Outlier-robust extreme learning machine for regression problems

Kai Zhang, Minxia Luo *

Department of Information and Mathematics Sciences, China Jiliang University, Hangzhou, Zhejiang Province 310018, China

A R T I C L E I N F O

Article history:
Received 7 October 2013
Received in revised form 6 May 2014
Accepted 17 September 2014
Communicated by M. Verleysen
Available online 7 October 2014

Keywords:
Extreme learning machine
\( \ell_1 \text{-norm} \)
Augmented Lagrange multipliers method
Outlier robustness

A B S T R A C T

Extreme learning machine (ELM), as one of the most useful techniques in machine learning, has attracted extensive attentions due to its unique ability for extremely fast learning. In particular, it is widely recognized that ELM has speed advantage while performing satisfying results. However, the presence of outliers may give rise to unreliable ELM model. In this paper, our study addresses the outlier robustness of ELM in regression problems. Based on the sparsity characteristic of outliers, this work proposes an outlier-robust ELM where the \( \ell_1 \)-norm loss function is used to enhance the robustness. Specially, the fast and accurate augmented Lagrangian multiplier method is applied to guarantee the effectiveness and efficiency. According to the experiments on function approximation and some real-world applications, the proposed approach not only maintains the advantages from original ELM, but also shows notable and stable accuracy in handling data with outliers.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Data sets for machine learning and statistical modeling in recent decades, especially in the age of big data, are becoming larger and it becomes easy to access large amounts of information about a studied phenomenon [1]. However, due to human or instrumentation error in the data acquisition processes, outliers which are outstanding and far away from other regular samples may occur in the training data [2]. The requirement for building an outlier-robust model thereby comes up as the regular learning algorithms have a natural tendency to favor the outliers, which appears to significantly reduce the accuracy and reliability of the learned model.

Extreme learning machine (ELM) as proposed by Huang et al. [3] is an effective and efficient learning method for its unique characteristics and notable performance [4]. The merits of ELM are attributed to the following features: (i) As the same as most neural networks, ELM not only has the capability of approximating the unknown functions embedded in a large amount of training data but also holds parallel structures to perform fast and efficient parallel computing during the training as well as in the process of testing. (ii) Numerous empirical studies have shown that ELM tends to have better scalability and generalization performance than traditional SVM [3,5]. (iii) ELM performs at extremely fast learning speed, whereas other traditional learning algorithms have to face the challenging issue of high computational cost. More precisely, this is the major merit essentially because all the hidden node parameters (input weights and biases) are randomly generated (even before ELM sees the training data) without tuning, and therefore the output weights can be analytically determined. In virtue of these remarkable merits, ELM has attracted intensive attentions in a wide range of fields. Particularly, for regression problems, ELM has been widely applied in many real-world engineering applications, such as electricity price forecasting [6], stock market forecasting [7,8] and permeability prediction for ferromagnetic material [9].

Though ELM has many advantages, it has been remarked that ELM tends to suffer from the presence of outliers in the training data, as is likely to happen when dealing with real-world applications [10]. To the best of our knowledge, little work has been done on the emphasis of outlier-robustness of ELM. Huang et al. [5] proposed the regularized ELM on the basis of the original ELM, and provided that the norm of output weight, as a regularization term, plays an important role on the final performance. Deng et al. [11] proposed weighted regularized ELM by incorporating the weighted least squares scheme and regularized ELM. Their work showed promising results on the outlier robustness problems. Horata et al. [10] proposed iteratively reweighted ELM, however, the performance is not ideal for the lack of regularization term and the computational amount largely increases because every iteration involves as much as the time spent by original ELM. All those methods use the training loss function characterized by \( \ell_2 \)-norm (or sum of squares) criterion to learn the model. However, it has been pointed out that the \( \ell_2 \)-norm is prone to be badly affected by outliers, since the \( \ell_2 \)-norm magnifies the effect of outliers associated with large deviations [12]. Typically, the ELM model with \( \ell_2 \)-norm loss function tends to be unstable in the presence of outliers.

