SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2022, Vol. 65 222203:1–222203:16
https://doi.org/10.1007/s11432-022-3560-8

Broad learning system based on the quantized minimum error entropy criterion

Simin ZHANG[1], Zhulin LIU[1*] & C. L. Philip CHEN[1,2]

[1]*School of Computer Science and Engineering, South China University of Technology, Guangzhou* 510641*, China;*
[2]*Pazhou Lab, Guangzhou* 510335*, China*

Received 4 March 2022/Revised 20 June 2022/Accepted 7 July 2022/Published online 17 November 2022

Abstract The broad learning system (BLS) based on the minimum mean square error (MMSE) criterion can achieve outstanding performance without spending too much time in various machine learning tasks. However, when data are polluted by non-Gaussian noise, the stability of BLS may be destroyed because the MMSE criterion is sensitive to outliers. Different from the MMSE criterion, the minimum error entropy (MEE) criterion utilizes the kernel function to capture high-dimensional information and decrease the negative influence of outliers, which will make BLS more discriminative and robust. Nevertheless, the computational complexity of MEE is high due to a double summation of the data size. To solve these issues, this paper proposes a new robust BLS variant based on the quantized minimum error entropy (QMEE) criterion, in which a quantization operation is used to reduce the computational complexity of MEE. The proposed model BLS-QMEE is optimized by the fixed-point iterative method, and a sufficient condition for its convergence is provided. Compared with the standard BLS and other existing robust variants of BLS, BLS-QMEE performs more satisfactorily without consuming too much time. The desirable performance of BLS-QMEE is verified by various experiments on function approximation, several public datasets, and a practical application.

Keywords broad learning system, quantized minimum error entropy, robustness, minimum error entropy, convergence

Citation Zhang S M, Liu Z L, Chen C L P. Broad learning system based on the quantized minimum error entropy criterion. Sci China Inf Sci, 2022, 65(12): 222203, https://doi.org/10.1007/s11432-022-3560-8

1 Introduction

Deep learning algorithms have been successfully applied in computer vision, speech recognition, and other challenging learning tasks [1], showing a strong learning capacity. However, there are multiple processing layers in the computational models of deep learning [2], of which most parameters are optimized by the backpropagation algorithm [3], which means their network structures are generally complicated, and it is time-consuming to learn so many parameters in the training process.

The broad learning system (BLS) [4, 5] is an emerging and efficient discriminative learning algorithm consisting of multiple groups of feature nodes, enhancement nodes, and an output layer. Feature nodes and enhancement nodes are generated randomly; only output weights need to be trained by the ridge regression algorithm [6]. Therefore, the training process can be significantly accelerated. In addition, the network can expand dynamically through the incremental learning algorithm to achieve better performance without spending a lot of time retraining the entire network. Due to the advantages mentioned above, BLS has received extensive attention recently and has been applied in many challenging tasks [7–12].

Recently, many variants of BLS have been proposed for better performance. Some substitute the simple feature mappings in BLS with convolution kernels or wavelet functions for better feature extraction [13, 14]. Some change the network structure of BLS for a better learning capacity [15, 16]. Apart from them, the performance of BLS could also be improved by modifying the design of loss function. The loss function of BLS is based on the minimum mean square error (MMSE) criterion, which is easy to implement

* Corresponding author (email: zhulinlau@gmail.com)

and performs well on the training data without noise or with Gaussian noise [17]. However, collected data may be contaminated by noise or outliers [18]. When it comes to complicated noise, especially outliers, the MMSE criterion may amplify the lousy influence of outliers, resulting in the performance degradation of BLS. Thus, it is necessary to design a more robust loss function for BLS. Noticing this problem, RBLS [19] uses the l_1-norm as the loss function and combines it with several different regularization terms, in which L1RBLS combines l_1-norm with l_1-regularization, and L2RBLS combines l_1-norm with l_2-regularization. To some extent, RBLS enhances the robustness of BLS. However, it is complex to calculate the l_1-norm. More recently, a new BLS based on the maximum correntropy criterion (MCC) has been proposed, named C-BLS [20]. Since correntropy can capture higher-order information of prediction errors than l_2-norm and can alleviate the bad influence of outliers [21], C-BLS further improves the robustness of BLS. Although these BLS variants are good choices when training data are contaminated by non-Gaussian noise, they may not perform well when facing a more complex non-Gaussian environment, such as noises with a multimodel distribution [22].

Minimum error entropy (MEE) [23] is a robust criterion aiming to minimize Renyi's entropy of predicted errors, which has successfully developed various robust learning algorithms [24–29]. As a criterion based on entropy, it can also capture higher-order information about prediction errors. However, the calculation of MEE requires a double summation over all sample data. To solve this problem, a quantized minimum error entropy (QMEE) [30] criterion is proposed to reduce the computational complexity of MEE. With a proper quantization threshold, QMEE can keep almost the same performance as MEE. However, its performance will degrade a lot if the quantization threshold is too large. Therefore, an auxiliary quantization code is introduced into the quantization method in this paper to avoid bad quantization code generation or excessive performance degradation. Based on this improved QMEE criterion, this paper proposes a robust BLS named BLS-QMEE, which performs well in Gaussian or complex non-Gaussian environments without consuming too much training time. The proposed BLS-QMEE uses the fixed-point iteration algorithm to optimize its output weights, whose convergence is ensured by a provided sufficient condition. Several experimental results demonstrate that the BLS-QMEE can obtain more discriminative and robust results in a short training time than existing methods. In summary, the main contributions are as follows.

(1) A robust extension of BLS termed BLS-QMEE is proposed by utilizing the quantized minimum error entropy criterion to optimize output weights in BLS. Because of the inherent superiority of MEE and the quantization, BLS-QMEE can achieve better performance than the standard BLS and other robust BLS variants within a short training time, especially in a non-Gaussian environment.

(2) The fixed-point iteration algorithm is used as the optimization method in BLS-QMEE, which is step-size free and converges fast. A sufficient condition is provided to guarantee the convergence of the fixed-point iterations in BLS-QMEE.

(3) Various experiments on function approximation, public datasets, and a practical application are conducted to evaluate the superiority and robustness of BLS-QMEE.

The remainder of this article is organized as follows. Section 2 is a brief review of the standard BLS. In Section 3, the improved QMEE criterion is introduced firstly. The details of the BLS-QMEE method are presented secondly. The convergence analysis of BLS-QMEE is described finally. In Section 4, various experiments on function approximation, public datasets, and a practical application are presented to demonstrate the performance of the proposed method. Finally, Section 5 concludes this paper.

2 Brief review of broad learning system

The BLS is a novel flat neural network providing an effective and efficient learning algorithm for various regression and classification tasks. Figure 1 illustrates the structure of BLS. The BLS algorithm can be divided into three main steps: firstly, input data are randomly mapped into different groups of feature nodes to generate some random features. Secondly, the random features are transformed randomly into groups of enhancement nodes. At last, both feature nodes and enhancement nodes are concatenated together and input to the output layer. Only output weights need to be trained through the ridge regression approximation algorithm.

Assume the training dataset of a supervised regression task as $\{(\boldsymbol{X}, \boldsymbol{y})\} = \{(\boldsymbol{x}_i^{\mathrm{T}}, y_i) | \boldsymbol{x}_i \in \mathbb{R}^D,\ y_i \in \mathbb{R}, i = 1, 2, \ldots, N\}$, in which N denotes the number of training samples and D indicates the dimension of each input vector. There are n groups of feature mappings and m groups of enhancement mappings in

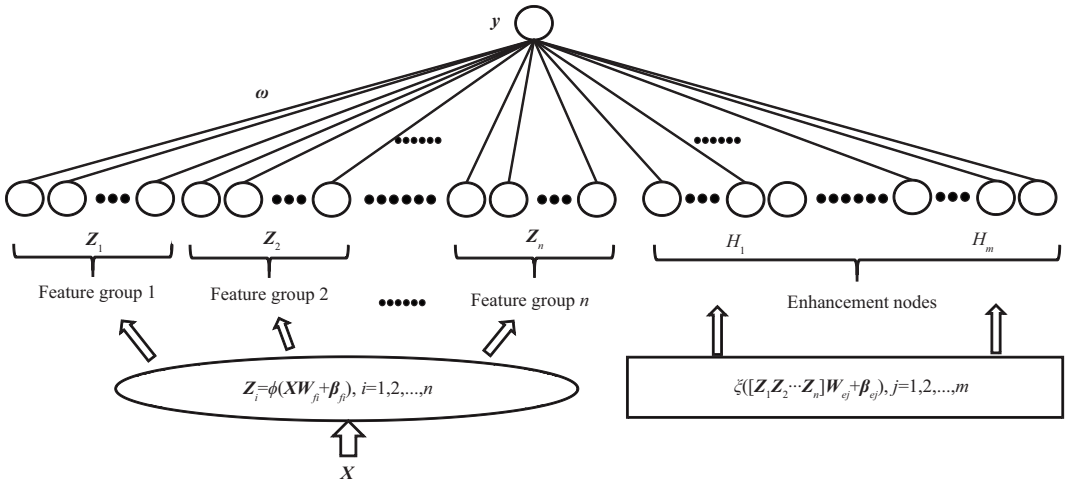

Figure 1 Structure of the broad learning system.

