Knowledge-Based Systems 215 (2021) 106652

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Robust multiclass least squares support vector classifier with optimal error distribution

Jiajun Ma [a,b], Shuisheng Zhou [a,*], Dong Li [a]

[a] School of Mathematics and Statistics, Xidian University, Xi'an, 710071, Shaanxi, China
[b] College of Mathematics and Computer Application, Shangluo University, Shangluo, 726000, Shaanxi, China

ARTICLE INFO

Article history:
Received 22 July 2020
Received in revised form 7 November 2020
Accepted 2 December 2020
Available online 4 December 2020

Keywords:
Outliers
Robust least squares support vector classifier
Error distribution
Multiclass classification

ABSTRACT

Robust least squares support vector regression (RLSSVR), minimizing the variance and mean of the global modeling errors, has achieved the excellent performance in dealing with outliers. However, generalizing the RLSSVR for solving the binary classification problems is easily misguided by the outliers because the differences in the modeling errors of the different classes are not considered. To address this issue, a robust least squares support vector classifier (RLSSVC) with optimal error distribution is proposed. RLSSVC minimizes the mean and variance of the modeling errors class-wisely, and considers the difference in the modeling errors of the different classes. Specifically, the binary classification problems are considered at first, the variance analysis indicates that the variance of the modeling errors of RLSSVC is smaller than that of RLSSVR. According to the validity in solving the binary classification problems, RLSSVC is naturally generalized for solving the multiclass classification problems by introducing multiple error adjusting factors. The robustness analysis provides a theoretical guarantee for the robustness of RLSSVC, which delivers that RLSSVC assigns the smaller weights for the training instances with the larger errors, while the larger weights for the training instances with the smaller errors. Furthermore, our optimization objective function is strictly convex and thus can obtain their corresponding closed-form solutions, resulting in higher computational performance. Finally, the performance of RLSSVC is further improved by introducing the metric learning and kernel trick. Theoretical and experimental results indicate that the proposed RLSSVC achieves the better classification effect with the lower computational costs.

1. Introduction

Least squares regression (LSR) [1–4] finds the optimal prediction function(s) for the training data by minimizing the squared errors, which has been widely used in machine learning because its formula is simple and easy to solve. Let the training set be $\boldsymbol{T} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)\}$, where $\boldsymbol{x}_i \in \Re^d$ is the training instance, and $\boldsymbol{y}_i$ is the corresponding target of $\boldsymbol{x}_i$. LSR is formulated as

$$\min_{\boldsymbol{W}} \frac{1}{2}\sum_{i=1}^{n} \|\boldsymbol{y}_i - \boldsymbol{W}^\top \boldsymbol{x}_i\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{W}\|_F^2, \tag{1}$$

where the first item of (1) is the empirical risk measured by the least squares loss, and the second one is the regularization part with a regularization parameter λ. For clarity, the bias in LSR is absorbed into the $\boldsymbol{W}^\top \boldsymbol{x}_i$ term.

According to the different attributes of $\boldsymbol{y}_i$, model (1) solves the following different problems:

- When $\boldsymbol{y}_i \in \Re$ is a continuous observation of $\boldsymbol{x}_i$, the model (1) solves a regression problem. Then, the prediction function for a test instance $\boldsymbol{x}$ becomes $f(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{x}$, where $\boldsymbol{W} \in \Re^d$ is the regression coefficient.
- When $\boldsymbol{y}_i \in \{+1, -1\}$ is a discrete label of $\boldsymbol{x}_i$, the model (1) solves the binary classification problem. Then, the decision function for a test instance $\boldsymbol{x}$ becomes $\boldsymbol{y} = sgn(\boldsymbol{W}^\top \boldsymbol{x})$, where $\boldsymbol{W} \in \Re^d$ is the hyperparameter of the decision boundary, and sgn$(\cdot)$ is a symbol function.
- When $\boldsymbol{y}_i$ is a c-dimensional binary codeword [5,6] for the label of $\boldsymbol{x}_i$, the model (1) solves the multiclass classification problem [5,7]. For a test instance $\boldsymbol{x}$, it is classified by $\text{argmax}_{k\in[c]}\, \boldsymbol{W}_{:,k}^\top \boldsymbol{x}$, where $\boldsymbol{W} \in \Re^{d\times c}$, and $\boldsymbol{W}_{:,k}$ is the kth column of $\boldsymbol{W}$.

For the regression and binary classification tasks, LSR (1) is equivalent to least squares support vector machine (LSSVM) [1,8].

In practice, the multiclass classification is becoming increasingly important in pattern recognition. In past decades, many multiclass classification methods based on support vector machine (SVM) [9] have been developed well. One way is decomposing the multiclass classification problems into a series of binary

* Corresponding author.
E-mail addresses: majiajun311@163.com (J. Ma), sszhou@mail.xidian.edu.cn (S. Zhou), lidong_xidian@foxmail.com (D. Li).

https://doi.org/10.1016/j.knosys.2020.106652
0950-7051/

classification problems using the one-vs-one (OVO) or one-vs-all (OVA) schemes [10]. Brunner [11] employed the pairwise SVM for handling the large scale multiclass classification problems. Allwein et al. [6] proposed a general method for combining the classifiers generated on the binary problems and proved a general empirical multiclass loss bound given the empirical loss of the individual binary learning algorithms. Liu et al. [12] used the Error-correcting output coding to transform the original multiclass classification problems into a series of binary classification problems, and mined the relationship between the binary classifiers. Takenouchi et al. [13] decomposed the original multiclass classification problems into the multiple binary classification problems based on the OVO method, and then decoded the outputs of the binary classifiers by minimization of weighted mixture of the Bregman divergence. Although these methods are intuitive, the OVO scheme can lead to a tie-in-vote problem and the OVA approach suffers from inconsistency when there is no dominant class [14]. Another way, considering all the classes simultaneously, has been proposed to overcome these drawbacks. As examples, Crammer et al. [15] and Tsochantaridis et al. [16] generalized the concept of margin for the multiclass problems and formulated the multiclass classification problems as a quadratic programming with constraints. Lee et al. [14] extended the SVM to the multiclass cases by devising a loss function with the suitable class codes. The detailed analysis and systematic comparison of the above multiclass classification methods based on SVM are provided in [17,18]. Subsequently, Xiang et al. [2] proposed a discriminative LSR (DLSR) model for the multiclass classification, which can be formulated as a single LSR model by using the ε-dragging technique. Zhang et al. [19] presented a retargeted LSR (ReLSR) model, which resets the regression target matrix and makes it have a large between-class margin. Wang et al. [20] proposed a margin scalable DLSR (MSDLSR) model, which improves the classification performance by minimizing the number of the support vectors of DLSR. Geng et al. [21] proposed a metric learning-guided least squares classifier (MLG-LSC), which learns a symmetric positive definite (SPD) metric matrix that yields the small distances for the LSR errors of the same class, while large ones for the LSR errors of the different classes. The above linear classification methods can be more expressive to deal with the more complex classification problems with the aid of metric learning [22,23], and can also effectively deal with the nonlinear classification problems by introducing the kernel trick [24]. Although the abovementioned algorithms have achieved great success, those are very sensitive to random noise [25]. For example, the label noise [26,27] generated by incorrectly labeling the training instances may mislead the learning of classifiers.

There are three types of approaches to improve the robustness of the model. One is assigning the different weights to each instance. Suykens et al. [28] proposed a weighted LSSVM (W-LSSVM) model, which reduces the negative influence of outliers by distributing the smaller weights to outliers. Liu et al. [29] presented the importance reweighting algorithms for classification with label noise by employing the inversed noise rates. Several other weight setting strategies are found in [30,31] and references therein. The theoretical analysis and experimental results show that those methods are very effective in dealing with outliers. Nevertheless, those methods need to solve LSSVM on the training data repeatedly, resulting in high computational complexity.

The second strategy is employing the robust surrogate loss functions. Ertekin et al. [32] and Ma et al. [33] proposed a non-convex online support vector machine algorithm based on the Ramp loss. Wang et al. [34] presented a robust LSSVM model, which employs a non-convex least squares loss function to suppress the influence of outliers. Yang et al. [35] and Chen et al. [36] gave the robust LSSVM (RLSSVM) with truncated least squares loss, which is illustrated to be more robust to outliers. Zhang et al. [37] proposed a robust angle-based multiclass SVM (RMSVM) model using truncated hinge loss and solved it with the difference convex (DC) algorithm [38], which shows the excellent performance in dealing with the outliers. However, solving the non-convex loss function is not only time-consuming but also requires more parameters to be preset [39].

Recently, Lu et al. [40] proposed a robust least squares support vector machine for regression (RLSSVR), which simultaneously minimizes the variance and mean of the global modeling errors (see Eq. (2)). The theoretical analysis and experimental results show that RLSSVR is less sensitive to outliers. The objective of RLSSVR is convex, which brings the easy-to-solve closed-form solutions and higher computational performance. Unfortunately, RLSSVR is tailored for regression. More exactly, the mean of the modeling errors of the different classes is considered to be equal when RLSSVR is directly used for classification, which results in the weakening of robustness.

In this paper, we attempt to construct a robust and simple multiclass classifier directly. With this intent, a novel robust least squares support vector classifier (RLSSVC) is proposed. Different from the existing algorithms, RLSSVC minimizes the variance and mean of the modeling errors of each class simultaneously, and can be formulated as a compact LSR model. According to the convexity of the objective of the RLSSVC, the closed-form solutions can be obtained, which brings the high computational performance. To the best of our knowledge, this is the first attempt to improve the robustness of the LSSVM for solving the classification problems by optimizing the distribution of the modeling errors for each class. Moreover, by introducing the geometric mean metric learning (GMML) [23] and kernel trick [24], RLSSVC can be further improved to solve the complex classification problems. The main contributions are summarized as follows.