In this paper, we present a new type of ELM with \( \ell_1 \)-norm loss function for outlier robustness problems. The reason why we use \( \ell_1 \)-norm rather than \( \ell_2 \)-norm includes two folds. First, the \( \ell_1 \)-norm is
more robust to atypical observations than the $\ell_2$-norm and $\ell_1$-norm loss function has been widely used to tackle outliers (see, e.g., [13–16]). Second, sparsity stands out as the outliers usually occupy little fraction of the whole training samples. Recent developments in two research fields, i.e., compressive sensing [17–20] and robust principal component analysis [21,22], have theoretically revealed that under certain conditions, sparsity can be achieved with the $\ell_1$-norm. In order to solve the resulting optimization problem, we take advantage of the augmented Lagrange multipliers (ALM) method [23] for implementation. Although this method is generally associated with the iterative scheme, it turns out that the simple formulas in each iteration make the computation very fast for efficient implementation and competitive comparisons. Taking into consideration that our proposed method can be viewed as a special ELM for outlier robustness problems, we refer to it as ORELM for simplicity.

The rest of this paper is organized as follows: in Section 2, we sketch the related work which is composed of original ELM, regularized ELM and weighted regularized ELM. Section 3 presents the ORELM method. To verify the efficiency and effectiveness of the proposed method on outlier robustness problems, comprehensive experiments for regression and binary classification applications are conducted in Section 4. Finally, we conclude the paper and provide some ideas for future research in Section 5.

2. Related works

2.1. Problem description and the original extreme learning machine (ELM)

The extreme learning machine (ELM) was proposed by Huang et al. as an original way of building a single hidden layer feed forward neural network (SLFN). The main nature of ELM is the random initialization of the input weights and biases, thus being only necessary for the optimization of the weights connecting the output layer, it surpasses a series of time-consuming algorithms such as BP algorithms.

In this paper, for the sake of simplicity, we consider the usual setup of ELM for regression and binary classification problems with single output (see, e.g., [24,25]).

Given a data set containing $N$ training samples $(\mathbf{x}_i, y_i)_{i=1}^N$, where input $\mathbf{x}_i \in \mathbb{R}^p$ and corresponding desired output $y_i \in \mathbb{R}$, under the assumption that the model perfectly describes the input–output relation, the output weights with a regularization parameter. This can be expressed as

$$\min_{\beta} C(\mathbf{y} - \mathbf{H}\beta)^2 + \|\beta\|_2^2$$

It is easy to obtain that Eq. (5) is equivalent to the following constrained optimization problem:

$$\min_{\beta} C(\mathbf{y} - \mathbf{H}\beta)^2 + \|\beta\|_2^2$$

subject to $\mathbf{y} - \mathbf{H}\beta = \mathbf{e}$

and the corresponding Lagrangian is defined by

$$L(\beta, \lambda, \mathbf{e}) = C(\mathbf{y} - \mathbf{H}\beta)^2 + \lambda^T(\mathbf{y} - \mathbf{H}\beta - \mathbf{e})$$

where $\mathbf{e} = [e_1, e_2, \ldots, e_n]^T$ contains the $N$ training error variables, $\lambda$ is a column vector of the Lagrangian multiplier. According to Karush–Kuhn–Tucker (KKT) theorem, the optimality conditions are given by

$$\begin{align}
\frac{\partial L}{\partial \beta} &= 0 \Rightarrow 2\beta - H^T\lambda = 0 \\
\frac{\partial L}{\partial \mathbf{e}} &= 0 \Rightarrow 2\mathbf{C}\mathbf{e} - \lambda = 0 \\
\frac{\partial L}{\partial \lambda} &= 0 \Rightarrow \mathbf{y} - \mathbf{H}\beta - \mathbf{e} = 0
\end{align}$$

This leads to the following solution of $\beta$:

$$\hat{\beta} = \left(H^TH + \frac{1}{\zeta}C\right)^{-1}H^T\mathbf{y}$$

As a reminder, when the number of training samples is less than the number of hidden nodes, an alternative solution of $\beta$ with less computational cost is as follows:

$$\hat{\beta} = H^T\left(HH^T + \frac{1}{\zeta}C\right)^{-1}\mathbf{y}$$
2.3. Weighted regularized extreme learning machine (WRELM)

Weighted regularized ELM was originally proposed to reduce the influence of outliers [11]. There are three steps in the training phase: (i) Initialize the network model by RELM. Because of the presence of outliers, the learned output weight maybe unstable and thus need to be updated. Consequently, the performance yielded by this current model tends to be unreliable. (ii) Conduct a weighting scheme based on the RELM training error in the first step. More precisely, samples with high training error are assigned with small weights, so that the negative effect of outliers can be cut down. (iii) Apply RELM with weighting information of the training samples to update the output weight.