the network. Each group of feature mappings contains p feature nodes, and each group of enhancement mappings contains q enhancement nodes. The output of the ith group of the feature mappings $\boldsymbol{Z}_i$ can be defined as

$$\boldsymbol{Z}_i = \phi(\boldsymbol{X}\boldsymbol{W}_{fi} + \boldsymbol{\beta}_{fi}), \quad i = 1, 2, \ldots, n, \tag{1}$$

where ϕ is a linear mapping function, $\boldsymbol{W}_{fi} \in \mathbb{R}^{D\times p}$ and $\boldsymbol{\beta}_{fi} \in \mathbb{R}^{N\times p}$ are the randomly generated weight matrix and biases for the ith group of the feature mappings, respectively. Let $\boldsymbol{Z}^n \equiv [\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n]$ denote the n groups of feature mappings. Then, send $\boldsymbol{Z}^n$ to enhancement nodes; the output of the jth group of the enhancement mappings $\boldsymbol{H}_j$ can be represented as

$$\boldsymbol{H}_j = \xi(\boldsymbol{Z}^n\boldsymbol{W}_{ej} + \boldsymbol{\beta}_{ej}), \quad j = 1, 2, \ldots, m, \tag{2}$$

where ξ is a nonlinear activation function. $\boldsymbol{W}_{ej} \in \mathbb{R}^{np\times q}$ and $\boldsymbol{\beta}_{ej} \in \mathbb{R}^{N\times q}$ are the randomly generated weight matrix and biases for the jth group of the enhancement mappings, respectively. The m groups of enhancement mappings can be defined as $\boldsymbol{H}^m \equiv [\boldsymbol{H}_1, \ldots, \boldsymbol{H}_m]$. The input matrix $\boldsymbol{A}$ of the output layer can be obtained by concatenating feature nodes and enhancement nodes together:

$$\boldsymbol{A} = [\boldsymbol{Z}^n|\boldsymbol{H}^m]. \tag{3}$$

Feeding $\boldsymbol{A}$ into the output layer, the BLS can be represented by

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{w}, \tag{4}$$

where $\boldsymbol{w}$ denotes output weights. The objective function of BLS is designed based on the MMSE criterion, which is

$$J(\boldsymbol{w}) = \|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2, \tag{5}$$

where λ is a regularization coefficient that approaches 0. The output weights $\boldsymbol{w}$ can be approximated by the ridge regression algorithm:

$$\boldsymbol{w} = \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}, \tag{6}$$

where $\boldsymbol{I}$ denotes an identity matrix. It is clear that only output weights need to be trained in the training process, which makes BLS be trained fast.

3 QMEE-based broad learning system

Although BLS can achieve good generalization performance in a noise-free or Gaussian noise environment, its performance will degrade a lot when dealing with more complicated noise, especially outliers [19]. In this section, this paper first introduces the improved QMEE criterion, a simplified MEE criterion, which performs well in a non-Gaussian environment. Second, a robust BLS based on the improved QMEE algorithm (BLS-QMEE) is developed. Since the output weights of BLS-QMEE are optimized by the fixed-point iteration algorithm and convergence is important for an iterative algorithm, a sufficient condition to ensure the convergence of BLS-QMEE is provided, finally.

3.1 Quantized minimum error entropy criterion

Renyi's α entropy ($\alpha > 0, \alpha \neq 1$) is a mathematical generalization of Shannon's entropy, which has been widely used as a cost function in information theoretic learning (ITL) [23]. The Renyi's α entropy of error e is defined as

$$H_\alpha(e) = \frac{1}{1-\alpha} \log \int p^\alpha(e) \mathrm{d}e = \frac{1}{1-\alpha} \log \left(I_\alpha(e)\right), \tag{7}$$

where $p(e)$ is the probability density function (PDF) of e, and $I_\alpha(e)$ denotes information potential (IP):

$$I_\alpha(e) = \int p^\alpha(e) \mathrm{d}e = \mathrm{E}[p^{\alpha-1}(e)]. \tag{8}$$

In ITL, the parameter α is usually set to 2 for simplicity. Thus, only the case of $\alpha = 2$ is taken into consideration in the rest of this paper. The empirical version of quadratic IP can be defined as

$$\hat{I}_2(e) = \frac{1}{N} \sum_{i=1}^{N} \hat{p}(e_i) = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_\sigma(e_i - e_j), \tag{9}$$

where $\hat{p}(\cdot)$ indicates the Parzen's PDF estimator [31, 32],

$$\hat{p}(e_i) = \frac{1}{N} \sum_{j=1}^{N} G_\sigma(e_i - e_j). \tag{10}$$

$G_\sigma(\cdot)$ is the Gaussian kernel function with bandwidth σ. N is the number of samples.

Under the MEE criterion, the optimal hypothesis can be solved by minimizing $H_2(e)$. Meanwhile, minimizing $H_2(e)$ is equivalent to maximizing $I_2(e)$ because the logarithm function is a monotone increasing function. It is worth noting that $\hat{p}(\cdot)$ can be regarded as an adaptive loss function varying with error samples, making the $\hat{I}_2(e)$ match the error distribution. Therefore, it is potentially beneficial to design the loss function according to the MEE criterion, and Ref. [33] has proved the robustness of MEE theoretically. However, the computational complexity of MEE is $O(N^2)$ because there is a double summation in (9), which means it is time-consuming, especially dealing with large-scale datasets. Therefore, Ref. [30] proposed a quantization method to simplify the computational complexity of MEE, whose main idea is to quantize error samples and reduce the number of inner summations from N to L, $L \ll N$. The quantized MEE is

$$\max\ \hat{I}_2^Q(e) = \max \left(\frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} G_\sigma(e_i - Q[e_j]) \right) = \max \left(\frac{1}{N^2} \sum_{i=1}^{N} \left(\sum_{j=1}^{L} s_j \cdot G_\sigma(e_i - c_j) \right) \right), \tag{11}$$

where $Q[\cdot]$ indicates the quantization operation, L is the number of the quantization codes, c_j indicates the jth quantization code, and s_j is the number of error samples quantized to the code c_j.

Although experimental results in [30] have shown the QMEE with a proper quantization threshold can obtain almost the same performance as MEE without too much time, there is still a deficiency in initializing the quantization codes collection. The first quantization code c_1 is initialized by the first error sample in the QMEE [30], which will result in a bad quantization codes collection when the first sample is an outlier. Besides, the performance of the QMEE [30] will degrade a lot when the quantization threshold is too large. To fix this issue, c_1 is forced to be 0 as an auxiliary in this paper to construct the whole collection of quantization codes further. If there are no error samples quantized to c_1, c_1 will be removed from the final collection of quantization codes. The improved online quantization method is presented in Algorithm 1.

The following can be observed from Algorithm 1:

• As ϵ approaches 0^+, L will approach N and the improved QMEE criterion will turn into the MEE criterion;

• As ϵ approaches $+\infty$, L will approach 1 and the improved QMEE criterion will become the MCC criterion [23]:

$$\max \left(\lim_{\epsilon \to +\infty} \hat{I}_2^Q(e) \right) = \max \left(\frac{1}{N} \sum_{i=1}^{N} G_\sigma(e_i) \right). \tag{12}$$

Algorithm 1 Online quantization

Input: error samples collection: $\{e_i\}_{i=1}^N$.
Output: quantization codes collection: $C = \{c_j\}_{j=1}^L$, the collection including the numbers of error samples quantized by each c_j: $S = \{s_j\}_{j=1}^L$.
Parameters setting: quantization threshold ϵ;
Initialization: set $C = \{c_1 = 0\}$, $S = \{s_1 = 0\}$ and $L = 1$;
for $i = 1, \ldots, N$ **do**
  for $j = 1, \ldots, L$ **do**
    Calculate the distance between e_i and c_j: $d(e_i, c_j) = |e_i - c_j|$;
  end for
  The distance between e_i and C is $d(e_i, C) = d(e_i, c_{j^*})$, where $j^* = \arg\min_{1 \leqslant j \leqslant L} |e_i - c_j|$ and c_{j^*} denotes the code closest to e_i;
  if $d(e_i, C) \leqslant \epsilon$ **then**
    Quantize e_i to c_{j^*}; keep the quantization code collection C unchanged; $s_{j^*} = s_{j^*} + 1$;
  else
    Quantize e_i to itself; update C and S: $C = \{C, e_i\}$, $S = \{S, 1\}$; $L = L + 1$;
  end if
end for
if $s_1 == 0$ **then**
  Remove c_1 from C; delete s_1 from S; $L = L - 1$;
end if

Therefore, the improved QMEE can combine the advantages of the MEE criterion and MCC criterion by adjusting ϵ; it can not only obtain more information than the MCC criterion in a more complicated non-Gaussian environment but need much less time than the MEE criterion.

3.2 BLS based on QMEE

The MMSE criterion applied in BLS relies heavily on the noise-free or Gaussian assumption and may be sensitive to non-Gaussian noise, especially large outliers. Therefore, BLS may not work well when training data are contaminated by complicated noise. To enhance the robustness of BLS, this paper proposes a novel robust BLS, in which the MMSE criterion is substituted by the improved QMEE criterion.

In the proposed BLS-QMEE, the input layer and hidden layer remain the same as BLS, which has been introduced in (1)–(3), while the objective function is different, which is

$$L(\boldsymbol{w}) = -\hat{I}_2^Q(e) + \lambda\|\boldsymbol{w}\|_2^2 = -\frac{1}{N^2}\sum_{i=1}^{N}\left(\sum_{j=1}^{L} s_j \cdot G_\sigma(e_i - c_j)\right) + \lambda\|\boldsymbol{w}\|_2^2, \tag{13}$$

where $e_i = y_i - \boldsymbol{a}_i\boldsymbol{w}$, $\boldsymbol{a}_i$ represents the ith row of $\boldsymbol{A}$, $c_j = y_{c_j} - \boldsymbol{a}_{c_j}\boldsymbol{w}$, and y_{c_j} and $\boldsymbol{a}_{c_j}$ denote y and $\boldsymbol{a}$ corresponding to the quantization code c_j, respectively.

Take the gradient of $L(\boldsymbol{w})$ concerning $\boldsymbol{w}$ to calculate $\partial L(\boldsymbol{w}/\partial\boldsymbol{w})$. By setting $\partial L(\boldsymbol{w})/\partial\boldsymbol{w} = 0$, the solution of $\boldsymbol{w}$ can be deduced,

$$\boldsymbol{w} = \left[\boldsymbol{R} + \lambda'\boldsymbol{I}\right]^{-1}\boldsymbol{P}, \tag{14}$$

where

$$\begin{cases} \boldsymbol{P} = \sum\limits_{i=1}^{N}\left(\sum\limits_{j=1}^{L} s_j \cdot G_\sigma(e_i - c_j)(y_i - y_{c_j})[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}\right), \\ \boldsymbol{R} = \sum\limits_{i=1}^{N}\left(\sum\limits_{j=1}^{L} s_j \cdot G_\sigma(e_i - c_j)[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]\right), \end{cases} \tag{15}$$

and $\lambda' = 2\lambda N^2\sigma^2$. Noting that $\boldsymbol{P}$ and $\boldsymbol{R}$ are related to $\boldsymbol{w}$, Eq. (14) can be represented as

$$\boldsymbol{w} = f(\boldsymbol{w}) = \left[\boldsymbol{R} + \lambda'\boldsymbol{I}\right]^{-1}\boldsymbol{P}, \tag{16}$$

which is a fixed-point equation. Thus, the fixed-point iteration algorithm [34] can be used to find the solution of this fixed-point equation. Choosing a proper $\boldsymbol{w}_0$, calculate $\boldsymbol{w}_t$ by

$$\boldsymbol{w}_t = f(\boldsymbol{w}_{t-1}), \tag{17}$$

where $\boldsymbol{w}_t$ indicates the solution at iteration t. When $\boldsymbol{w}_t$ in the iteration is almost unchanged, the unique solution (the fixed point) is found. The procedure of proposed BLS-QMEE is summarized in Algorithm 2.