- A novel robust least squares support vector classifier (RLSSVC) for the binary classification is proposed at first, which tries to enhance the robustness of classifier by minimizing the mean and variance of the modeling errors for each class. The theoretical analysis shows that the variance of the modeling errors of RLSSVC is smaller than that of RLSSVR in dealing with the binary classification problems.
- RLSSVC is generalized for solving the multiclass classification problems. The robustness analysis provides a theoretical guarantee for the robustness of RLSSVC, which delivers that RLSSVC assigns the smaller weights for the training instances with the larger errors, while the larger weights for the training instances with the smaller errors.
- RLSSVC is further improved for solving the complex classification problems with the aid of GMML and kernel.
- Experimental results verify our theoretical analysis, and illustrate that our method achieves the better classification effect with lower computational costs.

The rest of this paper is organized as follows. In Section 2, RLSSVR is to review briefly. In Section 3, RLSSVC is proposed and compared with RLSSVR in dealing with the binary classification problems. In Section 4, a framework of multiclass RLSSVC is proposed and robustness analysis is provided . The performance of RLSSVC is further improved by introducing the metric learning and kernel trick in Section 5. Section 6 includes several sets of experiments to demonstrate the effectiveness of the proposed method. The conclusion is finally given in Section 7.

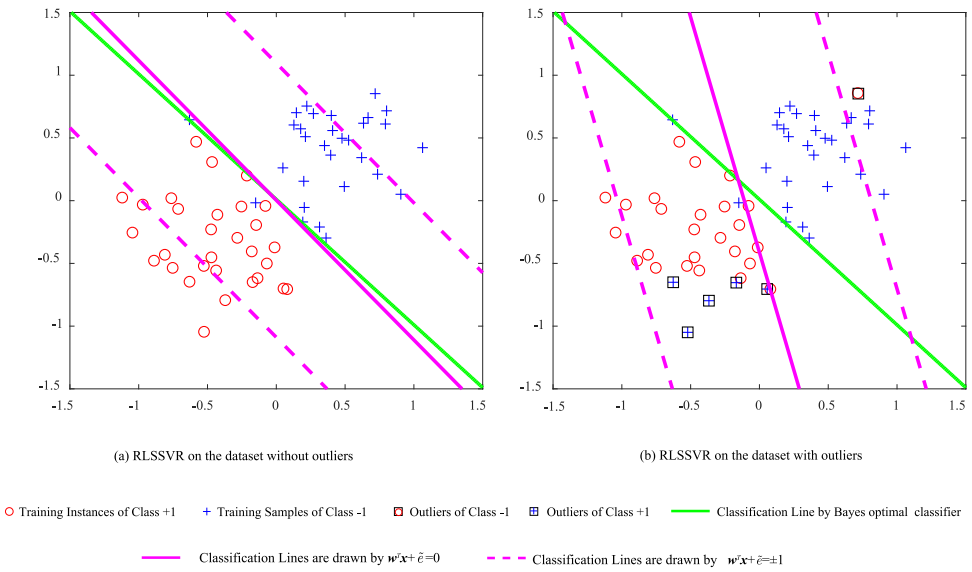

Fig. 1. Plots for the classification boundaries of RLSSVR on the binary classification dataset without and with outliers. The training dataset on the left panel has no outliers, whereas the training set on the right panel has six outliers simulated by labels flipping (five outliers marked as ⊞ in class +1, and one outlier marked as ⊡ in the class −1). The test accuracies of RLSSVR on the dataset without and with outliers are 95.71% and 91.86%, respectively.

2. Review of RLSSVR

Lu et al. [40] proposed RLSSVR based on minimizing the variance and mean of the global modeling errors to deal with the regression problems with outliers. Its regression model can be formulated as

$$\min_{\boldsymbol{w},\tilde{e}} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - \tilde{e})^2 + \frac{\lambda_1}{2}\|\boldsymbol{w}\|_2^2 + \frac{\lambda_2}{2}\tilde{e}^2, \tag{2}$$

where $y_i \in \Re$ is the target of $\boldsymbol{x}_i$, $\boldsymbol{w} \in \Re^d$ is the regression coefficient, $\tilde{e} \in \Re$ is a global error adjusting factor, λ_1 and λ_2 are the regularization parameters. According to the optimization conditions, $\tilde{e}$ is the scaled mean value of the modeling errors, and the scaling factor is $\frac{1}{1+\lambda_2}$. Therefore, optimizing the model (2) actually minimizes the mean and variance of the modeling errors simultaneously, thereby reducing the negative impact of outliers on the model. As a special case, when $\lambda_2 \to \infty$, model (2) degenerates to the LSR model.

As discussed in Section 1, when $y_i \in \{+1, -1\}$ is a discrete label of $\boldsymbol{x}_i$, RLSSVR is equivalent to the following binary classification model

$$\min_{\boldsymbol{w},\tilde{e}} \frac{1}{2n}\left(\sum_{\boldsymbol{y}_i=+1}(1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \tilde{e})^2 + \sum_{\boldsymbol{y}_i=-1}(-1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \tilde{e})^2\right) + \frac{\lambda_1}{2}\|\boldsymbol{w}\|_2^2 + \frac{\lambda_2}{2}\tilde{e}^2. \tag{3}$$

Model (3) employs only one error adjustment factor $\tilde{e}$ to minimize the variance and mean of the global modeling errors, and does not consider the difference between classes, which weakens the robustness of the classifier to outliers. A set of binary classification experiments were conducted on an artificial dataset that obeys the Gaussian distribution to show the influence of outliers on the RLSSVR , as shown in Fig. 1. Where the instances of class +1 and class −1 are drawn from $\mathcal{N}([-0.4, -0.4], [0.1\ \ 0; 0\ \ 0.1])$ and $\mathcal{N}([0.4, 0.4], [0.1\ \ 0; 0\ \ 0.1])$, respectively. The labels of six instances are flipped to simulate the outliers, which are marked as '⊞' and '⊡'. The Bayes optimal classifier [3] is chosen as the reference. Given the joint distribution $\mathcal{P}(\boldsymbol{X}, Y)$ for the data, the Bayes optimal classifier is defined as $h_{Bayes}(\boldsymbol{x}) = \text{argmax}_{y\in\{1,2,\ldots,c\}}\, \mathcal{P}[Y = y|X = \boldsymbol{x}]$, where c is the number of the classes, and the posterior probability $\mathcal{P}[Y = y|X = \boldsymbol{x}]$ is calculated based on the $\mathcal{P}(\boldsymbol{X}, Y)$.

As shown in Fig. 1(a), the classification performance of RLSSVR on the dataset without outliers is comparable to the Bayes optimal classifier. Actually, on the clean dataset, the mean of the modeling errors is 0, and there is almost no difference between the modeling errors of class +1 and class −1. This makes the RLSSVR equivalent to the LSSVM and can obtain the better classification results. In contrast, as shown in Fig. 1(b), the classification boundary of RLSSVR on the dataset with outliers is seriously deviated from the original position in Fig. 1(a), and the prediction accuracy deteriorates. This is because the appearance of outliers will cause the oscillations and differences in modeling errors of the different classes, while RLSSVR only introduces one error adjustment factor, ignoring the differences between the classes. Therefore, it is desirable to reduce the error oscillation for such outliers in RLSSVR by minimizing the mean and variance of the modeling errors for each class.

3. RLSSVC for binary classification

In this section, the binary classification problems will be considered. By introducing two error adjustment factors to minimize the variance and mean of the modeling errors for each class, a robust least squares support vector classifier with optimal error distribution is proposed. This new model is built in the following formula

$$\min_{\boldsymbol{w},\tilde{e}_1,\tilde{e}_2} \frac{1}{2n}\left(\sum_{\boldsymbol{y}_i=+1}(1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \tilde{e}_1)^2 + \sum_{\boldsymbol{y}_i=-1}(-1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \tilde{e}_2)^2\right) + \frac{\lambda_1}{2}\|\boldsymbol{w}\|_2^2 + \frac{\lambda_2}{2}\left(\frac{n_1}{n}\tilde{e}_1^2 + \frac{n_2}{n}\tilde{e}_2^2\right), \tag{4}$$

where $\tilde{e}_1 \in \Re$ and $\tilde{e}_2 \in \Re$ are the error adjustment factors for class +1 and class −1, n_1 and n_2 are the number of instances in class +1 and class −1. Fig. 2 illustrates the learning dynamics induced by the model (4).

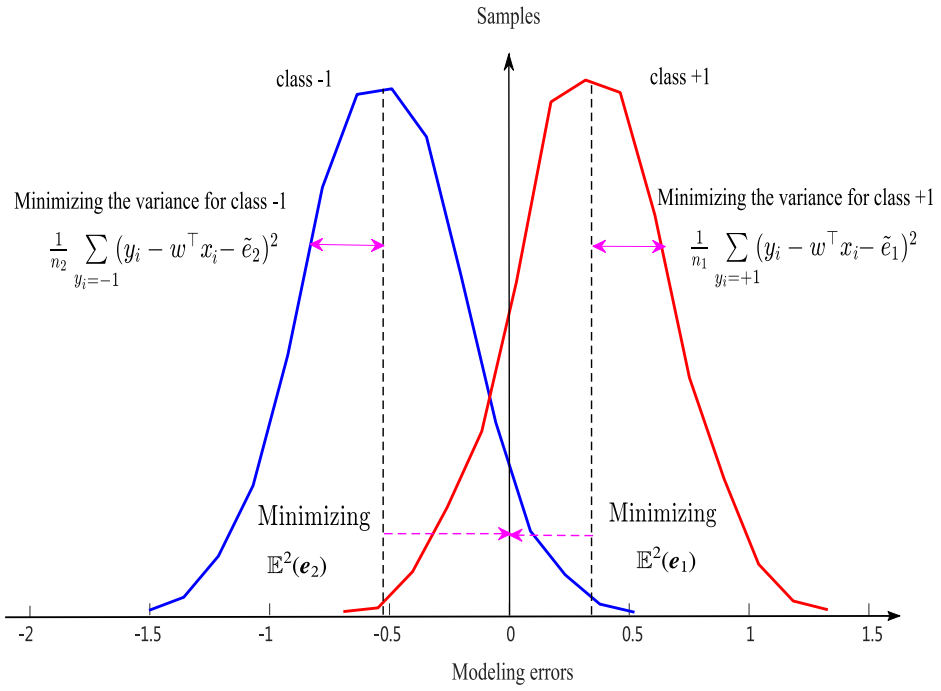

Fig. 2. Illustration of the RLSSVC model for the binary classification problem. Here $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ record the modeling errors for class +1 and class -1, $\tilde{e}_1$ and $\tilde{e}_2$ are the error adjustment factors for class +1 and class -1, $\mathbb{E}(\boldsymbol{e}_1)$ and $\mathbb{E}(\boldsymbol{e}_2)$ are the mean of the modeling errors for class +1 and class -1. As shown in the figure, the mean values of the modeling errors of the class +1 and the class -1 are significantly different. Therefore, unlike RLSSVR, we minimize the mean and variance of the modeling errors for each class.