The main idea of WRELM is the weights that are imposed on the training samples. One can weight the RELM error variable \(e_i\) by weighting factor \(w_i\) to obtain a robust model against outliers. So, \(|e_i|^2\) will be extended as \(\|W\|_2^2\), where \(W = \text{diag}(w_1, w_2, \ldots, w_N)\). There have been a lot of methods on robust estimation of \(W\), as suggested by Suykens et al. [28] and Deng et al. [11], we take the following expression:

\[
\begin{align*}
W_i &= \begin{cases} 
1, & |e_i|/\hat{s} \leq c_1 \\times 2 - |e_i|/\hat{s} \leq c_1, & |e_i|/\hat{s} < c_2 \\
10^{-4}, & \text{otherwise}
\end{cases} 
\end{align*}
\]

where \(\hat{s}\) is robust estimate of the standard deviation of the RELM error variables:

\[
\hat{s} = 2 \times 0.6745 \cdot \text{IQR}
\]

The inter-quartile range IQR is the difference between the 75th percentile and the 25th percentile. In the estimate \(\hat{s}\), one takes into account how much the estimated error distribution deviates from a Gaussian distribution. Due to the fact that for a Gaussian distribution there will be very few residuals larger than 2.5\(\hat{s}\), the constants \(c_1\) and \(c_2\) are typically set as \(c_1 = 2.5\) and \(c_2 = 3\).

As a result, the mathematic model of WRELM can be described as

\[
\min_{\beta} \|W\|_2^2 + \|\beta\|_2^2 \quad \text{subject to } y - H\beta = e
\]

Similar to Section 2.2, it can be derived that [11]

\[
\hat{\beta} = \left(H^T W^T H + \frac{1}{c} \right)^{-1} H^T W^T y
\]

As mentioned, WRELM mainly concentrates on eliminating the influence of outliers. Thus, it appears to be a robust solution when facing data contaminated by outliers. Nevertheless, there also remain drawbacks that limit its practical performance. On the one hand, because the weight estimation depends upon the previous RELM solution and plays a vital role in the final model, in this sense, WRELM requires good previous RELM solution which is a challenge for guarantee. On the other hand, while the influence of outliers is reduced, some good samples (which are not outliers) might be imposed with small weight which will weaken the training. This is particularly obvious if an outlier and a good sample have same and relative large RELM training error. In order to handle the above drawbacks, an alternative way is to take advantage of iteratively reweighted methods, however, the computational time will largely increase which keeps away from our original purpose of fast learning.

3. Main results

In general, the outliers occupy little fraction of the whole training samples, for training error \(e\), this characteristic can be described with sparsity. It is well-known that sparsity can be better reflected by using \(\ell_0\)-norm rather than \(\ell_2\)-norm. Therefore, we instead look for output weight \(\beta\) with small \(\ell_2\)-norm such that the training error \(e\) is sparse, i.e.,

\[
\min_{\beta} \|e\|_0 + \|\beta\|_2^2 \quad \text{subject to } y - H\beta = e
\]

Eq. \((15)\) is a non-convex programming problem. An easy way to solve this problem is to recast it into a tractable convex relaxation form without loss of the sparsity characteristic. According to compressive sensing and robust principal component analysis, sparse term can be achieved with \(\ell_1\)-norm. It is easy to see that replacing the \(\ell_2\)-norm by \(\ell_1\)-norm in Eq. \((15)\) not only guarantees the sparsity characteristic but also leads to the overall minimization convex. As a result, we get the convex relaxation of Eq. \((15)\), which has the form as follows:

\[
\min_{\beta} \|e\|_1 + \frac{1}{c}\|\beta\|_2^2 \quad \text{subject to } y - H\beta = e
\]