Algorithm 2 BLS-QMEE

Input: training set: $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$.
Output: output weights: $\boldsymbol{w}$.
Parameters setting: network parameters n, m, p, q, regularization coefficient λ, kernel bandwidth σ, quantization threshold ϵ, maximum iteration number T, termination tolerance η;
Initialization: set $\boldsymbol{w}_0$ according to (1)–(3) and (6);
Calculate $L(\boldsymbol{w}_0) = J(\boldsymbol{w}_0)$ according to (5);
for $t = 1, \ldots, T$ **do**
  Calculate the error samples with $\boldsymbol{w}_{t-1}$: $e_i = y_i - \boldsymbol{a}_i \boldsymbol{w}_{t-1}, i = 1, 2, \ldots, N$;
  Obtain the quantization codes collection C and the quantized numbers of error samples S according to Algorithm 1;
  Calculate $\boldsymbol{w}_t$ according to (14)–(17);
  Compute the objective function $L(\boldsymbol{w}_t)$ according to (13);
  if $|L(\boldsymbol{w}_t) - L(\boldsymbol{w}_{t-1})| < \eta$ **then**
    break;
  end if
end for

3.3 Convergence analysis of BLS-QMEE

The contraction mapping theorem is usually used to analyze the convergence of a fixed-point iteration algorithm [35, 36]. According to the contraction mapping theorem, the convergence of the fixed-point iterations in BLS-QMEE will be guaranteed in the range $\boldsymbol{w} \in \{\boldsymbol{w} \in \mathbb{R}^D : \|\boldsymbol{w}\|_1 \leqslant \nu\}$ if the following inequalities hold:

$$\begin{cases} \|\boldsymbol{f}(\boldsymbol{w})\|_1 \leqslant \nu, \\ \|\nabla_{\boldsymbol{w}} \boldsymbol{f}(\boldsymbol{w})\|_1 = \left\| \frac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial \boldsymbol{w}^{\mathrm{T}}} \right\|_1 \leqslant \tau < 1, \end{cases} \tag{18}$$

where $\|.\|_1$ represents the l_1-norm of a vector or an induced l_1-norm of a matrix, τ is the convergence speed, and $\nabla_{\boldsymbol{w}} \boldsymbol{f}(\boldsymbol{w})$ is the $D \times D$ Jacobian matrix of $\boldsymbol{f}(\boldsymbol{w})$ with respect to $\boldsymbol{w}$, given by

$$\nabla_{\boldsymbol{w}} \boldsymbol{f}(\boldsymbol{w}) = \left[\frac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial w_1} \quad \frac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial w_2} \quad \cdots \quad \frac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial w_D} \right], \tag{19}$$

where

$$\begin{aligned} \frac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial w_k} =& \frac{\partial}{\partial w_k} \left([\boldsymbol{R} + \lambda' \boldsymbol{I}]^{-1} \boldsymbol{P} \right) \\ =& - [\boldsymbol{R} + \lambda' \boldsymbol{I}]^{-1} \left(\frac{1}{\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{L} s_j (e_i - c_j)(a_{ik} - a_{c_j k}) G_\sigma (e_i - c_j) [\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}} [\boldsymbol{a}_i - \boldsymbol{a}_{c_j}] \right) \boldsymbol{f}(\boldsymbol{w}) \\ &+ [\boldsymbol{R} + \lambda' \boldsymbol{I}]^{-1} \left(\frac{1}{\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{L} s_j (e_i - c_j)(a_{ik} - a_{c_j k}) G_\sigma (e_i - c_j) \left(y_i - y_{c_j} \right) [\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}} \right). \end{aligned} \tag{20}$$

The following theorem is proved to ensure Eq. (18) holds.

Theorem 1. It holds that $\|\boldsymbol{f}(\boldsymbol{w})\|_1 \leqslant \nu$ and $\|\nabla_{\boldsymbol{w}} \boldsymbol{f}(\boldsymbol{w})\|_1 \leqslant \tau < 1$ for all $\boldsymbol{w} \in \{\boldsymbol{w} \in \mathbb{R}^D : \|\boldsymbol{w}\|_1 \leqslant \nu\}$ if $\sigma \geqslant \max\{\sigma^*, \sigma^\dagger\}$, where σ^* represents the solution of equation $\varphi(\sigma) = \nu$, and $\sigma^\dagger$ denotes the solution of equation $\psi(\sigma) = \tau (0 < \tau < 1)$ with

$$\varphi(\sigma) = \frac{\sqrt{D} \sum_{i=1}^{N} \sum_{j=1}^{L} s_j \times |y_i - y_{c_j}| \times \left\| \boldsymbol{a}_i - \boldsymbol{a}_{c_j} \right\|_1}{\delta_{\min} [\sum_{i=1}^{N} \sum_{j=1}^{L} s_j \exp(-\frac{(\nu \|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1 + |y_i - y_{c_j}|)^2}{2\sigma^2}) [\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}} [\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]] + 2\sqrt{2\pi} \lambda N^2 \sigma^3}, \tag{21}$$

$$\begin{aligned} \psi(\sigma) = \sqrt{D} \sum_{i=1}^{N} \sum_{j=1}^{L} s_j \left(\nu \left\| \boldsymbol{a}_i - \boldsymbol{a}_{c_j} \right\|_1 + |y_i - y_{c_j}| \right) \left\| \boldsymbol{a}_i - \boldsymbol{a}_{c_j} \right\|_1 \left(\nu \left\| [\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}} [\boldsymbol{a}_i - \boldsymbol{a}_{c_j}] \right\|_1 + |y_i - y_{c_j}| \right. \\ \left. \times \left\| \boldsymbol{a}_i - \boldsymbol{a}_{c_j} \right\|_1 \right) \Bigg/ \left(\sigma^2 \delta_{\min} \left[\sum_{i=1}^{N} \sum_{j=1}^{L} s_j \exp\left(-\frac{(\nu \|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1 + |y_i - y_{c_j}|)^2}{2\sigma^2} \right) [\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}} [\boldsymbol{a}_i - \boldsymbol{a}_{c_j}] \right] \right. \\ \left. + 2\sqrt{2\pi} \lambda N^2 \sigma^5 \right), \end{aligned} \tag{22}$$

where $\delta_{\min}[\cdot]$ denotes the minimum eigenvalue of a matrix and $\sigma \in (0, \infty)$.

Proof. The induced norm of a matrix is compatible with l_p-norm of the corresponding vector; hence

$$\|\boldsymbol{f}(\boldsymbol{w})\|_1 = \left\|[\boldsymbol{R}+\lambda'\boldsymbol{I}]^{-1}\boldsymbol{P}\right\|_1 \leqslant \left\|[\boldsymbol{R}+\lambda'\boldsymbol{I}]^{-1}\right\|_1 \|\boldsymbol{P}\|_1 . \tag{23}$$

Due to the matrix theory and $|e_i - c_j| = |y_i - y_{c_j} - (\boldsymbol{a}_i - \boldsymbol{a}_{c_j})\boldsymbol{w}| \leqslant \|\boldsymbol{w}\|_1 \left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1 + |y_i - y_{c_j}| \leqslant \nu \left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1 + |y_i - y_{c_j}|$,

$$\begin{aligned}\left\|[\boldsymbol{R}+\lambda'\boldsymbol{I}]^{-1}\right\|_1 &\leqslant \sqrt{D}\left\|[\boldsymbol{R}+\lambda'\boldsymbol{I}]^{-1}\right\|_2 \\ &\leqslant \frac{\sqrt{D}}{\delta_{\min}[\sum_{i=1}^{N}\sum_{j=1}^{L} s_j \cdot G_\sigma(\nu\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1 + |y_i - y_{c_j}|)[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]] + \lambda'}.\end{aligned} \tag{24}$$

In addition,

$$\|\boldsymbol{P}\|_1 \overset{\text{(a)}}{\leqslant} \sum_{i=1}^{N}\sum_{j=1}^{L}\left\|s_j \cdot G_\sigma(e_i - c_j)(y_i - y_{c_j})[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}\right\|_1 \overset{\text{(b)}}{\leqslant} \frac{1}{\sqrt{2\pi}\sigma}\sum_{i=1}^{N}\sum_{j=1}^{L} s_j \times \left|y_i - y_{c_j}\right| \times \left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1, \tag{25}$$

in which (a) is deduced from the convexity of the vector l_1-norm, and (b) is from $G_\sigma(x) \leqslant \frac{1}{\sqrt{2\pi}\sigma}$. Combining (23)–(25), the following can be derived:

$$\begin{aligned}\|\boldsymbol{f}(\boldsymbol{w})\|_1 &\leqslant \frac{\frac{1}{\sigma}\sqrt{\frac{D}{2\pi}}\sum_{i=1}^{N}\sum_{j=1}^{L} s_j \times \left|y_i - y_{c_j}\right| \times \left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1}{\delta_{\min}[\sum_{i=1}^{N}\sum_{j=1}^{L} s_j \cdot G_\sigma(\nu\left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1 + \left|y_i - y_{c_j}\right|)[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]] + \lambda'} \\ &= \frac{\sqrt{D}\sum_{i=1}^{N}\sum_{j=1}^{L} s_j \times \left|y_i - y_{c_j}\right| \times \left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1}{\delta_{\min}[\sum_{i=1}^{N}\sum_{j=1}^{L} s_j \exp(-\frac{(\nu\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1 + |y_i - y_{c_j}|)^2}{2\sigma^2})[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]] + 2\sqrt{2\pi}\lambda N^2\sigma^3} \\ &= \boldsymbol{\varphi}(\sigma).\end{aligned} \tag{26}$$

For $\sigma \in (0, \infty)$, $\varphi(\sigma)$ is a continuously monotonically decreasing function, and $\lim_{\sigma\to 0^+}\varphi(\sigma) = \infty$, $\lim_{\sigma\to\infty}\varphi(\sigma) = 0$. Thus, there will be a unique solution σ^* for the equation $\varphi(\sigma) = \nu$ if $\nu > 0$. Moreover, $\|\boldsymbol{f}(\boldsymbol{w})\|_1 \leqslant \varphi(\sigma) \leqslant \nu$ when $\sigma \geqslant \sigma^*$.