Because the objective of model (4) is convex, its closed-form solutions can be easily obtained through the optimality conditions.

$$\begin{cases} \boldsymbol{w} = (n\lambda_1 \boldsymbol{I}_d + \sum_{j=1}^{2} \boldsymbol{X}_j{}^\top \boldsymbol{L}_j \boldsymbol{X}_j)^{-1} \sum_{j=1}^{2} \boldsymbol{X}_j{}^\top \boldsymbol{L}_j \boldsymbol{y}_j \\ \tilde{e}_j = \dfrac{1}{(1+\lambda_2)n_j} \boldsymbol{1}_{n_j}^\top \boldsymbol{e}_j, \quad j = 1, 2 \end{cases} \tag{5}$$

where $\boldsymbol{I}_d \in \Re^{d\times d}$ is an identity matrix, $\boldsymbol{X}_1 \in \Re^{n_1\times d}$ and $\boldsymbol{X}_2 \in \Re^{n_2\times d}$ denote the instances of class +1 and class -1, $\boldsymbol{L}_j = \boldsymbol{I}_{n_j} - \frac{1}{n_j(1+\lambda_2)} \boldsymbol{1}_{n_j} \boldsymbol{1}_{n_j}^\top$, $\boldsymbol{1}_{n_j}$ denotes a n_j-dimensional column vector of ones, $\boldsymbol{y}_1 = [+1, +1, \ldots, +1]^\top \in \Re^{n_1}$, $\boldsymbol{y}_2 = [-1, -1, \ldots, -1]^\top \in \Re^{n_2}$, and $\boldsymbol{e}_j = \boldsymbol{y}_j - \boldsymbol{X}_j \boldsymbol{w}$ records the modeling errors of class j.

To estimate the robustness of the model (4), experimental studies are carried out on the dataset in Fig. 1, applying the model (4), and the results are shown in Fig. 3. From the comparison of Fig. 3(a) and (b), it can be seen that the classification boundary of the RLSSVC is almost unchanged, and its accuracy remains stable before and after adding outliers. Although the outliers cause the significant differences between the error distributions of class +1 and class -1, RLSSVC minimizes the variance and mean of the modeling errors class by class, effectively reducing the negative impact of outliers, but RLSSVR fails.

The above experiments intuitively demonstrate the superiority of RLSSVC in dealing with the classification problems with outliers. Next, we will theoretically prove that the variance of the modeling errors for RLSSVC is smaller than that of RLSSVR in dealing with the binary classification problems.

Proposition 3.1. *Solving model (4) actually minimizes the variance and mean of modeling errors for each class. When $\lambda_2 \to \infty$, the model (4) will degenerate to the LSR model.*

Proof. Let $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ record the modeling errors of class +1 and class -1, respectively. We have $\tilde{e}_1 = \frac{1}{1+\lambda_2}\mathbb{E}(\boldsymbol{e}_1)$ is the scaled mean of modeling errors of class +1, and $\tilde{e}_2 = \frac{1}{1+\lambda_2}\mathbb{E}(\boldsymbol{e}_2)$ is the scaled mean of modeling errors of class -1. The first item in model(4) can be relisted as

$$\begin{aligned} &\frac{1}{n}\left(\sum_{\boldsymbol{y}_i=+1}(1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \tilde{e}_1)^2 + \sum_{\boldsymbol{y}_i=-1}(-1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \tilde{e}_2)^2\right) \\ &= \frac{1}{n}\left(\sum_{\boldsymbol{y}_i=+1}(1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \mathbb{E}(\boldsymbol{e}_1) + \frac{\lambda_2}{1+\lambda_2}\mathbb{E}(\boldsymbol{e}_1))^2\right. \\ &\quad \left. + \sum_{\boldsymbol{y}_i=-1}(-1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \mathbb{E}(\boldsymbol{e}_2) + \frac{\lambda_2}{1+\lambda_2}\mathbb{E}(\boldsymbol{e}_2))^2\right) \\ &= \frac{1}{n}\left(\sum_{\boldsymbol{y}_i=+1}(1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \mathbb{E}(\boldsymbol{e}_1))^2 + \sum_{\boldsymbol{y}_i=-1}(-1 - \boldsymbol{w}^\top \boldsymbol{x}_i - \mathbb{E}(\boldsymbol{e}_2))^2\right. \\ &\quad \left. + n_1(\frac{\lambda_2}{1+\lambda_2})^2\mathbb{E}^2(\boldsymbol{e}_1) + n_2(\frac{\lambda_2}{1+\lambda_2})^2\mathbb{E}^2(\boldsymbol{e}_2)\right) \\ &= \frac{n_1}{n}\mathbb{D}(\boldsymbol{e}_1) + \frac{n_2}{n}\mathbb{D}(\boldsymbol{e}_2) + \frac{n_1\lambda_2^2}{n(1+\lambda_2)^2}\mathbb{E}^2(\boldsymbol{e}_1) + \frac{n_2\lambda_2^2}{n(1+\lambda_2)^2}\mathbb{E}^2(\boldsymbol{e}_2), \end{aligned} \tag{6}$$

where $\mathbb{D}(\boldsymbol{e}_j)$ denotes the variance of the modeling errors of class j. Therefore, the first and third terms of model (4) minimize the variance and mean of the modeling errors for each class. Moreover, when $\lambda_2 \to \infty$, $\tilde{e}_1 \to 0$ and $\tilde{e}_2 \to 0$, the model (4) degenerates to the LSR. □

Theorem 3.2. *The variance of the modeling errors of RLSSVC (the first term in model (4)) is smaller than that of RLSSVR (the first term in model (3)), and they are equal if and only if the mean of the modeling errors of class +1 and class -1 are equal.*

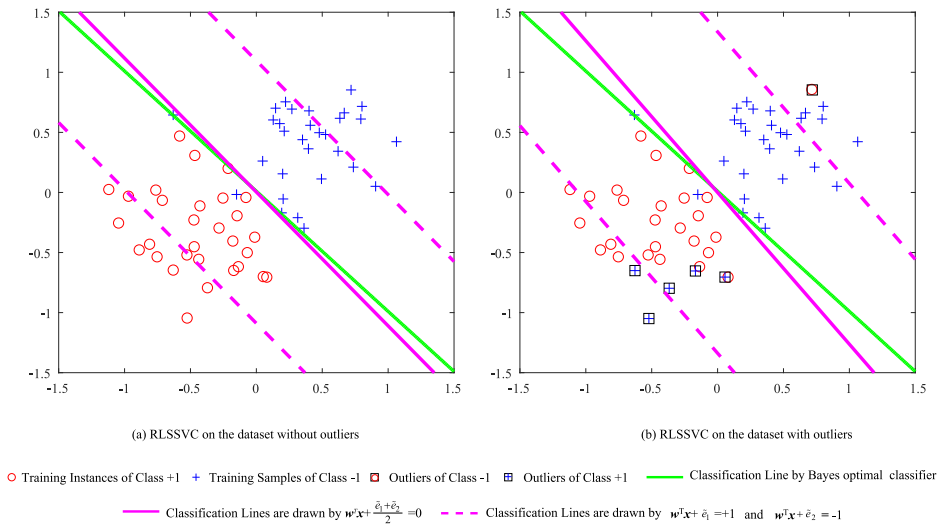

Fig. 3. Plots of the classification boundaries of RLSSVC on the binary classification dataset with and without outliers. The training dataset on the left panel has no outliers, whereas the training set on the right panel has six outliers simulated by labels flipping (five outliers marked as ⊞ in class +1, and one outlier marked as ⧈ in the class −1). The test accuracies of RLSSVC on the dataset without and with outliers are 95.71% and 95.43%, respectively.

Proof. The variance term $\mathbb{D}(\boldsymbol{e})$ of the global modeling errors in model (3) can be rewritten as

$$\begin{aligned}
\mathbb{D}(\boldsymbol{e}) &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i - \tilde{e})^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 - 2\tilde{e}\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i) + \frac{1}{n}\sum_{i=1}^{n}\tilde{e}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 - \frac{1+2\lambda_2}{(1+\lambda_2)^2}\left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)\right)^2 \\
&= \frac{1}{n}\left(\sum_{y_i=+1}^{n}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 + \sum_{y_i=-1}^{n}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2\right) \\
&\quad - \frac{1+2\lambda_2}{(1+\lambda_2)^2}\left(\frac{n_1\mathbb{E}(\boldsymbol{e}_1) + n_2\mathbb{E}(\boldsymbol{e}_2)}{n}\right)^2 \\
&= \frac{n_1\mathbb{D}(\boldsymbol{e}_1) + n_1\mathbb{E}^2(\boldsymbol{e}_1) + n_2\mathbb{D}(\boldsymbol{e}_2) + n_1\mathbb{E}^2(\boldsymbol{e}_2)}{n} \\
&\quad - \frac{1+2\lambda_2}{(1+\lambda_2)^2}\left(\frac{n_1\mathbb{E}(\boldsymbol{e}_1) + n_2\mathbb{E}(\boldsymbol{e}_2)}{n}\right)^2 \\
&= \frac{n_1}{n}\mathbb{D}(\boldsymbol{e}_1) + \frac{n_2}{n}\mathbb{D}(\boldsymbol{e}_2) + \frac{n_1\lambda_2^2}{n(1+\lambda_2)^2}\mathbb{E}^2(\boldsymbol{e}_1) \\
&\quad + \frac{n_2\lambda_2^2}{n(1+\lambda_2)^2}\mathbb{E}^2(\boldsymbol{e}_2) + \frac{(1+2\lambda_2)n_1n_2}{(1+\lambda_2)^2n^2}\left(\mathbb{E}(\boldsymbol{e}_1) - \mathbb{E}(\boldsymbol{e}_2)\right)^2 \qquad (7)
\end{aligned}$$

where $y_i \in \{+1, -1\}$ is the label of $\boldsymbol{x}_i$. Combining (7) with (6), the proof is provided. □

For the dataset with outliers in Fig. 1(b), the mean of the modeling errors for class +1 are greater than those for class −1, that is, $\mathbb{E}(\boldsymbol{e}_1) \gg \mathbb{E}(\boldsymbol{e}_2)$, which causes the variance term in model (3) is much larger than that in model (4). Therefore, for the classification problems with outliers, it is more effective to minimize the mean and variance of the modeling errors for each class than to minimize the mean and variance of the global modeling errors.

