, L. Lukas, J. Vandewalle, Weighted least squares support vector machines : Robustness and sparse approximation, Neurocomputing 48 (10) (2002) 85–105. doi:10.1016/s0925-2312(01)00644-0.

[44] B. Yang, Q. Shao, L. Pan, W. Li, A study on regularized weighted least square support vector classifier, Pattern Recognition Letters 108 (6) (2018) 48–55. doi:10.1016/j.patrec.2018.03.002.

[45] A. Christmann, I. Steinwart, Support Vector Machines, Springer New York, 2008. doi:10.1007/978-0-387-77242-4.

[46] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, W. A. Stahel, Robust Statistics: The Approach Based on Influence Functions, Wiley New York, 1986. doi:10.2307/1269782.

[47] A. Christmann, I. Steinwart, Consistency and robustness of kernel based regression, Bernoulli 13 (3) (2007) 799–819. doi:10.3150/07-BEJ5102.

[48] W. Deng, Q. Zheng, L. Chen, Regularized extreme learning machine, in: IEEE Symposium on Computational Intelligence and Data Mining, 2009, pp. 389–395. doi:10.1109/CIDM.2009.4938676.

[49] D. Dua, E. Karra Taniskidou, UCI machine learning repository (http://archive.ics.uci.edu/ml), University of California, Irvine, School of Information and Computer Sciences (2017).