Then, to prove $\|\nabla_{\boldsymbol{w}}\boldsymbol{f}(\boldsymbol{w})\|_1 \leqslant \tau$, it suffices to prove $\forall k$, $\|\frac{\partial \boldsymbol{f}(\boldsymbol{w})}{w_k}\|_1 \leqslant \tau$. By (20),

$$\begin{aligned}\left\|\frac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial w_k}\right\|_1 \leqslant &\left\|[\boldsymbol{R}+\lambda'\boldsymbol{I}]^{-1}\left(\frac{1}{\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{L} s_j(e_i - c_j)(a_{ik} - a_{c_jk})G_\sigma(e_i - c_j)[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]\right)\boldsymbol{f}(\boldsymbol{w})\right\|_1 \\ &+ \left\|[\boldsymbol{R}+\lambda'\boldsymbol{I}]^{-1}\left(\frac{1}{\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{L} s_j(e_i - c_j)(a_{ik} - a_{c_jk})G_\sigma(e_i - c_j)(y_i - y_{c_j})[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}\right)\right\|_1 .\end{aligned} \tag{27}$$

Divide (27) into two parts:

$$\begin{aligned}&\left\|[\boldsymbol{R}+\lambda'\boldsymbol{I}]^{-1}\left(\frac{1}{\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{L} s_j(e_i - c_j)(a_{ik} - a_{c_jk})G_\sigma(e_i - c_j)[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]\right)\boldsymbol{f}(\boldsymbol{w})\right\|_1 \\ &\overset{\text{(c)}}{\leqslant} \frac{\nu}{\sigma^2}\left\|[\boldsymbol{R}+\lambda'\boldsymbol{I}]^{-1}\right\|_1\left\{\sum_{i=1}^{N}\sum_{j=1}^{L}\left\|s_j(e_i - c_j)(a_{ik} - a_{c_jk})G_\sigma(e_i - c_j)[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]\right\|_1\right\} \\ &\overset{\text{(d)}}{\leqslant} \frac{\nu}{\sqrt{2\pi}\sigma^3}\left\|[\boldsymbol{R}+\lambda'\boldsymbol{I}]^{-1}\right\|_1\left\{\sum_{i=1}^{N}\sum_{j=1}^{L} s_j\left(\nu\left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1 + \left|y_i - y_{c_j}\right|\right)\left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1\left\|[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]\right\|_1\right\}\end{aligned} \tag{28}$$

in which (c) is deduced because of the convexity of the vector l_1-norm and $\|\boldsymbol{f}(\boldsymbol{w})\|_1 \leqslant \nu$, (d) is because $\left|(e_i - c_j)(a_{ik} - a_{c_j k})\right| \leqslant \left(\nu\left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1 + \left|y_i - y_{c_j}\right|\right)\left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1$ and $G_\sigma(x) \leqslant \frac{1}{\sqrt{2\pi}\sigma}$. Similar to (28),

$$\begin{aligned}
&\left\|[\boldsymbol{R} + \lambda' \boldsymbol{I}]^{-1}\left(\frac{1}{\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{L} s_j(e_i - c_j)(a_{ik} - a_{c_j k})G_\sigma(e_i - c_j)(y_i - y_{c_j})[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}\right)\right\|_1 \\
&\leqslant \frac{1}{\sqrt{2\pi}\sigma^3}\left\|[\boldsymbol{R} + \lambda' \boldsymbol{I}]^{-1}\right\|_1 \left\{\sum_{i=1}^{N}\sum_{j=1}^{L} s_j\left(\nu\left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1 + \left|y_i - y_{c_j}\right|\right) \times \left|y_i - y_{c_j}\right| \times \left\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\right\|_1^2\right\}.
\end{aligned} \tag{29}$$

Combining (27)–(29) and (24), the following can be derived:

$$\begin{aligned}
&\left\|\frac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial w_k}\right\|_1 \\
&\leqslant \frac{\sqrt{D}\sum_{i=1}^{N}\sum_{j=1}^{L} s_j(\nu\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1 + |y_i - y_{c_j}|)\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1(\nu\|[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]\|_1 + |y_i - y_{c_j}| \times \|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1)}{\sqrt{2\pi}\sigma^3(\delta_{\min}[\sum_{i=1}^{N}\sum_{j=1}^{L} s_j \cdot G_\sigma(\nu\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1 + |y_i - y_{c_j}|)[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]] + \lambda')} \\
&= \frac{\sqrt{D}\sum_{i=1}^{N}\sum_{j=1}^{L} s_j(\nu\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1 + |y_i - y_{c_j}|)\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1(\nu\|[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]\|_1 + |y_i - y_{c_j}| \times \|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1)}{\sigma^2\delta_{\min}[\sum_{i=1}^{N}\sum_{j=1}^{L} s_j \exp(-\frac{(\nu\|\boldsymbol{a}_i - \boldsymbol{a}_{c_j}\|_1 + |y_i - y_{c_j}|)^2}{2\sigma^2})[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]^{\mathrm{T}}[\boldsymbol{a}_i - \boldsymbol{a}_{c_j}]] + 2\sqrt{2\pi}\lambda N^2\sigma^5} \\
&= \psi(\sigma).
\end{aligned} \tag{30}$$

For $\sigma \in (0, \infty)$, $\psi(\sigma)$ is a continuously monotonically decreasing function, and $\lim_{\sigma \to 0^+}\psi(\sigma) = \infty$, $\lim_{\sigma \to \infty}\psi(\sigma) = 0$. Thus, there will be a unique solution $\sigma^\dagger$ for the equation $\psi(\sigma) = \tau$ if $0 < \tau < 1$. Besides, $\|\frac{\partial \boldsymbol{f}(\boldsymbol{w})}{\partial w_k}\|_1 \leqslant \psi(\sigma) \leqslant \tau < 1$ when $\sigma \geqslant \sigma^\dagger$. Theorem 1 is proved.

According to the contraction mapping theorem and Theorem 1, the convergence of the fixed-point BLS-QMEE algorithm induced by (17) will be ensured in the range $\boldsymbol{w} \in \{\boldsymbol{w} \in \mathbb{R}^D : \|\boldsymbol{w}\|_1 \leqslant \nu\}$ when provided a kernel bandwidth $\sigma \geqslant \max\{\sigma^*, \sigma^\dagger\}$. It is worth noting that this is only a sufficient condition for convergence, not a necessary condition. According to the scaling in the proof process, the fixed-point BLS-QMEE may still converge if σ does not meet the requirement $\sigma \geqslant \max\{\sigma^*, \sigma^\dagger\}$.

4 Experiments

In this section, the performance of the proposed method BLS-QMEE is evaluated on function approximation, eight public datasets and a practical application. The root mean square error (RMSE) criterion:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{31}$$

is used to measure the modeling performance, where y_i and $\hat{y}_i$ indicate the actual value and the predictive output of the ith sample, respectively. BLS-QMEE is not only compared with the standard BLS but also with several BLS robust variants such as L1RBLS, L2RBLS [19], and CBLS [20]. These robust variants also substitute the MMSE criterion with another robust one. All experiments are implemented using MATLAB(R2016a) on a 3.00 GHz machine with 8 GB RAM.

In the following experiments, the parameters are set as follows: the number of feature mapping groups are defined as N_g, the feature mapping nodes in each group as N_f, the enhancement nodes as N_e, the bandwidth of Gaussian kernel as σ, the quantization threshold as ϵ, and the regularization coefficient as λ. For a fair comparison, all parameters are determined appropriately by grid search, except that λ is simply set to 2^{-30} as in [20]. All methods obtain the best results in the same range of parameters. All experimental results are averaged 10 times to reduce the random error.

4.1 Regression for function approximation

In this subsection, the performance of BLS-QMEE will be measured on the following function approximation task:

$$f(x) = \sin(2\pi x), \quad x \in [0, 1]. \tag{32}$$

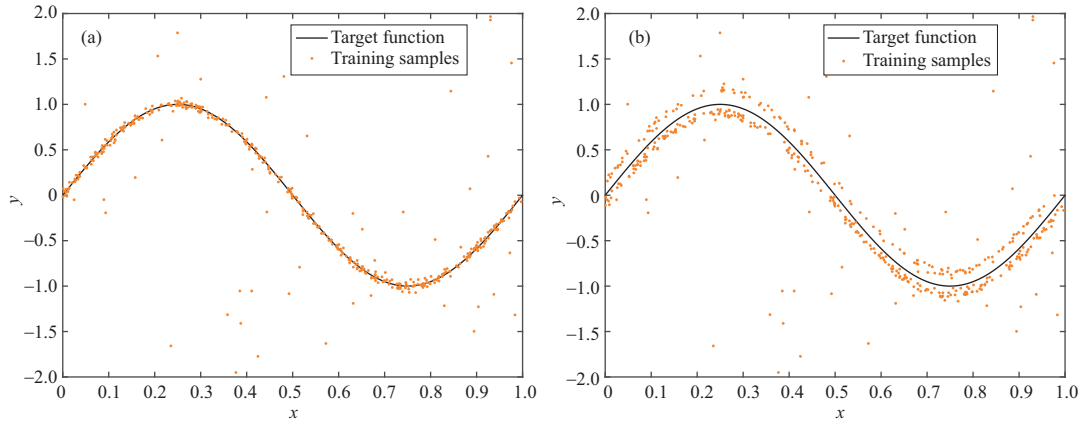

Figure 2 The target sin function and training samples with background noise of different distributions. The occurrence frequency of outliers $r = 20\%$. (a) Case (1) of the background noise; (b) case (2) of the background noise.