4. RLSSVC for multiclass classification

Motivated by the effectiveness of RLSSVC in solving the binary classification problems, a multiclass RLSSVC learning framework is proposed and some theoretical analysis is carried out in this section. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]^\top \in \Re^{n\times d}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n]^\top \in \Re^{n\times c}$ denote the instance matrix and corresponding label matrix, where c is the number of classes. Here, -1 or +1 is used to denote the regression label for each instance. For example, if $\boldsymbol{x}_i$ belongs to the class j, its label is defined as $\boldsymbol{y}_i = [-1, -1, \ldots, +1, \ldots, -1, -1]^\top$ with only the jth element equal to +1. For clarity, $\boldsymbol{X}_j = [\boldsymbol{x}_{j1}, \ldots, \boldsymbol{x}_{jn_j}]^\top$ and $\boldsymbol{Y}_j = [\boldsymbol{y}_{j1}, \ldots, \boldsymbol{y}_{jn_j}]^\top$ are employed to record the instance from class j and their corresponding labels, where n_j is the number of training instances in class j. The instances and their corresponding labels that are not in the class j are recorded as $\boldsymbol{X}_{\bar{j}}$ and $\boldsymbol{Y}_{\bar{j}}$, respectively.

4.1. Model construction and optimization

For a c-class classification problem, we introduce c error adjustment factor vectors to minimize the variance and mean of the modeling errors for each class. As a result, we have the following multiclass RLSSVC model:

$$\min_{\boldsymbol{W},\tilde{\boldsymbol{e}}_j} \frac{1}{2n}\sum_{j=1}^{c}\|\boldsymbol{Y}_j - \boldsymbol{X}_j\boldsymbol{W} - \boldsymbol{1}_{n_j}\tilde{\boldsymbol{e}}_j^\top\|_F^2 + \frac{\lambda_1}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_2}{2}\sum_{j=1}^{c}\frac{n_j}{n}\|\tilde{\boldsymbol{e}}_j\|_2^2, \qquad (8)$$

where $\tilde{\boldsymbol{e}}_j \in \Re^{c\times 1}$ is the error adjustment factor vector for class j, λ_1 and λ_2 are the regularization parameters. With model (8), the first-order statistics (mean value) and second-order statistics (variance) are employed to characterize the distribution of the modeling errors for each class, so as to the distribution of the modeling errors for each class is optimal. The objective of model (8) is convex, its closed-form solutions can be easily obtained through the optimality conditions, as follows

$$\begin{cases}
\boldsymbol{W} = \left(n\lambda_1\boldsymbol{I}_d + \sum_{j=1}^{c}\boldsymbol{X}_j^\top\boldsymbol{L}_j\boldsymbol{X}_j\right)^{-1}\sum_{j=1}^{c}\boldsymbol{X}_j^\top\boldsymbol{L}_j\boldsymbol{Y}_j, \\
\tilde{\boldsymbol{e}}_j^\top = \frac{1}{n_j(1+\lambda_2)}\boldsymbol{1}_{n_j}^\top(\boldsymbol{Y}_j - \boldsymbol{X}_j\boldsymbol{W}), \quad j \in [c],
\end{cases} \qquad (9)$$

where $\boldsymbol{L}_j = \boldsymbol{I}_{n_j} - \frac{1}{n_j(1+\lambda_2)}\boldsymbol{1}_{n_j}\boldsymbol{1}_{n_j}^\top$. To reduce the costs of calculating the inverse of $\boldsymbol{H} = n\lambda_1\boldsymbol{I}_d + \sum_{j=1}^{c}\boldsymbol{X}_j^\top\boldsymbol{L}_j\boldsymbol{X}_j$ in the cases of $n \ll d$,

Sherman–Morrison-Woodbury (SMW) identity [41] can be used to simplify the calculation of $\boldsymbol{H}^{-1}$, as described below

$$\boldsymbol{H}^{-1} = \frac{1}{n\lambda_1}\boldsymbol{I}_d - \frac{1}{n\lambda_1}\boldsymbol{X}^\top(n\lambda_1\boldsymbol{L}^{-1} + \boldsymbol{X}\boldsymbol{X}^\top)^{-1}\boldsymbol{X},$$

here $\boldsymbol{L} = diag(\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots, \boldsymbol{L}_c)$, and $\boldsymbol{L}_j^{-1}$ can be further simplified by the SMW identity.

In the optimization process, the model (8) enjoys the following four valuable features:

- $\tilde{\boldsymbol{e}}_j$ is actually the scaled mean of the modeling errors for class j, which will be proved in Section 4.2. Thus, minimizing the first term $\frac{1}{2n}\sum_{j=1}^{c}\|\boldsymbol{Y}_j - \boldsymbol{X}_j\boldsymbol{W} - \boldsymbol{1}_{n_j}\tilde{\boldsymbol{e}}_j^\top\|_F^2$ will optimize the variance of the modeling errors for each class, thereby reducing the error oscillations caused by the outliers and enhancing the stability of the model.
- The second regularization term $\frac{\lambda_1}{2}\|\boldsymbol{W}\|_F^2$ is introduced to relieve overfitting by imposing prior constraints on $\boldsymbol{W}$.
- Minimizing the third term $\frac{\lambda_2}{2}\sum_{j=1}^{c}\frac{n_j}{n}\|\tilde{\boldsymbol{e}}_j\|_2^2$ will minimize the mean of the modeling errors for each class. Hence, it can improve the classification accuracy.
- When $c = 2$, that is, the regression label for $\boldsymbol{x}_i$ is $[+1\ -\ 1]^\top$ or $[-1\ +\ 1]^\top$, the sum of the two columns of the optimal solution $\boldsymbol{W}$ of the model (8) is $\boldsymbol{0}$, which leads to the model (8) is equivalent to the model (4) in dealing with the binary classification problems. In short, model (8) is a natural generalization of model (4) for solving the multiclass classification problems, inheriting the good properties of the model (4).

Based on the learned optimal $\boldsymbol{W}$ and $\tilde{\boldsymbol{e}}_j$, each test instance $\boldsymbol{x}$ is classified by

$$\underset{k\in[c]}{\operatorname{argmin}}(\boldsymbol{y}^k - \boldsymbol{W}^\top\boldsymbol{x} - \tilde{\boldsymbol{e}}_k)^\top(\boldsymbol{y}^k - \boldsymbol{W}^\top\boldsymbol{x} - \tilde{\boldsymbol{e}}_k), \tag{10}$$

where $\boldsymbol{y}^k$ denotes a column vector whose kth element is +1 and the rest of elements is -1.

4.2. Robustness analysis

In this subsection, a robust analysis for RLSSVC from the perspective of the contribution weights of the instances is proposed. It can be first proven that the optimal error adjustment factor $\tilde{\boldsymbol{e}}_j$ is the scaled mean of the modeling errors for class j.

Proposition 4.1. *The error adjustment factor vector $\tilde{\boldsymbol{e}}_j$ is the scaled mean of the modeling errors of class j.*

Proof. Based on the optimality conditions for model (8), we have

$$\begin{aligned}\tilde{\boldsymbol{e}}_j^\top &= \frac{1}{n_j(1+\lambda_2)}\boldsymbol{1}_{n_j}^\top(\boldsymbol{Y}_j - \boldsymbol{X}_j\boldsymbol{W})\\ &= \frac{1}{(1+\lambda_2)}\mathbb{E}(\boldsymbol{E}_j), \quad j \in [c],\end{aligned}$$

where $\boldsymbol{E}_j = \boldsymbol{Y}_j - \boldsymbol{X}_j\boldsymbol{W}$ records the modeling errors of class j, and $\lambda_2 \geq 0$ is the regularization parameter. Therefore, the error adjustment factor $\tilde{\boldsymbol{e}}_j$ is the scaled mean of the modeling errors of class j. □

From Proposition 4.1, it can be concluded that a multiclass regression model with minimization of the mean and variance of the modeling errors for each class can be obtained through optimizing the model (8). Now, we prove that compared to LSR, RLSSVC assigns the smaller weights for the training instances with the larger errors, while the larger weights for the training instances with the smaller errors.