Eq. \((16)\) is a constrained convex optimization problem and it fits perfectly the applicable range of the augmented Lagrange multiplier (ALM) method. Correspondingly, the augmented Lagrangian function is given by

\[
L_{\mu}(e, \beta, \lambda) = \|e\|_1 + \frac{1}{c}\|\beta\|_2^2 + \lambda^T (y - H\beta - e) + \frac{\mu}{2}\|y - H\beta - e\|_2^2
\]

where \(\lambda \in \mathbb{R}^n\) is a vector of the Lagrange multiplier and \(\mu\) is a penalty parameter. As suggested by Yang and Zhang [29], we choose \(\mu = 2N \|y\|_1\) in this paper. The ALM algorithm estimates the optimal solution \((e, \beta)\) and the Lagrange multiplier \(\lambda\) by iteratively minimizing the augmented Lagrangian function. In short, given \((\lambda_k, \mu)\), the iterative scheme of ALM is

\[
\begin{align*}
(e_{k+1}, \beta_{k+1}) &= \arg \min_{e, \beta} L_{\mu}(e, \beta, \lambda_k) \\
\lambda_{k+1} &= \lambda_k + \mu \left(y - H\beta_{k+1} - e_{k+1}\right)
\end{align*}
\]

The minimization in the first stage (i.e., Eq. \((18a)\)) of the ALM iteration could be implemented by the widely used alternating direction technique which updates the two unknowns \(e\) and \(\beta\) serially. More precisely, ALM aims to solve the following problems to generate new iteration:

\[
\begin{align*}
\beta_{k+1} &= \arg \min_{\beta} L_{\mu}(e_k, \beta, \lambda_k) \\
e_{k+1} &= \arg \min_{e} L_{\mu}(e, \beta_{k+1}, \lambda_k)
\end{align*}
\]

According to Zhang et al. [30], \(\beta_{k+1}\) admits the following explicit solution:

\[
\beta_{k+1} = (H^T H + 2/\mu I)^{-1} H^T y - e_k + \lambda_k/\mu
\]

and \(e_{k+1}\) is given explicitly by

\[
e_{k+1} = \text{sign}(y - H\beta_{k+1} + \lambda_k/\mu) - 1/\mu, 0) = \text{sign}(y - H\beta_{k+1} + \lambda_k/\mu)
\]

where "\(\circ\)" represents the element-wise multiplication.

It is easy to see that the computation amount of each iteration is mainly spent on solving the inverse of \(H^T H + 2/\mu I\). Fortunately, in each iteration, there shows same matrix \((H^T H + 2/\mu I)^{-1} H^T\) which can be precalculated before the start of whole iterations, thus making the computation very fast. Meanwhile, it should be noted that a continuation technique which varies \(\mu\) can speed up the convergence with less iterations [31,32]. However, this requires various calculations of \((H^T H + 2/\mu I)^{-1} H^T\) which will lead to a larger amount of computation, even though a fast SVD-based technique can be employed (see [24]). Also, it should be noted that...
although this iterative method is not necessarily the fastest in reaching an extremely high accuracy when training data are outlier-free, it is arguably the most effective in obtaining the best achievable level of accuracy whenever data contain outliers, which is the case most relevant to practical applications. For convenience, we terminated the iterations when the number of iterations reaches a predefined threshold $\text{MaxIter}$. Other possible termination criteria can be found in [20,33]. For the convergence of this ALM type, refer to the work by Lin et al. [31] and He et al. [34].

Based on this analysis, we now summarize the following algorithm of ORELM.

**Algorithm 1.** Outlier-robust extreme learning machine (ORELM).

**Input:** $N$ training samples $(x_i, y_i)$, $i = 1, 2, \ldots, N$, activation function $g(x)$, hidden node number $L$, regularization parameter $C$ and maximum number of iterations $\text{MaxIter}$.