Table 1 Performance of different methods on sin function approximation in case (1)

Algorithm	$r = 0\%$			$r = 20\%$			$r = 40\%$		
	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE ± STD($\times 10^{-2}$)	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE ± STD($\times 10^{-2}$)	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE ± STD($\times 10^{-2}$)
BLS	(1, 5, 26)	0.0109	0.34 ± 0.08	(1, 1, 31)	0.0103	13.2 ± 2.45	(7, 1, 6)	0.0202	19.13 ± 3.41
L1RBLS	(10, 1, 196)	0.0200	0.47 ± 0.17	(10, 1, 196)	0.0198	0.62 ± 0.15	(4, 3, 171)	0.0124	0.74 ± 0.25
L2RBLS	(1, 1, 126)	0.0085	0.41 ± 0.15	(1, 1, 126)	0.0069	0.48 ± 0.18	(1, 1, 126)	0.0068	0.67 ± 0.20
CBLS	(2, 1, 46, 2^{-3})	0.0124	**0.32 ± 0.10**	(2, 1, 46, 2^{-4})	0.0129	**0.31 ± 0.07**	(3, 1, 16, 2^{-3})	0.0099	0.44 ± 0.17
BLS-QMEE	(2, 1, 46, 2^{-4}, 0.5)	0.0301	**0.32 ± 0.09**	(2, 1, 46, 2^{-4}, 1)	0.1126	**0.31 ± 0.08**	(2, 1, 76, 2^{-4}, 1)	0.2213	**0.41 ± 0.14**

This is a typical regression task of nonlinear function [20], and the black curve in Figure 2 shows its shape.

In the interval [0, 1], 500 training samples $\{\boldsymbol{X}, \boldsymbol{y}\}$ and 500 testing samples are sampled, both of which are based on the uniform distribution. Following [19, 20], the $\boldsymbol{y}$ of training samples are contaminated by additional noise while testing samples remain noise-free. The noise can be generated by

$$\upsilon_i = (1 - \mu_i)U_i + \mu_i V_i, \tag{33}$$

where μ_i is a binary variable with probability mass $\Pr\{\mu_i = 1\} = r$, $\Pr\{\mu_i = 0\} = 1 - r$, and $0 \leqslant r \leqslant 1$, r is the occurrence frequency of outliers. U_i represents background noise with small variance and V_i indicates outliers with large variance. μ_i, U_i and V_i are independent of each other. In the following simulations, V_i is assumed to follow a Gaussian distribution $\mathcal{N}(0, 10)$. As for U_i, two cases are considered in this paper: (1) Gaussian distribution $\mathcal{N}(0, 0.001)$; (2) mixture Gaussian distribution $(1/3)\mathcal{N}(5, 0.001) + (2/3)\mathcal{N}(-3, 0.001)$. The orange dots in Figure 2 represent the training samples added noise of these two cases at $r = 20\%$. For the parameters' search ranges, N_g is searched in $[1, 20]$ with a stride of 1, N_f in $[1, 20]$ with a stride of 2, and N_e in $[1, 200]$ with a stride of 5. The σ and ϵ are chosen from $\{2^{-7}, 2^{-6}, \ldots, 2^6, 2^7\}$ and $\{0.01, 0.05, 0.1, 0.5, 1, 2, 3, 4, 5\}$, respectively.

Table 1 shows the comparison of different methods when training samples are polluted by different occurrence frequencies of outliers and case (1) of the background noise. The best result in the testing period is highlighted in bold. It can be seen that BLS can perform better than L1RBLS and L2RBLS when the training samples are only polluted by single Gaussian noise with a small variance. However, the performance of BLS drops sharply when there are some outliers in the training data, which indicates BLS is sensitive to outliers. In contrast, other methods show better robustness to outliers, of which BLS-QMEE performs best without consuming too much time.

To evaluate the robustness of different methods in a more complicated environment further, their performance on training samples contaminated by outliers and case (2) of the background noise is shown in Table 2. Results marked in bold are the best. It can be seen that the performance of all methods becomes worse than that in Table 1. L1BLS and L2BLS change the most, which means these two methods may not perform well when dealing with a more complex noise environment. It is worth noting that BLS-QMEE still maintains the best robustness among all compared methods.

Figure 3 shows the prediction results of different methods in case (2). It can be observed that BLS, CBLS, and BLS-QMEE can approximate the function well when there are no outliers in the training

Table 2 Performance of different methods on sin function approximation in case (2)

Algorithm	$r = 0\%$			$r = 20\%$			$r = 40\%$		
	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE $\pm$ STD($\times 10^{-2}$)	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE $\pm$ STD($\times 10^{-2}$)	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE $\pm$ STD($\times 10^{-2}$)
BLS	(1, 1, 196)	0.0229	1.55 $\pm$ 0.32	(1, 1, 31)	0.0109	13.03 $\pm$ 2.41	(7, 1, 6)	0.0201	19.05 $\pm$ 3.14
L1RBLS	(1, 13, 76)	0.0062	7.07 $\pm$ 0.43	(5, 1, 11)	0.0089	7.05 $\pm$ 0.88	(7, 3, 81)	0.0141	7.07 $\pm$ 0.56
L2RBLS	(3, 13, 186)	0.0129	7.16 $\pm$ 0.33	(1, 15, 6)	0.0048	7.13 $\pm$ 0.32	(1, 11, 6)	0.0045	7.16 $\pm$ 0.65
CBLS	(2, 1, 61,2^2)	0.0168	1.54 $\pm$ 0.24	(3, 1, 21, 2^{-1})	0.0107	1.92 $\pm$ 0.50	(1, 1, 91, 2^{-1})	0.0291	2.39 $\pm$ 0.54
BLS-QMEE	(1, 9, 46, 2^{-4}, 0.01)	0.7059	**0.93 $\pm$ 0.36**	(1, 3, 21, 2^{-1},1)	0.0787	**1.89 $\pm$ 0.44**	(2, 1, 61, 2^{-1}, 1)	0.1289	**2.08 $\pm$ 0.41**

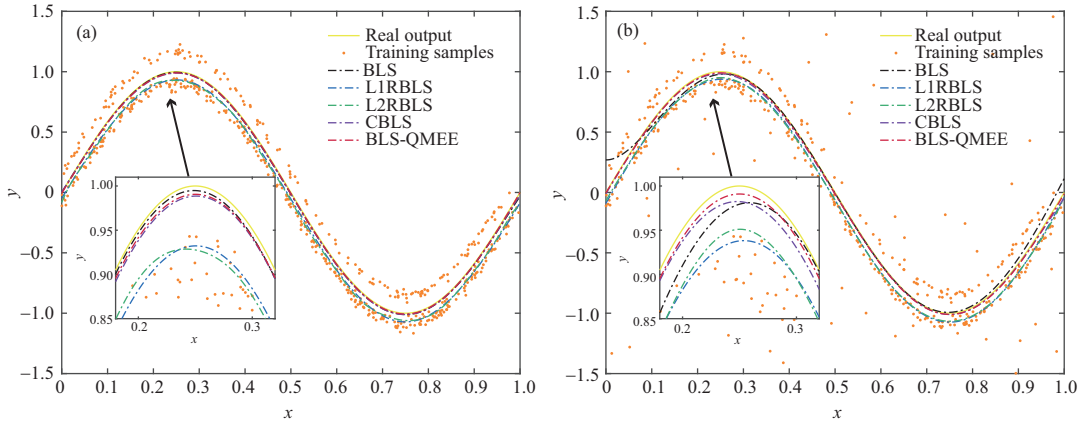

Figure 3 Prediction results of different methods for testing samples of sin function. The training data contaminated by (a) case (2) of the background noise and $r = 0\%$, (b) case (2) of the background noise and $r = 30\%$.

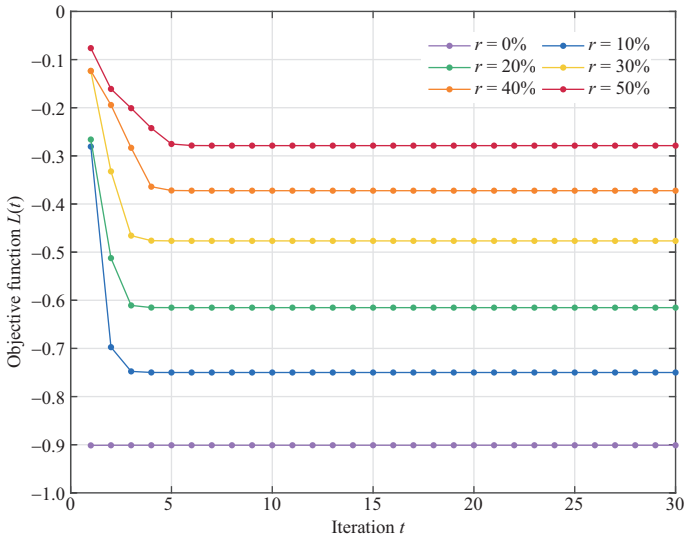

Figure 4 Convergence curves of BLS-QMEE on sin function set with case (1) of the background noise.

samples, while L1RBLS and L2RBLS have a deviation. When there are 30% outliers, the deviation of BLS becomes higher, especially in the interval $[0, 0.1]$. The prediction curve of BLS-QMEE is the closest to the true output curve. Therefore, whether in a Gaussian environment or a more complex environment with mixed Gaussian noise, BLS-QMEE can achieve the best robustness among all comparison methods.