Theorem 4.2. *Compared with LSR, in the optimization process for the RLSSVC model, the normal instances are assigned the larger contribution weights, while the outliers are assigned the smaller contribution weights.*

Proof. Let $\boldsymbol{x}_{jk}$ be the kth instance in class j, $\boldsymbol{e}_{jk}$ is the modeling error of $\boldsymbol{x}_{jk}$, and $\boldsymbol{E}_j = [\boldsymbol{e}_{j1}, \ldots, \boldsymbol{e}_{jk}, \ldots, \boldsymbol{e}_{jn_j}]$ records the modeling errors of the instances from class j. For simplicity, it is assumed that an instance $\boldsymbol{x}_{jk}$ satisfying $\|\boldsymbol{e}_{jk}\|_2 \leq 0.5\|\tilde{\boldsymbol{e}}_j\|_2$ is regarded as a normal instance, otherwise it is regarded as an outlier. The proof for [40, Theorem 1] is adapted to complete the proof of Theorem 4.2. The contribution weights of $\boldsymbol{x}_{jk}$ in LSR and RLSSVC are defined as

$$C_{LSR}^{jk} = \frac{\|\boldsymbol{e}_{jk}\|_2^2}{\sum_{i=1}^{n_j}\|\boldsymbol{e}_{ji}\|_2^2}, \tag{11}$$

and

$$C_{RLSSVC}^{jk} = \frac{\|\boldsymbol{e}_{jk} - \tilde{\boldsymbol{e}}_j\|_2^2 + \lambda_2\|\tilde{\boldsymbol{e}}_j\|_2^2}{\sum_{i=1}^{n_j}\|\boldsymbol{e}_{ji} - \tilde{\boldsymbol{e}}_j\|_2^2 + \lambda_2 n_j\|\tilde{\boldsymbol{e}}_j\|_2^2}. \tag{12}$$

To compare the contribution weights of $\boldsymbol{x}_{jk}$ in LSR and RLSSVC, the difference between C_{LSR}^{jk} and C_{RLSSVC}^{jk} is defined as $\triangle^{jk} = C_{RLSSVC}^{jk} - C_{LSR}^{jk}$. By substituting (11) and (12) into $\triangle^{jk}$, we have

$$\triangle^{jk} = \frac{(\mu\|\tilde{\boldsymbol{e}}_j\|_2^2 - 2\tilde{\boldsymbol{e}}_j^\mathrm{T}\boldsymbol{e}_{jk})\sum_{i=1}^{n_j}\|\boldsymbol{e}_{ji}\|_2^2 + n_j\mu\|\tilde{\boldsymbol{e}}_j\|_2^2\|\boldsymbol{e}_{jk}\|_2^2}{\sum_{i=1}^{n_j}\|\boldsymbol{e}_{ji}\|_2^2(\sum_{i=1}^{n_j}\|\boldsymbol{e}_{ji} - \tilde{\boldsymbol{e}}_j\|_2^2 + \lambda_2 n_j\|\tilde{\boldsymbol{e}}_j\|_2^2)}, \tag{13}$$

where $\mu = 1 + \lambda_2$. When $\|\boldsymbol{e}_{jk}\|_2 \leq 0.5\|\tilde{\boldsymbol{e}}_j\|_2$, the numerator on the right side of (13) can be derived as

$$\begin{aligned}&(\mu\|\tilde{\boldsymbol{e}}_j\|_2^2 - 2\tilde{\boldsymbol{e}}_j^\mathrm{T}\boldsymbol{e}_{jk})\sum_{i=1}^{n_j}\|\boldsymbol{e}_{ji}\|_2^2 + n_j\mu\|\tilde{\boldsymbol{e}}_j\|_2^2\|\boldsymbol{e}_{jk}\|_2^2\\ &\geq \lambda_2\|\tilde{\boldsymbol{e}}_j\|_2^2\sum_{i=1}^{n_j}\|\boldsymbol{e}_{ji}\|_2^2 + n_j(1+\lambda_2)\|\tilde{\boldsymbol{e}}_j\|_2^2\|\boldsymbol{e}_{jk}\|_2^2\\ &\geq 0.\end{aligned} \tag{14}$$

This means that for a normal instance, the contribution weight assigned by RLSSVC is greater than that assigned by LSR. On the other hand, the training instances of class j satisfy $\sum_{k=1}^{n_j}C_{LSR}^{jk} = \sum_{k=1}^{n_j}C_{RLSSVC}^{jk} = 1$. Thus, we have

$$\begin{aligned}\sum_{\|\boldsymbol{e}_{jk}\|_2\leq 0.5\|\tilde{\boldsymbol{e}}_j\|_2} C_{RLSSVC}^{jk} &\geq \sum_{\|\boldsymbol{e}_{jk}\|_2\leq 0.5\|\tilde{\boldsymbol{e}}_j\|_2} C_{LSR}^{jk}\\ \sum_{\|\boldsymbol{e}_{jk}\|_2> 0.5\|\tilde{\boldsymbol{e}}_j\|_2} C_{RLSSVC}^{jk} &< \sum_{\|\boldsymbol{e}_{jk}\|_2> 0.5\|\tilde{\boldsymbol{e}}_j\|_2} C_{LSR}^{jk}.\end{aligned} \tag{15}$$

□

Theorem 4.2 shows that compared with LSR, RLSSVC gives the smaller weights to the training instances with the larger errors ($\|\boldsymbol{e}_{jk}\|_2 > 0.5\|\tilde{\boldsymbol{e}}_j\|_2$), and gives the larger weights to the training instances with the smaller errors ($\|\boldsymbol{e}_{jk}\|_2 \leq 0.5\|\tilde{\boldsymbol{e}}_j\|_2$). From the above theoretical analysis, it can be concluded that the proposed RLSSVC can reduce the negative impact of outliers on the classifier to some extent.

4.3. Complexity analysis

The optimization processes for DSLR, MSDSLR and ResLSR are very similar, and their main computational costs come from the matrix inversion and matrix multiplication. The closed-form solutions of RLSSVC can be obtained by the optimality conditions. In Eq. (9), the computational complexity for solving the optimal $\boldsymbol{W}$ and $\boldsymbol{e}_j$ are $O\left(d^3 + nd^2 + ncd\right)$ and $O(n_j dc)$, respectively. In summary, the total computational complexity of RLSSVC is

Table 1
Computational complexity of the comparison methods.

Method	Computational complexity
DLSR [2]	$O\left(d^3+2nd^2+T\times 2ndc\right)$
RMSVM [37]	$O\left(T\times(c^3n^2d+t\times c^2n^2)\right)$
RLSSVM [35]	$O\left(T\times cn^3\right)$
MSDLSR [20]	$O\left(d^3+2nd^2+T\times 2ndc\right)$
ReLSR [19]	$O\left(d^3+2nd^2+T\times 2nc2d\right)$
MLG-LSC [21]	$O(d^3+2nd^2+2ndc+2nc^2)$
RLSSVC	$O\left(d^3+nd^2+2ncd\right)$

$O\left(d^3+nd^2+2ncd\right)$. The formulation of RMSVM involves a non-differentiable non-convex optimization problem with nc constraints. The authors therefore use the DC algorithm with the coordinate descent method. Where the computational complexity of each outer iteration for DC algorithm is $O(c^3n^2d)$ and the computational complexity of each inner iteration for coordinate descent algorithm is $O(c^2n^2)$. The objective of RLSSVM is neither differentiable nor convex, the DC algorithm is employed, and from [35], the computational complexity of RLSSVM is $O(cn^3)$ for each iteration. Let T be the iterative number of the DLSR, RMSVM, RLSSVM, MSDLSR and ReLSR, and t be the iterative number for solving the subproblems in RMSVM, then the computational complexity of these methods can be compared in detail, as shown in Table 1. It is note that the number of iterations T for each method may be different.

5. Improvement of RLSSVC based on metric learning and kernel

In this section, RLSSVC is further improved by using the Geometric mean metric learning (GMML) [23] and kernel method [24]. Specifically, GMML can be used as a simple and effective post-processing for RLSSVC, meanwhile the kernel trick can assist RLSSVC to deal with the nonlinear classification problems.

5.1. RLSSVC with metric learning

Motivated by MLG-LSC [21], we learn a SPD metric matrix $\boldsymbol{M}$ for the centralized modeling errors of RLSSVC such that matrix $\boldsymbol{M}$ can yield small distances for the same class, while large ones for the different classes. After obtaining the optimal $\boldsymbol{W}$ and $\tilde{\boldsymbol{e}}_j$ through Eq. (9), the centralized modeling errors of the class j and the centralized modeling errors excluding the class j are calculated as follows

$$\begin{aligned}\boldsymbol{C}_j &= \boldsymbol{Y}_j-\boldsymbol{X}_j\boldsymbol{W}-\frac{1}{2}(\boldsymbol{Y}_j+\boldsymbol{1}_{n_j}\boldsymbol{1}_c^\top)\tilde{\boldsymbol{E}},\\ \boldsymbol{C}_{\bar{j}} &= \boldsymbol{Y}_{\bar{j}}-\boldsymbol{X}_{\bar{j}}\boldsymbol{W}-\frac{1}{2}(\boldsymbol{Y}_{\bar{j}}+\boldsymbol{1}_{n_{\bar{j}}}\boldsymbol{1}_c^\top)\tilde{\boldsymbol{E}},\quad j\in[c],\end{aligned}\tag{16}$$

where $\tilde{\boldsymbol{E}}=[\tilde{\boldsymbol{e}}_1,\tilde{\boldsymbol{e}}_2,\ldots,\tilde{\boldsymbol{e}}_c]^\top\in\Re^{c\times c}$, $n_{\bar{j}}=n-n_j$, $\boldsymbol{C}_j$ and $\boldsymbol{C}_{\bar{j}}$ record the centralized modeling errors for the instances included and excluded in the class j, respectively.

Then, a SPD matrix $\boldsymbol{M}$ for the centralized errors is learned by the following optimization objective

$$\min_{\boldsymbol{M}\succ 0}(1-\alpha)\rho_R^2(\boldsymbol{M},\boldsymbol{S}^{-1})+\alpha\rho_R^2(\boldsymbol{M},\boldsymbol{D}),\tag{17}$$

where $\alpha\in[0,1]$ is a parameter that determines the balance, $\boldsymbol{S}=\sum_{j=1}^c\boldsymbol{C}_j^\top\boldsymbol{C}_j$ and $\boldsymbol{D}=\sum_{j=1}^c\boldsymbol{C}_{\bar{j}}^\top\boldsymbol{C}_{\bar{j}}$, ρ_R stands for the Riemannian distance between two SPD matrices, which is defined as follows

$$\rho_R(\boldsymbol{X},\boldsymbol{Z}):=\|\log(\boldsymbol{Z}^{-1/2}\boldsymbol{X}\boldsymbol{Z}^{-1/2})\|_F\quad for\ \ \boldsymbol{X},\boldsymbol{Z}\succ 0$$

As discussed in [21], the closed-form solutions of (17) is obtained:

$$\boldsymbol{M}=\boldsymbol{S}^{-1/2}(\boldsymbol{S}^{1/2}\boldsymbol{D}\boldsymbol{S}^{1/2})^\alpha\boldsymbol{S}^{-1/2}.\tag{18}$$

Based on the learned optimal $\boldsymbol{W}$, $\tilde{\boldsymbol{e}}_j$ and $\boldsymbol{M}$, each test instance $\boldsymbol{x}$ is classified by

$$\operatorname*{argmin}_{k\in[c]}(\boldsymbol{y}^k-\boldsymbol{W}^\top\boldsymbol{x}-\tilde{\boldsymbol{e}}_k)^\top\boldsymbol{M}(\boldsymbol{y}^k-\boldsymbol{W}^\top\boldsymbol{x}-\tilde{\boldsymbol{e}}_k),\tag{19}$$

where $\boldsymbol{y}^k$ denotes a column vector whose kth element is +1, and the rest of elements is -1.