1: Randomly generate hidden node parameters $(w_i, b_i), i = 1, 2, \ldots, L$.
2: Calculate the hidden layer output matrix $H(w_1, \ldots, w_L, x_1, \ldots, x_N, b_1, \ldots, b_L)$.
3: Parameter setting and initialization: $\mu = 2N/\|y\|_1$, $P = (H^TH + 2/C\mu I)^{-1}H^T$, $e_1 = 0$, $\lambda_1 = 0$.
4: for $k = 1$ to $\text{MaxIter}$ do
5: $\beta_{k+1} = Py − e_k + \lambda_k/\mu$;
6: $e_{k+1} = \text{shrink}(y − H\beta_{k+1} + \lambda_k/\mu, 1/\mu)$;
7: $\lambda_{k+1} = \lambda_k + \mu(y − H\beta_{k+1} − e_{k+1});$
8: $k ← k + 1$;
9: end for

4. Experimental results

This section shows performance of the proposed ORELM algorithm on function regression and real-world applications, compared with ELM, RELM and WRELM. The following experimental design aims to verify the outlier robustness and the computational efficiency of ORELM. First, a comparable empirical demonstration on a nonlinear function approximation problem is illustrated. Subsequently, experiments for real-world applications, such as abalone regression and breast cancer diagnosis, are conducted for contrastive study. Note that the sigmoid function $g(x) = 1/(1 + \exp(−x))$ is chosen as activation function for all models and the maximum number of iterations of ORELM is set as $\text{MaxIter} = 20$ unless otherwise specified. All the experiments are carried out in the Matlab 7.14 (2012a) environment running in Intel (R) Core(TM) i3 M380 processor with the speed of 2.55 GHz.

4.1. Outlier robustness evaluation via function approximation

First of all, we consider the approximation of the SinC function which is a popular choice to perform regression problems in the literature [3,13]. The target function is as follows:

$$y(x) = \begin{cases} \sin(x)/x, & x \neq 0, \\ 1, & x = 0. \end{cases}$$  \hspace{1cm} (22)

For each data point $(x,y)$, $x$ used for training is generated from the uniform distribution on the interval $[-10, 10]$, while $x$ used for testing is chosen with uniform step length. Additionally, for the purpose of evaluating the outlier robustness comprehensively, some $y$’s of the training points have been added by two types of disturbance values which are randomly chosen from the set $(-1, 1)$ and interval $[-1, 1]$. Correspondingly, these points are termed $[-1, 1]$ outliers and $[-1, 1]$ outliers in this paper. In regard to the two different types of outliers, experiments will be conducted in the following sections. It should be pointed out that the testing data do not contain any outliers.

4.1.1. Results on training data contaminated by $[-1, 1]$ outliers

The objective of this evaluation is to measure the outlier robustness of the different algorithms with respect to $[-1, 1]$ outliers. We generate 200 training data points, among which there are some $[-1, 1]$ outliers for contamination purpose. Taking Fig. 1(a) as an illustration, the data points far away from the SinC curve are regarded as outliers. As for testing data, 11,001 data points are generated, where $x$ is chosen in $[-11, 11]$. The reason why we choose a longer interval $[-11, 11]$ rather than the training interval $[-10, 10]$ is that the predictive ability of a learned model thereby can be assessed in a more reasonable manner [35]. Owing to the fact that the output weight is the focus for comparative study, the hidden nodes parameters in each trial are kept same. Therefore, once training phase is done, the four learned models hold same structural parameters except the output weight. Note that the hidden nodes number $L$ is assigned with 20, and regularization parameter $C$ is chosen from the range $(2^5, 2^{10}, 2^{15}, 2^{20}, 2^{25}, 2^{30})$.

Fig. 1(a)-(d) shows the performances of the four different algorithms on training data contaminated by a different number of outliers. The number of outliers is 10, 20, 40 and 80, respectively. From those figure, intuitively, we can obtain the results as follows:

1. The curves derived from ELM and RELM are pushed toward the side of outliers.
2. RELM has a great improvement over ELM due to the regularization.
3. When the number of outliers is low (say 10 and 20), WRELM shows good performance in the interval $[-10, 10]$ which is much better than that in the additional interval. As shown in Fig. 1(d), the curve derived from WRELM depart from the true SinC curve to a certain degree when the training data contain a relative large amount of outliers.
4. ORELM achieves the best performance among those algorithms. In particular, ORELM is more robust to the number of outliers (more precisely, the contamination rate) than WRELM.

| Table 1 | shows the comparison of training and testing RMSE (Root Mean Square Error) by ELM, RELM, WELM and ORELM in Fig. 1(a)-(d). It can be seen that even though ELM obtains the smallest training RMSE, its testing RMSE is not satisfactory. Compared to ELM and RELM, WRELM gains significant improvement in testing RMSE. However, further improvement can be obtained by ORELM.