Then, the convergence of BLS-QMEE on the sin function set with different noises is investigated. The objective function at iteration t can be defined as

$$L(t) = -\frac{1}{N^2}\sum_{i=1}^{N}\left(\sum_{j=1}^{L} s_j \cdot G_\sigma(y_i - \boldsymbol{a}_i\boldsymbol{w}_{(t)} - y_{c_j} + \boldsymbol{a}_{c_j}\boldsymbol{w}_{(t)})\right) + \lambda\|\boldsymbol{w}_{(t)}\|_2^2. \tag{34}$$

It indicates BLS-QMEE has converged if $L(t)$ does not decrease with the increase in the number of iterations t. Figure 4 shows the convergence curves of BLS-QMEE when the training samples are con-

Table 3 Description of eight real-world UCI datasets

Dataset	Training	Testing	Attribute
Bodyfat	168	84	14
Cleveland	202	101	13
Housing	337	169	13
Strike	416	209	6
Mortgage	699	350	15
Weather Izmir	974	487	9
Quake	1452	726	3
Abalone	2784	1393	8

taminated by case (1) of the background noise with different outliers. It can be seen that BLS-QMEE has converged at the beginning when the training samples are polluted by single Gaussian noise with small variance, as the initial weights $\boldsymbol{w}_0$ of BLS-QMEE are calculated according to BLS. When there are some outliers, BLS cannot obtain the best result, so the convergence curves of BLS-QMEE decrease in a few iterations and converge finally. Under different occurrence frequencies of outliers, BLS-QMEE can converge in a few iterations, which means it is very efficient.

4.2 Regression for public datasets

In this subsection, the performance of the proposed method is investigated on some small and large real-world datasets from UCI machine learning repository [37], which is proposed for machine learning by the University of California Irvine. Table 3 shows the specification of these datasets.

Both attributes' values and target values in the datasets are normalized into $[0, 1]$ to eliminate the impact of data scales in this experiment. Considering there may exist some noise in real-world datasets, only some random outliers are added to the $\boldsymbol{y}$ of training samples. The distribution of outliers remains the same as in Subsection 4.1; the noise could be generated by $\upsilon_i = \mu_i V_i$.

Table 4 presents the performance comparison between different methods on the UCI datasets with different occurrence frequencies of outliers. The best result is marked in bold. All parameters are searched in the same range as in Subsection 4.1. It can be seen that the existing robust BLS variants L1RBLS, L2RBLS, and CBLS are inferior to the standard BLS in larger datasets without artificial outliers, such as Mortgage, Weather Izmir, Quake, Abalone, but BLS-QMEE performs better. With the increase of outliers, the test RMSE of BLS-QMEE changes little. Besides, BLS-QMEE achieves the best performance in terms of test RMSE in most cases, especially on the dataset Housing, revealing excellent generalization ability. Nevertheless, BLS-QMEE may need a bit more training time than other compared methods in large datasets because it involves some necessary iterations to get the best output weights.

To further compare the performance of different robust BLS variants, Figure 5 shows the performance of L1RBLS, L2RBLS, CBLS, and BLS-QMEE on the Mortgage dataset contaminated by different outliers, whose occurrence frequency is from 0% to 40%. It can be observed that BLS-QMEE achieves the minimum test RMSE regardless of the occurrence frequency of outliers. With the increasing occurrence frequency of outliers, the test RMSE of all methods increases, while BLS-QMEE increases less than other robust BLS variants, which suggests the proposed method has satisfactory robustness.

Since the kernel bandwidth σ is an important parameter in the information entropy-based methods, how σ influences the performance of the proposed BLS-QMEE is investigated on Housing contaminated by 10% outliers, and a comparison between BLS variants based on different information entropy criteria is illustrated in Figure 6. The BLS based on the MCC criterion (CBLS), BLS based on the MEE criterion (BLS-MEE), BLS based on the QMEE criterion [30] (BLS-QMEE*) and the proposed BLS-QMEE are taken into consideration. The σ is chosen from $\{2^{-4}, 2^{-3}, \ldots, 2^{1}, 2^{2}\}$. It can be seen that the performance of all methods will get worse when the σ is too small or too large. A proper σ can be selected by cross-validation in practical applications. With a proper σ, CBLS can perform better than BLS-MEE and BLS-QMEE*, but it is more sensitive to the selection of σ than other methods. In all cases, BLS-QMEE* performs slightly worse than BLS-MEE but the proposed BLS-QMEE combining advantages of MCC and MEE obtains the minimum RMSE, revealing satisfactory performance.

In addition, to exemplify the advantages of the improved online quantization method, the test RMSE and training time under different quantization threshold ϵ are illustrated in Figure 7. Note that BLS-QMEE* and BLS-QMEE are equal to BLS-MEE when $\epsilon = 0$. It can be observed that the test RMSE

Table 4 Performance of different methods on real UCI datasets polluted by different outliers

Dataset	Algorithm	$r = 0\%$			$r = 10\%$			$r = 30\%$		
		Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE ± STD($\times 10^{-2}$)	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE ± STD($\times 10^{-2}$)	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE ± STD($\times 10^{-2}$)
Bodyfat	BLS	(9, 3, 1)	0.013	0.64 ± 0.21	(4, 1, 1)	0.017	14.78 ± 5.07	(1, 1, 1)	0.012	23.95 ± 8.16
	L1RBLS	(11, 13, 6)	0.019	0.57 ± 0.29	(20, 19, 1)	0.050	0.60 ± 0.31	(10, 17, 1)	0.018	0.65 ± 0.28
	L2RBLS	(4, 5, 11)	0.007	0.56 ± 0.29	(3, 7, 6)	0.006	**0.56 ± 0.24**	(16, 5, 1)	0.023	**0.58 ± 0.31**
	CBLS	(19, 19, 6, 2^{-5})	0.032	0.56 ± 0.24	(14, 19, 1, 2^{-5})	0.025	0.60 ± 0.30	(10, 19, 1, 2^{-4})	0.033	0.65 ± 0.25
	BLS-QMEE	(19, 5, 11, 2^{-6}, 0.01)	0.024	**0.55 ± 0.20**	(14, 3, 1, 2^{-5}, 0.5)	0.055	0.59 ± 0.28	(16, 1, 1, 2^{-4}, 1)	0.032	0.67 ± 0.26
Cleveland	BLS	(8, 5, 1)	0.025	11.01 ± 0.72	(1, 1, 1)	0.011	14.71 ± 2.56	(2, 1, 1)	0.013	21.04 ± 5.14
	L1RBLS	(7, 1, 1)	0.011	11.04 ± 0.70	(4, 3, 1)	0.008	11.10 ± 0.83	(4, 1, 6)	0.008	11.36 ± 0.90
	L2RBLS	(1, 7, 1)	0.004	11.08 ± 0.71	(8, 1, 1)	0.012	11.01 ± 0.75	(8, 1, 1)	0.011	11.37 ± 0.71
	CBLS	(4, 5, 1, 2^{-1})	0.004	10.97 ± 0.67	(5, 7, 1, 2^{-2})	0.005	11.04 ± 0.85	(7, 5, 1, 2^{-2})	0.006	11.10 ± 0.81
	BLS-QMEE	(1, 9, 1, 2^{-1}, 0.05)	0.054	**10.91 ± 0.73**	(1, 9, 1, 2^{-1}, 1)	0.023	**10.92 ± 0.83**	(1, 9, 1, 2^{-2}, 1)	0.016	**11.09 ± 0.89**
Housing	BLS	(9, 3, 1)	0.013	8.69 ± 1.09	(8, 1, 1)	0.023	18.6 ± 4.39	(2, 1, 1)	0.005	23.62 ± 4.18
	L1RBLS	(2, 9, 186)	0.011	8.46 ± 1.48	(2, 11, 81)	0.009	9.05 ± 1.27	(5, 5, 26)	0.013	10.71 ± 1.02
	L2RBLS	(2, 13, 46)	0.005	9.08 ± 1.39	(10, 3, 36)	0.018	9.74 ± 1.60	(5, 3, 16)	0.012	10.81 ± 1.26
	CBLS	(3, 7, 46, 2^{0})	0.010	8.39 ± 1.00	(17, 1, 66, 2^{-2})	0.018	9.06 ± 1.25	(16, 3, 21, 2^{-2})	0.015	10.08 ± 1.41
	BLS-QMEE	(2, 9, 186, 2^{7}, 0.5)	0.015	**7.77 ± 0.90**	(2, 15, 31, 2^{-2}, 1)	0.064	**8.89 ± 1.16**	(14, 3, 21, 2^{-2}, 1)	0.097	**9.81 ± 0.8**
Strike	BLS	(14, 7, 1)	0.038	7.62 ± 1.99	(1, 1, 1)	0.011	10.16 ± 1.76	(1, 1, 1)	0.010	12.45 ± 2.97
	L1RBLS	(19, 3, 181)	0.064	**7.23 ± 2.33**	(7, 3, 146)	0.017	7.41 ± 2.21	(12, 15, 6)	0.024	7.68 ± 2.17
	L2RBLS	(7, 3, 96)	0.027	7.35 ± 2.23	(8, 3, 86)	0.016	7.49 ± 2.21	(3, 3, 36)	0.007	7.68 ± 2.13
	CBLS	(2, 19, 71, 2^{-3})	0.012	7.41 ± 2.11	(6, 11, 56, 2^{-3})	0.013	7.41 ± 2.06	(8, 19, 56, 2^{-3})	0.020	7.45 ± 2.16
	BLS-QMEE	(6, 5, 86, 2^{-4}, 1)	0.042	7.37 ± 2.16	(17, 5, 86, 2^{-4}, 2)	0.146	**7.34 ± 2.22**	(9, 5, 56, 2^{-3}, 1)	0.280	**7.44 ± 2.20**
Mortgage	BLS	(9, 3, 126)	0.028	0.51 ± 0.04	(4, 1, 1)	0.017	7.29 ± 1.52	(3, 1, 1)	0.016	12.64 ± 5.06
	L1RBLS	(3, 17, 191)	0.019	0.57 ± 0.04	(2, 17, 146)	0.015	0.71 ± 0.08	(10, 19, 31)	0.029	0.89 ± 0.07
	L2RBLS	(6, 5, 96)	0.015	0.55 ± 0.04	(19, 3, 111)	0.035	0.58 ± 0.05	(18, 3, 51)	0.030	0.66 ± 0.10
	CBLS	(2, 9, 101, 2^{-3})	0.013	0.54 ± 0.03	(3, 11, 76, 2^{-3})	0.020	0.63 ± 0.07	(2, 11, 36, 2^{-3})	0.009	0.81 ± 0.09
	BLS-QMEE	(16, 3, 166, 2^{-4}, 0.1)	0.073	**0.50 ± 0.03**	(17, 3, 101, 2^{-4}, 5)	0.125	**0.52 ± 0.03**	(15, 3, 36, 2^{-4}, 1.5)	0.276	**0.63 ± 0.06**
Weather Izmir	BLS	(3, 5, 51)	0.019	1.86 ± 0.06	(1, 7, 1)	0.012	7.48 ± 1.98	(2, 1, 1)	0.013	12.95 ± 4.19
	L1RBLS	(2, 7, 36)	0.009	1.87 ± 0.06	(2, 5, 51)	0.009	1.91 ± 0.08	(4, 7, 16)	0.010	1.98 ± 0.09
	L2RBLS	(3, 5, 56)	0.010	1.88 ± 0.09	(8, 3, 51)	0.016	1.90 ± 0.08	(3, 5, 26)	0.008	1.93 ± 0.09
	CBLS	(16, 1, 71, 2^{-5})	0.030	1.86 ± 0.08	(5, 5, 41, 2^{-5})	0.030	1.86 ± 0.07	(19, 1, 41, 2^{-4})	0.025	1.92 ± 0.09
	BLS-QMEE	(3, 5, 46, 2^{-5}, 0.1)	0.110	**1.85 ± 0.07**	(3, 7, 56, 2^{-5}, 5)	0.280	**1.85 ± 0.07**	(3, 7, 36, 2^{-5}, 5)	0.294	**1.89 ± 0.09**
Quake	BLS	(20, 1, 11)	0.040	17.16 ± 0.37	(17, 5, 1)	0.041	17.66 ± 0.5	(1, 1, 1)	0.010	18.79 ± 1.58
	L1RBLS	(16, 19, 41)	0.061	17.91 ± 0.44	(16, 1, 81)	0.027	17.93 ± 0.45	(9, 1, 71)	0.017	17.89 ± 0.46
	L2RBLS	(14, 1, 26)	0.022	17.92 ± 0.46	(7, 11, 21)	0.020	17.96 ± 0.43	(7, 11, 16)	0.017	17.95 ± 0.42
	CBLS	(2, 7, 16, 2^{0})	0.010	17.16 ± 0.37	(15, 15, 1, 2^{0})	0.037	**17.17 ± 0.36**	(6, 17, 1, 2^{0})	0.038	17.20 ± 0.40
	BLS-QMEE	(6, 1, 41, 2^{1}, 1)	0.072	**17.15 ± 0.36**	(3, 17, 1, 2^{0}, 2)	0.313	17.19 ± 0.35	(10, 13, 1, 2^{-1}, 5)	0.290	**17.19 ± 0.37**
Abalone	BLS	(1, 9, 36)	0.013	7.49 ± 0.17	(5, 1, 1)	0.019	9.32 ± 0.97	(2, 3, 1)	0.014	10.82 ± 1.44
	L1RBLS	(2, 11, 156)	0.026	7.57 ± 0.26	(1, 17, 66)	0.013	7.61 ± 0.28	(18, 7, 36)	0.051	7.67 ± 0.27
	L2RBLS	(1, 13, 56)	0.012	7.60 ± 0.25	(7, 5, 26)	0.021	7.65 ± 0.24	(6, 7, 21)	0.020	7.72 ± 0.24
	CBLS	(9, 1, 36, 2^{-1})	0.024	7.51 ± 0.20	(2, 7, 21, 2^{-1})	0.009	7.56 ± 0.18	(3, 13, 16, 2^{-1})	0.018	7.63 ± 0.17
	BLS-QMEE	(1, 17, 126, 2^{4}, 1)	0.203	**7.45 ± 0.17**	(1, 17, 26, 2^{-1}, 5)	0.650	**7.54 ± 0.2**	(3, 5, 16, 2^{-2}, 5)	0.884	**7.57 ± 0.23**