5.2. Scalable kernel RLSSVC

Another improvement of RLSSVC is to solve the nonlinear classification problems by introducing kernel trick [24]. In kernel method, the original instance $\boldsymbol{x}_i$ is transformed to a higher dimensional or even infinite dimensional feature vector $\phi(\boldsymbol{x}_i)$, where ϕ is a nonlinear function accomplished $k(\boldsymbol{x}_i,\boldsymbol{x}_j)=\langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)\rangle$. Let $\boldsymbol{\Phi}=[\phi(\boldsymbol{x}_1),\phi(\boldsymbol{x}_2),\ldots,\phi(\boldsymbol{x}_n)]^\top$ and $\boldsymbol{\Phi}_j=[\phi(\boldsymbol{x}_{j1}),\phi(\boldsymbol{x}_{j2}),\ldots,\phi(\boldsymbol{x}_{jn_j})]^\top$ record the transformed features of all instances and the instances in class j. By the representer theorem [42], $\boldsymbol{W}$ can be expressed as a linear combination of the transformed instances

$$\boldsymbol{W}=\boldsymbol{\Phi}^\top\boldsymbol{A},\tag{20}$$

where $\boldsymbol{A}=[\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_c]^\top\in\Re^{n\times c}$ is a linear combination coefficient matrix. Inserting (20) into the model (8), kernel RLSSVC can be formulated as

$$\min_{\boldsymbol{A},\tilde{\boldsymbol{e}}_j}\frac{1}{2n}\sum_{j=1}^c\|\boldsymbol{Y}_j-\boldsymbol{K}_j\boldsymbol{A}-\boldsymbol{1}_{n_j}\tilde{\boldsymbol{e}}_j^\top\|_F^2+\frac{\lambda_1}{2}\mathrm{tr}(\boldsymbol{A}^\top\boldsymbol{K}\boldsymbol{A})+\frac{\lambda_2}{2}\sum_{j=1}^c\frac{n_j}{n}\|\tilde{\boldsymbol{e}}_j\|_2^2\tag{21}$$

where $\boldsymbol{K}=\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\in\Re^{n\times n}$ is the kernel matrix, and $\boldsymbol{K}_j=\boldsymbol{\Phi}_j\boldsymbol{\Phi}^\top\in\Re^{n_j\times n}$.

Solving (21) for the large datasets is challenging. To scale up (21) on limited resources, one common approach is to approximate the kernel learning problem with a linear learning problem. The popular methods including Nyström [43,44], random features [45], and their numerous extensions. Here, the Nyström method is introduced to approximate the kernel matrix $\boldsymbol{K}$, that is

$$\boldsymbol{K}\approx\boldsymbol{K}_{\mathbb{M}\mathbb{B}}\boldsymbol{K}_{\mathbb{B}\mathbb{B}}^{-1}\boldsymbol{K}_{\mathbb{M}\mathbb{B}}^\top,\tag{22}$$

where $\mathbb{M}=\{1,2,\ldots,n\}$ is the index set of the input instances, $\mathbb{B}\subset\mathbb{M}$ is a landmark set of $r(=|\mathbb{B}|)$ instances, $\boldsymbol{K}_{\mathbb{M}\mathbb{B}}\in\Re^{n\times r}$ is the submatrix of $\boldsymbol{K}$, whose elements are $k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ for $i\in\mathbb{M}$ and $j\in\mathbb{B}$, and $\boldsymbol{K}_{\mathbb{B}\mathbb{B}}\in\Re^{r\times r}$ has the similar meanings. Actually, Nyström method provides a natural approximation to the optimal rank-r kernel map $\tilde{\phi}$ accomplished $\tilde{\phi}(\boldsymbol{x})=\boldsymbol{K}_{\boldsymbol{x}\mathbb{B}}\boldsymbol{U}_r\boldsymbol{\Sigma}_r^{-1/2}$. Here $\boldsymbol{K}_{\boldsymbol{x}\mathbb{B}}=[k(\boldsymbol{x},\boldsymbol{x}_1),\ldots,k(\boldsymbol{x},\boldsymbol{x}_i),\ldots,k(\boldsymbol{x},\boldsymbol{x}_{|\mathbb{B}|})]^\top$ represents the approximate kernel map of $\boldsymbol{x}$ based on the selected landmark set $\mathbb{B}$, $\boldsymbol{\Sigma}_r$ is a diagonal matrix where the diagonal entries are the eigenvalues of $\boldsymbol{K}_{\mathbb{B}\mathbb{B}}$, and $\boldsymbol{U}_r$ are the corresponding eigenvectors. Then, kernel RLSSVC is reduced as

$$\min_{\boldsymbol{W},\tilde{\boldsymbol{e}}_j}\frac{1}{2n}\sum_{j=1}^c\|\boldsymbol{Y}_j-\tilde{\boldsymbol{\Phi}}_j\boldsymbol{W}-\boldsymbol{1}_{n_j}\tilde{\boldsymbol{e}}_j^\top\|_F^2+\frac{\lambda_1}{2}\|\boldsymbol{W}\|_F^2+\frac{\lambda_2}{2}\sum_{j=1}^c\frac{n_j}{n}\|\tilde{\boldsymbol{e}}_j\|_2^2,\tag{23}$$

where $\tilde{\boldsymbol{\Phi}}_j=[\tilde{\phi}(\boldsymbol{x}_{j1}),\tilde{\phi}(\boldsymbol{x}_{j2}),\ldots,\tilde{\phi}(\boldsymbol{x}_{jn_j})]^\top\in\Re^{n_j\times r}$ is the kernel approximation of the instances in class j. By the optimality conditions, the following equation is obtained.

$$\begin{cases}\boldsymbol{W}=\left(n\lambda_1\boldsymbol{I}_d+\sum_{j=1}^c\tilde{\boldsymbol{\Phi}}_j^\top\boldsymbol{L}_j\tilde{\boldsymbol{\Phi}}_j\right)^{-1}\sum_{j=1}^c\tilde{\boldsymbol{\Phi}}_j^\top\boldsymbol{L}_j\boldsymbol{Y}_j,\\ \tilde{\boldsymbol{e}}_j^\top=\dfrac{1}{n_j(1+\lambda_2)}\boldsymbol{1}_{n_j}^\top(\boldsymbol{Y}_j-\tilde{\boldsymbol{\Phi}}_j\boldsymbol{W}),\quad j\in[c].\end{cases}\tag{24}$$

For a new test instance $\boldsymbol{x} \in \Re^d$, it is assigned to class k, depending on

$$\underset{k\in[c]}{\operatorname{argmin}}(\boldsymbol{y}^k - \boldsymbol{W}^\top\tilde{\phi}(\boldsymbol{x}) - \tilde{\boldsymbol{e}}_k)^\top(\boldsymbol{y}^k - \boldsymbol{W}^\top\tilde{\phi}(\boldsymbol{x}) - \tilde{\boldsymbol{e}}_k), \tag{25}$$

where $\boldsymbol{y}^k$ denotes a column vector whose kth element is +1 and the rest of elements is -1.

6. Experiments

In this section, we investigate the performance of our proposed RLSSVC using the artificial and benchmark datasets. The experiment results in [19,20] show that the classification performance of MSDLSR and ReLSR is better than that of DLSR [2], L_1-SVM, L_2-SVM [15] and logistic regression [46] on the most datasets. The classification experiments in [35] show that RLSSVM is better than W-LSSVM [28] in terms of the classification accuracy and training time. Therefore, we compare our RLSSVC with several state-of-the-art multiclass learning methods:

- MSVM [15,16]: Generalizes the notions of margin and large margin loss to the multiclass problems, and casts the learning problem into a single quadratic program. Codes are available in https://www.cs.cornell.edu/people/tj/svm_light/svm_multiclass.html.
- RMSVM [37]: A robust angle-based multiclass SVM (RMSVM) model using truncated hinge loss with L_2 regularization. As what has been done in [37], we also set the truncated parameter s in RMSVM to $-1/(c-1)$, where c is the number of classes. As recommended in [37], we employ the DC algorithm to solve the nonconvex problem via a sequence of convex subproblems, and apply the coordinate descent method [47] to solve the sub-problems. In our experiment, the code for RMSVM is written according to the pseudo-code in [37,47].
- RLSSVM [35]: Uses a truncated least squares loss to deal with the outliers. The truncated parameter τ in RLSSVM is selected from the interval of [0.2:0.3:3]. Our code for RLSSVM is written according to the RLSSVM algorithm in [35].
- MLG-LSC [21]: Based on the geometric mean metric learning, a metric matrix for the errors of LSR is learned, which yields small distances for the same class, while large ones for the different classes. The weight parameters α in metric learning is selected from the set $\{10^{-6}, 10^{-5}, 10^{-4}, \ldots, 10^{-1}\}$. The codes for MLG-LSC are available in https://github.com/ChuanxingGeng/MLG-LSC.
- MSDLSR [20]: Imposes an regularization with respect to the margin on the DLSR to control the number of support vectors. The margin scalable parameter β in MSDLSR is selected from the interval of [0:0.005:0.05]. The code for MSDLSR is written according to the pseudo-code in [20].
- ReLSR [19]: This method directly learns the regression target and projection matrix from the training data, which guarantees a large margin constraint for the requirement of the correct classification for each instance. The code for ReLSR is written according to the pseudo-code in [19].

The regularization parameters λ_1, λ_2 in RLSSVC, and λ in MSVM, RMSVM, MLG-LSC, MSDLSR, and ReLSR are selected from the set $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$. All the parameters in these methods are selected from the corresponding candidate set by the five-fold cross-validation technique on the corresponding training datasets. All the experiments are implemented in Matlab R2017a environment on a PC with an Intel core i7-4790 processor (3.60 GHz) and 8 GB RAM.