4.1.2. Results on training data contaminated by $[-1, 1]$ outliers

In this section, another type of outlier (i.e., $[-1, 1]$ outliers) is taken into consideration for further evaluation. We generate 400 data points for training and 10,001 data points in $[-10, 10]$ for testing. In an attempt to prove the outlier robustness of our proposed ORELM thoroughly, the training times and testing results of RMSE with respect to regularization parameter $C$, number of hidden nodes $L$ and number of outliers are presented in Table 2. Note that, for the sake of eliminating the random error in this experiment, 100 trials have been conducted for each pair $(C, L)$ and the training time and testing RMSE shown are averaged. It can be seen, in accordance with the results from Section 4.1.1, ORELM outperforms the other algorithms in testing RMSE with significant improvement. In addition, ORELM performs better than ELM, RELM and WRELM, independent of the regularization parameter and the number of hidden nodes. Moreover, ORELM shows more remarkable computational efficiency than WRELM. Hence, ORELM outperforms WRELM not only in robustness to outlier but also in computational efficiency.

Fig. 2 shows an example to illustrate the curves derived from the four algorithms and corresponding curve of objective function value of ORELM with respect to the number of iterations. As one
can see, the convergence rate of ORELM is quick. This phenomenon accords that ALM is fast for implementation.

4.2. Outlier robustness evaluation via real-world applications

In this section, we conduct the performance comparison of the proposed ORELM with the other three algorithms for three real problems: two regression tasks including Abalone and Housing, and one binary classification task for breast cancer diagnosis. All the data sets are obtained from the well-known UCI machine learning repository [36]. The specifications of the data sets are listed in Table 3.

Considering that the outliers do not have normal target values, in the data preprocess, only the attributes’ values are normalized into $\frac{1}{C0 \cdot C1}$, while the target values are not. The results are averaged over 50 trials. In each trial, the training data are randomly generated from the data sets while the remaining data are used for testing.

4.2.1. Regression applications

Abalone data set is used for predicting the age of abalone from eight physical measurements. Because standard measurement of the age of abalone is a time-consuming and tiring task, other easily obtained measurements are usually used for age prediction. Housing

Table 1

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>RMSE</th>
<th>(a) Training</th>
<th>Testing</th>
<th>(b) Training</th>
<th>Testing</th>
<th>(c) Training</th>
<th>Testing</th>
<th>(d) Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELM</td>
<td>0.1996</td>
<td>0.4216</td>
<td>0.3015</td>
<td>0.5414</td>
<td>0.4203</td>
<td>0.2988</td>
<td>0.6049</td>
<td>0.4125</td>
<td></td>
</tr>
<tr>
<td>RELM</td>
<td>0.2108</td>
<td>0.1138</td>
<td>0.3085</td>
<td>0.0748</td>
<td>0.4378</td>
<td>0.0932</td>
<td>0.6218</td>
<td>0.1526</td>
<td></td>
</tr>
<tr>
<td>WRELM</td>
<td>0.2221</td>
<td>0.0300</td>
<td>0.3152</td>
<td>0.0229</td>
<td>0.4486</td>
<td>0.0246</td>
<td>0.6289</td>
<td>0.0925</td>
<td></td>
</tr>
<tr>
<td>ORELM</td>
<td>0.2235</td>
<td>0.0083</td>
<td>0.3162</td>
<td>0.0016</td>
<td>0.4472</td>
<td>0.0053</td>
<td>0.6323</td>
<td>0.0088</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1. Performance on contaminated training data with a different number of $[-1, 1]$ outliers: (a) 10 outliers, (b) 20 outliers, (c) 40 outliers, (d) 80 outliers.
data set concerns the relationship between housing values and some attributes such as average number of rooms per dwelling and per capita crime rate by town, thus it benefits for housing value assessment. Those two regression problems are typical examples in real-world applications aiming at a better and easier understanding of the inherent properties of a studied phenomenon.