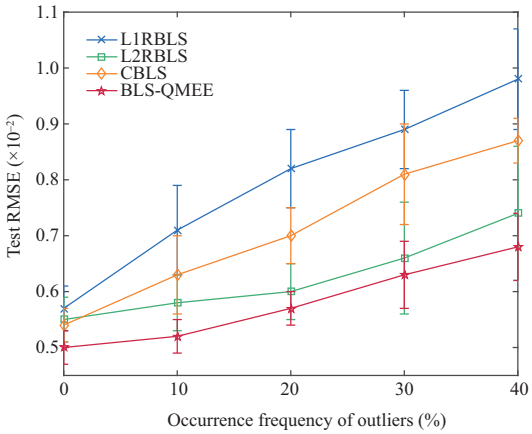

Figure 5 Comparison on test RMSE of L1RBLS, L2RBLS, CBLS, and the proposed BLS-QMEE on the Mortgage dataset contaminated by different outliers.

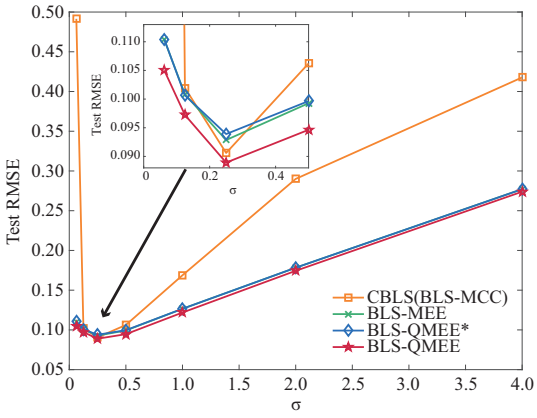

Figure 6 Comparison on test RMSE of CBLS, BLS-MEE, BLS-QMEE*, and the proposed BLS-QMEE with different σ on the Housing dataset contaminated by 10% outliers.

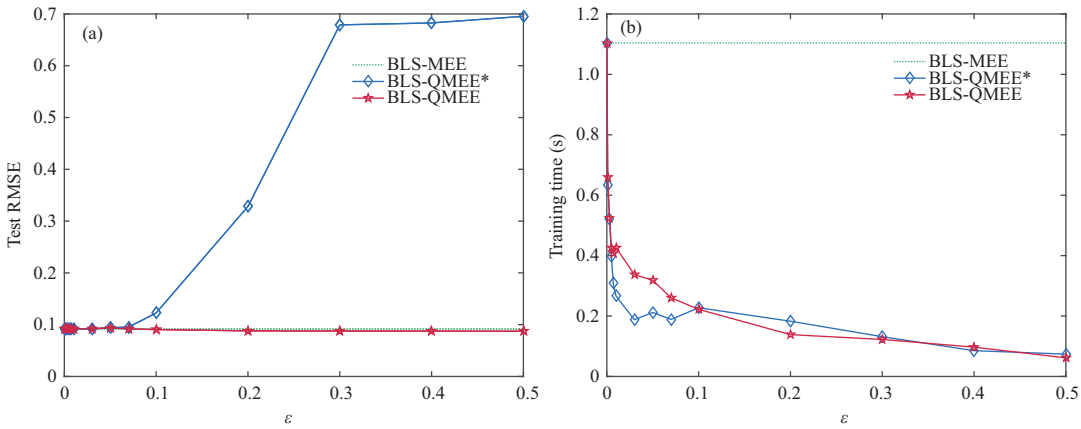

Figure 7 Comparison of BLS-QMEE* and BLS-QMEE with different ϵ on Housing dataset polluted by 10% outliers. (a) Test RMSE; (b) training time.

of BLS-QMEE* and BLS-QMEE is similar to that of BLS-MEE when ϵ is small, but a small ϵ means much training time. With the increase of ϵ, the training time is reduced obviously; BLS-QMEE still maintains good performance, even better than BLS-MEE; however, the performance of BLS-QMEE* becomes worse. Therefore, BLS-QMEE can obtain good performance in a short training time and is more robust to the selection of quantization threshold ϵ than BLS-QMEE*.

4.3 Regression for practical application

Cement is a widely used material in civil engineering. It is of great significance to quickly develop a model for predicting the setting time of various cement since the setting time is an important indicator of cement quality, and it will influence the concrete strength and the construction progress [38]. To evaluate the feasibility of all methods, they will be used to predict the setting time of cement by its chemical compositions and physical properties in this subsection.

The setting time of cement includes initial and final setting time [39]. All data in this experiment are recorded from a cement plant in real time, in which 445 samples are used to predict the initial setting time and 446 samples for predicting the final setting time. For predicting the initial or final setting time, 300 samples are used for training and the rest for testing. Each sample consists of 7 attributes, including the loss on ignition, specific surface area, sieve residue, the contents of CaO, MgO, SO_3, and Cl. All experimental results are averaged 10 times. The training samples are chosen each time randomly.