Table 2
Brief description of the twelve benchmark datasets.

Data set	Classes	Features	Total num.	Train num.
Cora_OS	4	6737	1246	997
Coil20	20	256	1440	1152
DNA	3	180	2586	1400
Satimage	6	36	6435	4435
Usps	10	256	9298	7291
Letter	26	16	15 500	10 500
Shuttle	7	9	58 000	43 500
Sensorless	11	48	58 509	48 509
Connect-4	3	126	67 557	54 046
Hand-poses	5	37	78 096	62 477
Acoustic	3	50	98 528	78 823
Covtype	7	54	580 382	464 180

6.1. Experimental results with artificial data

To demonstrate the numerical performance of our new classifier, we compare the proposed RLSSVC with LSR, MLG-LSC, MSVM, RLSSVM and RMSVM on a three classification dataset that includes 120 training instances and 180 test instances. The dataset obeys the Gaussian distribution. Specifically, the instances of class j satisfy $\boldsymbol{X}_j \sim \mathcal{N}(\mu_j, \sigma)$, where $\mu_1 = (0, 1)$, $\mu_2 = (-\sqrt{3}/2, -1/2)$, $\mu_3 = (\sqrt{3}/2, -1/2)$ and $\sigma = [0.15 \;\; 0; 0 \;\; 0.15]$. In particular, we contaminate the dataset with outliers as following: 1) selecting six instances from class 2; 2) relabeling the three instances as class 1, and the remaining as class 3. The Bayes optimal classifier [3] is also chosen as the baseline method. Fig. 4 shows the experimental results.

From the comparisons of Fig. 4(a)–(f), it can be seen that for the dataset without outliers, the classification boundaries of MLG-LSC, MSVM, RLSSVM, RMSVM and RLSSVC are closer to that of the Bayes optimal classifier, and the classification accuracies are higher than that of LSR. Further, Fig. 4 shows that the classification boundaries of LSR, MLG-LSC and MSVM are more shifted toward the outliers than RLSSVM, RMSVM and RLSSVC. For example, in Fig. 4(a)–(c), the classification boundaries between the class 1 and class 2, the class 2 and class 3 slopes very sharply toward the six outliers (the three instances marked as '⊡' are located in the area of the '+' class but are labeled as the '○' class, and the three instances marked as '△' are located in the area of the '+' class but are labeled as the '△' class), while the classification boundaries of RLSSVM, RMSVM and RLSSVC are almost unchanged, as shown in Fig. 4(d)–(f). It can be concluded that our RLSSVC is insensitive to outliers.

6.2. Experimental results with benchmark datasets

In this subsection, the proposed RLSSVC is then tested on several benchmark datasets. The information of the datasets are listed in Table 2, and the details about the datasets are described as follows.

(1) Cora_OS is a subset containing the research papers about operating system [48].
(2) The dataset of Coil20 includes 20 objects [49],[1] each of which has 72 gray images, which are taken from the different view directions. Each image is down-sampled to have 16×16 pixels. Thus, the dimensionality is 256.
(3) Hand-poses dataset consists of 5 static gestures (hand poses) captured for 12 users, which has 62477 training instances and 15619 test instances. Each instance has 37

[1] www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

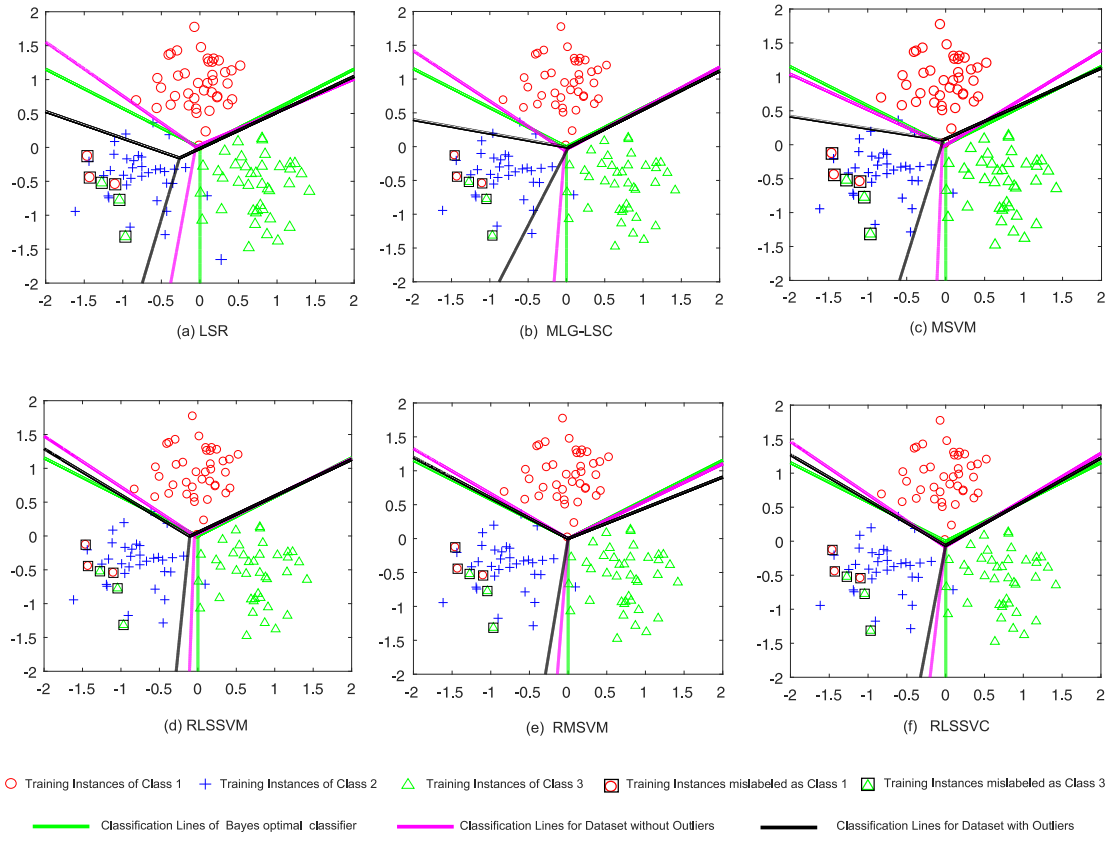

Fig. 4. Plots for comparing the classification boundaries of LSR, MLG-LSC, MSVM, RLSSVM, RMSVM and RLSSVC on the artificial dataset without and with outliers. For the dataset without outliers, the test accuracies of these six algorithms are 95.56%, 96.11%, 96.11%, 96.11%, 96.11% and 96.11%, respectively. For the dataset with outliers, the test accuracies of LSR, MLG-LSC, MSVM, RLSSVM, RMSVM and RLSSVC are 91.67%, 93.89%, 93.96%, %96.11%, 96.11% and 96.11%, respectively.

features. Which can be taken from UCI machine learning data repository.[2]

(4) The rest of the datasets (DNA, Satimage, Usps, Letter, Shuttle, Sensorless, Connect-4, Acoustic, Covtype) is taken from the LIBSVM machine learning data repository.[3]

6.2.1. Comparison of robustness

In order to evaluate the robustness of the proposed method to varying degrees of outliers, we demonstrate the performances of all the methods with respect to the different degrees of outliers on the benchmark datasets, as shown in Fig. 5. Specifically, we inject the label noises into the training datasets as following: (1) randomly selecting {0%, 5%, 10%, 15%, 20%} of the training instances from each class; (2) randomly relabeling them into the other classes. All the plots were averaged over 10 random trials. The only exception is to repeat the experiments 3 times for the RMSVM because it is computationally very expensive. In addition, the experimental result of the RMSVM on the Covtype data is missing because the training time is too long.

As shown in Fig. 5, it is obvious that the average test accuracy of every method for all the datasets descends with the increase of the outlier ratio. However, the average test accuracies of RLSSVC, RMSVM and RLSSVM on all the datasets are much more stable than the remaining methods. Given any degree of outlier, the average test accuracy of RLSSVC is higher than that of the other methods on the most datasets. For the Shuttle, Connect-4 and Acoustic, although the RLSSVC performs inferior to the other methods under the low outlier ratio, RLSSVC still surpass the other methods based on LSR under the higher error ratio. For the DNA, Shuttle and Acoustic, the classification accuracy of RMSVM is slightly higher than that of RLSSVC, but RMSVM is much slower than RLSSVC. Especially for the Shuttle and Acoustic, the training time of RMSVM is more than 10,000 times slower than that of RLSSVC. It is noteworthy that our method is rather stable even for the training data with the serious outliers, which empirically validates the robustness of our method for outliers.

In order to compare the performance of all the methods objectively and fairly, Macro averaged F_1 scores (F_1^{Macro}) [50] and Matthews correlation coefficient (MCC) [51] are also adopted as the evaluation criteria. For each dataset, RMSVM was run 3 times, and the remaining methods were run 10 times. The average and standard deviation of the accuracy, F_1^{Macro} and MCC for these methods on the benchmark datasets with outliers (20%) are reported in Table 3. From the results, we can see that our proposed method with minimizing the mean and variance of the modeling errors for each class is very competitive in terms of both accuracy, F_1^{Macro} and MCC, especially when the dataset is contaminated by the outliers. This is because minimizing the mean and variance of the modeling errors for each class can reduce the error oscillations caused by the outliers.