In order to add outliers, training samples with various contamination rates are generated. This is made by assigning the random values from \([y_{\min}, y_{\max}]\) to the target values of some training samples. The user-specified parameter of ELM is the number of hidden nodes \(L\) and it is chosen from \([20, 30, 50, 80, 100, 200]\). According to Huang et al. \([5]\), only regularization parameter \(C\) needs to be specified if the regularized type of ELM has a relative large number of hidden nodes. Thus, the number of hidden nodes \(L\) in RELM, WRELM, ORELM is assigned with 200 in the experiments. As for the regularization parameter \(C\), it is chosen from \([2^{-10}, 2^{-8}, 2^{-6}, \ldots, 2^{10}]\). The final parameter setting (i.e., number of hidden nodes \(L\) and regularization parameter \(C\)), training time and testing results of RMSE are listed in Table 4. We can find that when there is no outlier in the training data, ORELM shows results of RMSE comparable to ELM, RELM and WRELM. More importantly, ORELM outperforms the other methods with the smallest testing RMSE when the contamination rate is high (e.g., 20% and 40%). This indicates that ORELM is more robust to the outliers, compared with ELM, RELM and WRELM.

### 4.2.2. Binary classification application

Disease diagnosis plays a valuable role in biomedical engineering. In some cases, wrong diagnosis is inevitable and this may bring negative impact to future diagnosis. Hence, there is a demand for a robust and stable method to cope with the cases.

In order to verify the robustness of ORELM for binary classification problems, we choose the breast cancer data set as an example. Note that in the experiments \(y_i\) is either 1 or –1 to indicate the “benign” class or the “malignant” class, respectively. Outliers which represent wrong diagnosis are generated by moving some training samples to the opposite class. For different algorithms and contamination rates, the best user-specified pair \((C, L)\) is chosen, where \(C\) and \(L\) share the same range in Section 4.2.1.

As shown in Table 5, WRELM does not get the expected results as it holds similar testing RMSE and accuracy of RELM. The

![Fig. 2. An illustration of results of ELM, RELM, WRELM and ORELM on training data contaminated by \([-1, 1]\) outliers (40 hidden nodes, 160 outliers) (left) and corresponding objective function value of ORELM versus number of iterations (right).](image)
underlying reason is that the weights of outliers present weak discriminating information. It can be interpreted through Fig. 3 which shows the training outputs of the four methods. Because of the presence of outliers and the mechanism of $\ell_2$-norm loss function, the outputs of RELM get close to zero, and thus the following weighting scheme of WRELM loses its effectiveness. Also, it can be inferred from the training outputs of ORELM in Fig. 3 that the training error $e$ is sparse. Back to Table 5, we can see that the performance of ORELM gives favorable results even when the contamination rate is 40%.

Corresponding to Fig. 3, Fig. 4 depicts the testing outputs of the four methods. As one can see, most of the ORELM outputs compactly surround the desired output, whereas the outputs of other three methods are far away from the desired output. According to Luo and Zhang [37], it means that ORELM shows more convincing prediction than ELM, RELM and WRELM.

Table 3
Specifications of data sets from UCI repository.

<table>
<thead>
<tr>
<th>Data sets</th>
<th># Samples Training</th>
<th># Samples Testing</th>
<th># Attributes</th>
<th>Range of targets values $[y_{\min}, y_{\max}]$</th>
<th>Associated task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abalone</td>
<td>2506</td>
<td>1671</td>
<td>8</td>
<td>$[1, 29]$</td>
<td>Regression</td>
</tr>
<tr>
<td>Housing</td>
<td>304</td>
<td>202</td>
<td>13</td>
<td>$[5, 50]$</td>
<td>Regression</td>
</tr>
<tr>
<td>Breast cancer</td>
<td>420</td>
<td>279</td>
<td>10</td>
<td>$[-1, 1]$</td>
<td>Regression and binary classification</td>
</tr>
</tbody>
</table>

Fig. 3. Training outputs of ELM, RELM, WRELM and ORELM (contamination rate: 40%).

Table 4
Performance of ELM, RELM, WRELM and ORELM (the best results are highlighted in bold).