All attributes' values and setting time are normalized into $[0, 1]$ to remove the impact of data scales. Parameters of all methods are searched in the same range. N_g is searched in the range $[1, 20]$ with a stride

Table 5 Performance of different methods in predicting the initial and final setting time of cement

Algorithm	Initial setting time				Final setting time			
	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE ± STD (min)	Test MAE ± STD (min)	Parameters $(N_g, N_f, N_e, \sigma, \epsilon)$	Training Time (s)	Test RMSE ± STD (min)	Test MAE ± STD (min)
BLS	(15, 5, 13)	0.025	18.76 ± 1.64	14.32 ± 1.06	(11, 3, 25)	0.018	18.48 ± 0.89	14.45 ± 0.73
L1RBLS	(4, 5, 55)	0.012	18.91 ± 1.68	14.34 ± 0.99	(18, 1, 37)	0.025	18.65 ± 0.77	14.68 ± 0.66
L2RBLS	(17, 5, 25)	0.027	18.94 ± 1.22	14.14 ± 0.71	(8, 3, 7)	0.012	18.78 ± 0.81	14.91 ± 0.69
CBLS	(8, 13, 13, 2^0)	0.022	18.97 ± 0.94	14.55 ± 0.56	(15, 1, 7, 2^{-1})	0.010	18.53 ± 0.88	14.84 ± 0.74
BLS-QMEE	(18, 5, 88, 2^{10}, 0.01)	0.061	**18.18 ± 1.12**	**13.79 ± 0.66**	(4, 3, 55, 2^7, 0.01)	0.028	**18.22 ± 1.49**	**14.19 ± 1.05**

Table 6 The percentage (%) of absolute prediction errors for setting time

Algorithm	Absolute initial setting errors			Absolute final setting errors		
	<5 min	<10 min	<20 min	<5 min	<10 min	<20 min
BLS	24.14	48.28	75.17	23.29	44.52	74.66
L1RBLS	24.14	48.28	79.31	22.60	41.10	71.92
L2RBLS	23.45	50.34	79.31	23.29	37.67	73.97
CBLS	**26.90**	43.45	75.17	13.70	40.41	75.34
BLS-QMEE	25.52	**53.10**	**82.07**	**29.45**	**45.89**	**78.77**

of 1, N_f in $[1, 20]$ with a stride of 2, and N_e in $[1, 100]$ with a stride of 3. The σ and ϵ are chosen from $\{2^{-3}, 2^{-2}, \ldots, 2^{10}, 2^{11}\}$ and $\{0.01, 0.05, 0.1, 0.5, 1, 2, 3, 4, 5\}$, respectively. RMSE and mean absolute error (MAE) are utilized to measure the effectiveness of all methods. MAE is calculated by

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|. \tag{35}$$

Both RMSE and MAE are inversely proportional to the effectiveness.

Table 5 compares the performance of different methods for predicting the initial and final setting time. It can be observed that the test RMSE of L1RBLS, L2RBLS, and CBLS are higher than BLS in predicting both initial and final setting time. In terms of test MAE, BLS performs better than L1RBLS, L2RBLS, and CBLS, except that L2RBLS obtains a better result in predicting the initial setting time. It is worth noting that BLS-QMEE obtains both the minimum RMSE and minimum MAE in predicting both initial and final setting time with a short training time. These results illustrate that the proposed method can be a more satisfactory choice compared with BLS and other robust BLS variants for predicting the setting time of cement.

Table 6 shows the percentages of absolute prediction errors in predicting initial and final setting time. The absolute error is calculated by $e_i = |y_i - \hat{y}_i|$. It can be seen that most predicted results of BLS-QMEE are close to the actual value, whose absolute prediction errors are controlled in 20 min. In predicting the initial setting time, although the percentage of absolute errors <5 min in BLS-QMEE is not higher than that in CBLS, the percentage of absolute errors <10 min in BLS-QMEE is at least improved 2.7% than other methods, and the percentage of absolute errors <20 min of BLS-QMEE reaches 82.07%. As for predicting the final setting time, it is worth noting that the percentage of absolute errors <5 min in BLS-QMEE is at least 6% higher than that in other methods. The percentage of absolute errors <20 min of BLS-QMEE also has an absolute improvement compared with other methods. These results suggested that the proposed method is accurate and has a good generalization ability.

5 Conlusion

In this paper, a robust and efficient variant of BLS named BLS-QMEE is proposed, in which the improved QMEE criterion was used as the optimality criterion, and the fixed-point iteration method was used as the optimization method to optimize output weights. Due to the improved quantization operation, BLS-QMEE performs well in a short training time. A sufficient condition is provided to guarantee the convergence of optimization based on the fixed-point iteration. Experiments for function approximation, regression on public datasets, and a real-world application confirm that BLS-QMEE can achieve more satisfactory performance than the standard BLS and other existing robust BLS variants. However, this

paper mainly concentrates on the robustness of BLS in regression tasks, while robust recognition is not considered. Thus, the performance of BLS-QMEE in recognition will be investigated in future work.

Acknowledgements This work was supported by National Key Research and Development Program of China (Grant No. 2019YFB1703600), National Natural Science Foundation of China (Grant Nos. 62006079, 61751202, U1813203, U1801262), China Postdoctoral Science Foundation (Grant No. 2020TQ0105), Science and Technology Major Project of Guangzhou (Grant No. 202007030006), Natural Science Foundation of Guangdong Province (Grant No. 2021A1515011998), Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2019ZT08X214), and Science and Technology Project of Guangzhou (Grant No. 202102020634).

References

1 Deng L, Yu D. Deep learning: methods and applications. Found Trends Signal, 2014, 7: 197–387
2 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444
3 Leung H, Haykin S. The complex backpropagation algorithm. IEEE Trans Signal Process, 1991, 39: 2101–2104
4 Chen C L P, Liu Z. Broad learning system: an effective and efficient incremental learning system without the need for deep architecture. IEEE Trans Neural Netw Learn Syst, 2017, 29: 10–24
5 Chen C L P, Liu Z, Feng S. Universal approximation capability of broad learning system and its structural variations. IEEE Trans Neural Netw Learn Syst, 2018, 30: 1191–1204
6 Hoerl A E, Kennard R W. Ridge regression: biased estimation for nonorthogonal problems. Technometrics, 1970, 12: 55–67
7 Gong X, Zhang T, Chen C L P, et al. Research review for broad learning system: algorithms, theory, and applications. IEEE Trans Cybern, 2022, 52: 8922–8950
8 Jin J, Li Y, Yang T, et al. Discriminative group-sparsity constrained broad learning system for visual recognition. Inf Sci, 2021, 576: 800–818
9 Sheng B, Li P, Zhang Y, et al. GreenSea: visual soccer analysis using broad learning system. IEEE Trans Cybern, 2020, 51: 1463–1477
10 Jin J W, Liu Z L, Chen C L P. Discriminative graph regularized broad learning system for image recognition. Sci China Inf Sci, 2018, 61: 112209
11 Wang Y, Jia P, Cui H, et al. A novel regression prediction method for electronic nose based on broad learning system. IEEE Sens J, 2021, 21: 19374–19381
12 Zhang D, Li T S, Chen C L P, et al. Target tracking algorithm based on a broad learning system. Sci China Inf Sci, 2022, 65: 154201
13 Yu W, Zhao C. Broad convolutional neural network based industrial process fault diagnosis with incremental learning capability. IEEE Trans Ind Electron, 2019, 67: 5081–5091
14 Lin J, Liu Z, Chen C L P, et al. Three-domain fuzzy wavelet broad learning system for tremor estimation. Knowledge-Based Syst, 2020, 192: 105295
15 Liu Z, Chen C L P, Feng S, et al. Stacked broad learning system: from incremental flatted structure to deep model. IEEE Trans Syst Man Cybern Syst, 2020, 51: 209–222
16 Ye H, Li H, Chen C L P. Adaptive deep cascade broad learning system and its application in image denoising. IEEE Trans Cybern, 2020, 51: 4450–4463
17 Guo D N, Wu Y H, Shitz S S, et al. Estimation in Gaussian noise: properties of the minimum mean-square error. IEEE Trans Inform Theor, 2011, 57: 2371–2385
18 Hampel F R, Ronchetti E M, Rousseeuw P J, et al. Robust Statistics: The Approach Based on Influence Functions. Hoboken: John Wiley & Sons, 2011
19 Jin J W, Chen C L P. Regularized robust broad learning system for uncertain data modeling. Neurocomputing, 2018, 322: 58–69
20 Zheng Y, Chen B, Wang S, et al. Broad learning system based on maximum correntropy criterion. IEEE Trans Neural Netw Learn Syst, 2020, 32: 3083–3097
21 He R, Zheng W-S, Hu B-G. Maximum correntropy criterion for robust face recognition. IEEE Trans Pattern Anal Mach Intell, 2010, 33: 1561–1576
22 Chen B, Dang L, Gu Y, et al. Minimum error entropy Kalman filter. IEEE Trans Syst Man Cybern Syst, 2019, 51: 5819–5829
23 Principe J C. Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives. Berlin: Springer, 2010
24 de Sá J P M, Silva L M A, Santos J M F, et al. Minimum Error Entropy Classification. Berlin: Springer, 2013
25 Jiang J, He Y L, Dai D X, et al. A new kernel density estimator based on the minimum entropy of data set. Inf Sci, 2019, 491: 223–231
26 Peng S, Ser W, Chen B, et al. Robust constrained adaptive filtering under minimum error entropy criterion. IEEE Trans Circuits Syst II, 2018, 65: 1119–1123
27 Wang Y, Tang Y Y, Li L. Minimum error entropy based sparse representation for robust subspace clustering. IEEE Trans Signal Process, 2015, 63: 4010–4021
28 Lu L, Zhao H, Chen C. A new normalized subband adaptive filter under minimum error entropy criterion. Signal Image Video Process, 2016, 10: 1097–1103

29 Wang G, Chen B, Yang X, et al. Numerically stable minimum error entropy Kalman filter. Signal Process, 2021, 181: 107914
30 Chen B, Xing L, Zheng N, et al. Quantized minimum error entropy criterion. IEEE Trans Neural Netw Learn Syst, 2018, 30: 1370–1380
31 Silverman B W. Density Estimation for Statistics and Data Analysis. New York: Routledge, 2018
32 Parzen E. On estimation of a probability density function and mode. Ann Math Statist, 1962, 33: 1065–1076
33 Chen B, Xing L, Xu B, et al. Insights into the robustness of minimum error entropy estimation. IEEE Trans Neural Netw Learn Syst, 2016, 29: 731–737
34 Agarwal R P, Meehan M, O'regan D. Fixed Point Theory and Applications. Cambridge: Cambridge University Press, 2001
35 Heravi A R, Hodtani G A. A new robust fixed-point algorithm and its convergence analysis. J Fixed Point Theor Appl, 2017, 19: 3191–3215
36 Xie Y, Li Y, Gu Y, et al. Fixed-point minimum error entropy with fiducial points. IEEE Trans Signal Process, 2020, 68: 3824–3833
37 Blake C. UCI repository of machine learning databases. 1998. http://www.ics.uci.edu/ mlearn/MLRepository.html
38 Shi X. Effect of water-cement ratio on setting time of cement paste (in Chinese). Sichuan Cem, 2018, 7: 8
39 Baseri H, Rabiee S M, Moztarzadeh F, et al. Mechanical strength and setting times estimation of hydroxyapatite cement by using neural network. Mater Des (1980–2015), 2010, 31: 2585–2591