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Algorithms (L, C)</th>
<th>Training time (s)</th>
<th>Contamination rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
| Abalone     | ELM (30, −)       | 0.0052            | 2.19 ± 0.10 | 2.26 ± 0.08 | 2.50 ± 0.07 | 3.20 ± 0.09 | Testing RMSE ± standard deviation
|             | RELM (200, $2^{19}$) | 0.0692       | 2.15 ± 0.06 | 2.23 ± 0.06 | 2.45 ± 0.07 | 3.15 ± 0.17 |
|             | WRELM (200, $2^{19}$) | 0.1253       | 2.17 ± 0.06 | 2.15 ± 0.05 | 2.27 ± 0.10 | 2.43 ± 0.19 |
|             | ORELM (200, $2^{14}$) | 0.1062       | 2.18 ± 0.07 | 2.16 ± 0.06 | 2.20 ± 0.07 | 2.22 ± 0.06 |
| Housing     | ELM (50, −)        | 0.0030            | 4.59 ± 0.53 | 5.10 ± 0.51 | 5.94 ± 0.74 | 7.31 ± 0.94 |
|             | RELM (200, $2^5$)  | 0.0310            | 3.64 ± 0.49 | 4.62 ± 0.52 | 5.20 ± 0.69 | 6.70 ± 0.71 |
|             | WRELM (200, $2^5$) | 0.0608            | 3.85 ± 0.65 | 4.12 ± 0.62 | 4.69 ± 0.66 | 5.91 ± 0.71 |
|             | ORELM (200, $2^5$) | 0.0354            | 3.89 ± 0.69 | 4.09 ± 0.45 | 4.27 ± 0.67 | 4.81 ± 0.58 |

Table 5
Performance of ELM, RELM, WRELM and ORELM on breast cancer data set (the best results are highlighted in bold).

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>(Training, Testing)</th>
<th>Contamination rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE</td>
<td>Acc.(%)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ELM</td>
<td>(0.31, 0.36)</td>
<td>(972, 96.1)</td>
</tr>
<tr>
<td>RELM</td>
<td>(0.26, 0.33)</td>
<td>(978, 96.4)</td>
</tr>
<tr>
<td>WRELM</td>
<td>(0.29, 0.33)</td>
<td>(96.9, 96.3)</td>
</tr>
<tr>
<td>ORELM</td>
<td>(0.29, 0.33)</td>
<td>(96.8, 96.2)</td>
</tr>
</tbody>
</table>
5. Conclusions and future work

In this paper, we have proposed an augmented Lagrange multiplier based ELM algorithm, namely ORELM, for solving the outlier-robust regression problems. By replacing the $\ell_2$-norm loss function with $\ell_1$-norm loss function, the proposed approach is preferable for handling data contaminated with outliers. Although solving the approach needs iterative solution, the utilization of ALM technique makes the iterations effective and efficient. Extensive experimental results on various classes of test problems with outliers show that, compared with other ELM-based algorithms, the proposed ORELM not only keeps the advantage of extremely fast training speed but also shows notable performance in dealing with outliers. Therefore, we argue that our approach provides a useful tool which may benefit state-of-the-art regression applications.

Throughout this paper, we do concentrate on outlier robustness which, however, is only one of the studies on ELM. In order to handle various practical applications, it is necessary to incorporate this study with some of the others for yielding a more useful method. Thus, our future work will focus on the following aspects: (a) combine with the evolutionary ensemble method proposed by Wang and Alhamdoosh [38] to improve the generalization and reliability; (b) construct a compact ELM network by taking the pruned-ELM [39] into consideration; and (c) seek a more robust model for imbalanced data processing in light of the work by Zong et al. [40].

Acknowledgments

The authors would like to thank the reviewers for their valuable comments and suggestions. This work is supported by the National Natural Science Foundation of China (No. 61273018) and the Natural Science Foundation of Zhejiang Province of China (No. Y1110651).

References


Y. Wang, F. Cao, Y. Yuan, A study on effectiveness of extreme learning machine, Neurocomputing 74 (2011) 2483–2490.


Kai Zhang received the M.Sc. degree in applied mathematics from China Jiliang University, Hangzhou, China. He is currently pursuing the Ph.D. degree from Harbin Institute of Technology, Harbin, China. His research interests include machine learning, fuzzy logic, image processing and pattern recognition.

Minxia Luo received the Ph.D. degree on Computer Science from Northwestern Polytechnical University, China. She is now a Professor at Mathematics Department of the China Jiliang University, China. Her research fields are fuzzy logic theory, universal logic theory and artificial neural networks. She has published over 90 papers in international and domestic peer reviewed journals.