6.2.2. Comparison of training time

Next, we mainly compare the computational performance among our RLSSVC, MSVM, RMSVM, RLSSVM, MLG-LSC, MSDLSR and ReLSR. Table 4 reports the training time of the seven methods on the benchmark datasets. It can be seen that RLSSVC is the fastest algorithm among the compared approaches. Our RLSSVC is almost 100 times faster than the ReLSR on some large datasets, such as Letter, Shuttle and Covtype. Compared

[2] https://archive.ics.uci.edu/ml/index.php
[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

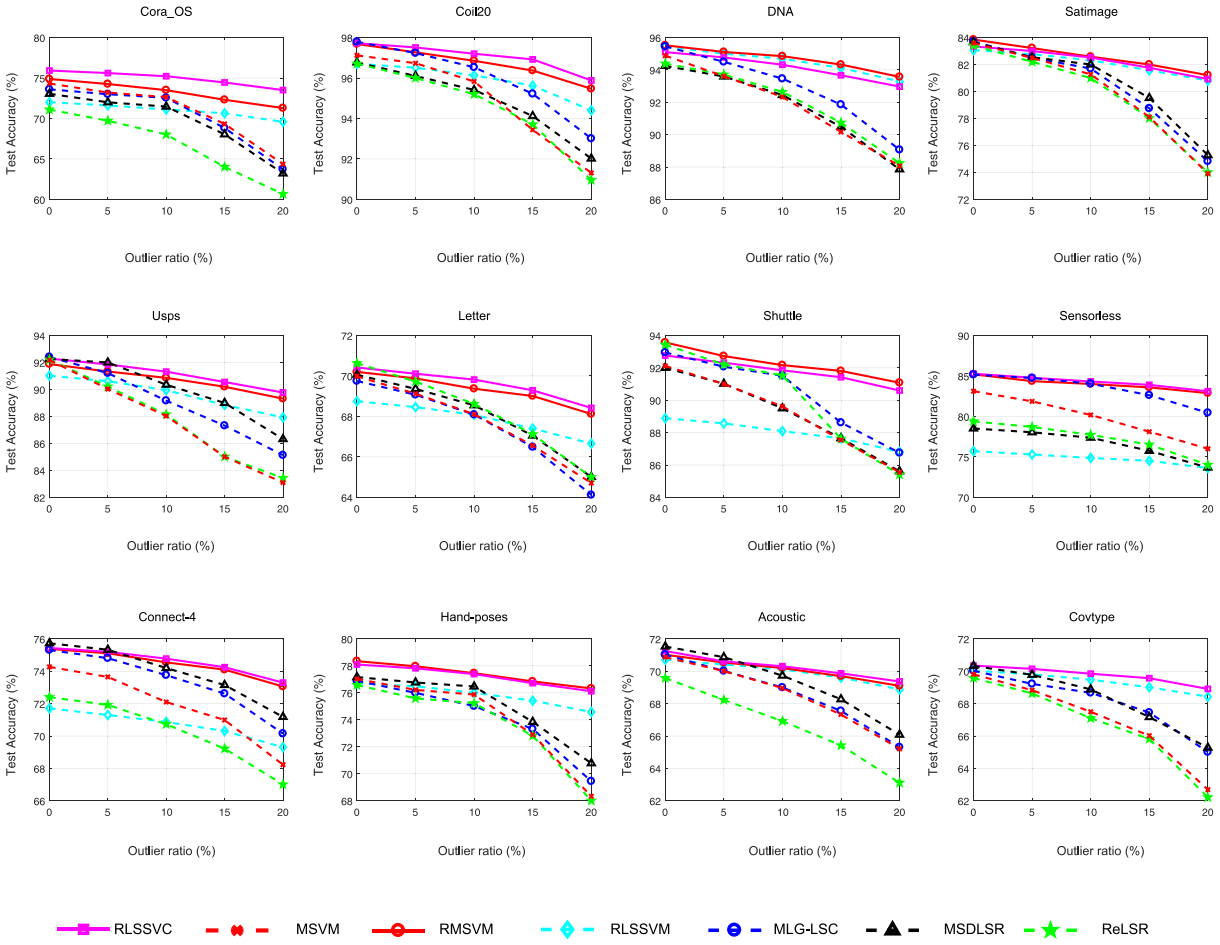

Fig. 5. Robustness experiments with different degrees of outliers. From a comparison between the plots of related algorithms, it is clear that the RLSSVC is more stable under any outlier ratio on the most datasets, while those methods based on the least square loss become worse with increase of outliers. The experimental result of the RMSVM on the Covtype data is missing because the training time is too long.

with MSVM, RLSSVM, MLG-LSC and MSDLSR, RLSSVC is also very competitive in terms of the training time. This is because the objective of RLSSVC is convex, and the closed-form solutions can be obtained only through the KKT condition, resulting in the high computational performance. In particular, with the help of SMW identity [41], the training time of RLSSVC on the Cora_OS dataset is significantly reduced compared with the other methods. This scales our approach up to the high-dimensional data effectively. In contrast, RMSVM has to deal with a non-differentiable non-convex optimization problem with nc constraints. Even if the difference convex algorithm and coordinate descent method are employed to solve RMSVM, it is still computationally very expensive. Even for the smaller datasets, the training time of RMSVM is almost 1000 times slower than our RLSSVC. For the Covtype data, the training procedure of RMSVM already takes more than two days.

6.2.3. Statistical comparisons by friedman test

In order to compare the multiple methods systematically, the Friedman test [52] is employed to compare the test accuracies of the seven methods over the first 11 (the experimental result of the RMSVM on the Covtype data is missing because the training time is too long) benchmark datasets with the different degrees of outliers. For the different degrees of outliers, Friedman test at significance level $\alpha = 0.05$ rejects the null hypothesis of equal performance, which leads to the use of post-hoc tests to find out which algorithms are actually different. Specifically, Nemenyi test is used where the performance of two algorithms is significantly different if their average ranks over all datasets differ by at least one critical difference (CD$= q_\alpha\sqrt{\frac{K(K+1)}{6N}}$), where critical values q_α are based on the studentized range statistic divided by $\sqrt{2}$, K is the number of comparison algorithms, and N is the number of datasets. For clarification, Fig. 6 illustrates the CD diagrams [52] for the seven comparison methods on the eleven benchmark datasets with the different degrees of outliers, where the average rank of each comparing method is marked along the axis. The axis is turned so that the lowest (best) ranks are to the right. Groups of algorithms that are not significantly different according to Nemenyi test are connected with a red line. The critical difference (CD $= 2.7155$ at 0.05 significance level) is also shown above the axis in each subfigure.

To sum up, out of all the 30 comparisons (6 algorithms to compare $\times 5$ degrees of outliers), RLSSVC achieves the statistically comparable performance in only 36.67% cases, i.e. the 11 comparisons against the RMSVM on the datasets with outliers, against the MLG-LSC on the datasets including 0% and 5% outliers, against the MSDLSR on the datasets including 0% outliers, against the MSVM on the datasets including 0% outliers, and against the RLSSVM on the datasets including 10% and 15% outliers. Rather impressively, RLSSVC achieves the statistically superior performance in all the other 63.33% cases and no algorithms have once outperformed RLSSVC.

Based on the above analyses, we conclude that our proposed method (RLSSVC) is more suitable for the classification problems

Table 3
Experimental results of each comparing method (mean ± standard deviation) on the twelve benchmark datasets with outliers (20%). The best values are highlighted in bold. The '-'means the experimental results are missing, because the training time is too long.

Data	Evaluation criterion	MSVM	RMSVM	RLSSVM	MLG-LSC	MSDLSR	ReLSR	RLSSVC
Cora_OS	Accuracy (%)	64.36 ± 0.62	71.29 ± 0.74	69.59 ± 0.88	63.72 ± 0.67	63.21 ± 0.72	60.67 ± 0.56	**73.51 ± 0.51**
	Macro-F_1(%)	56.16 ± 0.68	62.97 ± 0.72	62.21 ± 0.83	55.86 ± 0.68	55.37 ± 0.63	52.03 ± 0.61	**65.16 ± 0.58**
	MCC(%)	51.88 ± 0.81	59.33 ± 0.71	55.04 ± 0.85	50.79 ± 0.81	50.11 ± 0.77	47.67 ± 0.65	**61.38 ± 0.61**
Coil20	Accuracy (%)	91.31 ± 0.55	94.47 ± 0.66	94.40 ± 0.91	93.01 ± 0.66	92.02 ± 0.63	90.96 ± 0.96	**95.88 ± 0.50**
	Macro-F_1(%)	90.21 ± 0.78	94.13 ± 0.72	94.04 ± 0.87	91.00 ± 0.64	89.50 ± 0.60	88.01 ± 1.04	**94.50 ± 0.53**
	MCC(%)	90.16 ± 0.92	94.27 ± 0.88	94.18 ± 0.96	91.15 ± 1.26	89.63 ± 1.06	88.17 ± 1.02	**94.60 ± 0.55**
DNA	Accuracy (%)	88.07 ± 0.71	**93.57 ± 0.62**	92.29 ± 0.88	89.07 ± 1.07	87.86 ± 0.95	88.24 ± 1.72	92.97 ± 0.52
	Macro-F_1(%)	85.42 ± 0.67	**93.13 ± 0.68**	91.87 ± 0.61	86.91 ± 0.73	85.47 ± 0.84	85.67 ± 1.43	92.56 ± 0.53
	MCC(%)	82.86 ± 0.91	**90.77 ± 0.63**	89.73 ± 1.41	84.78 ± 1.52	82.51 ± 1.66	83.09 ± 1.71	90.04 ± 0.67
Satimage	Accuracy (%)	73.92 ± 0.91	**81.21 ± 0.75**	80.02 ± 1.26	74.83 ± 1.15	75.28 ± 1.06	74.01 ± 1.31	80.81 ± 0.70
	Macro-F_1(%)	64.42 ± 0.83	**72.85 ± 0.79**	70.43 ± 1.06	66.50 ± 0.96	67.00 ± 0.93	64.84 ± 0.99	71.54 ± 0.88
	MCC(%)	65.86 ± 1.01	**77.38 ± 0.85**	74.55 ± 1.17	70.91 ± 0.92	72.01 ± 1.03	66.12 ± 1.12	76.03 ± 1.03
Usps	Accuracy (%)	83.10 ± 0.87	88.32 ± 0.62	87.93 ± 0.57	85.13 ± 0.68	86.32 ± 0.51	83.43 ± 1.22	**89.77 ± 0.69**
	Macro-F_1(%)	82.17 ± 0.83	87.33 ± 0.61	85.54 ± 0.62	84.03 ± 0.73	85.17 ± 0.58	82.61 ± 1.15	**88.90 ± 0.71**
	MCC(%)	81.03 ± 1.04	87.17 ± 0.91	84.95 ± 1.23	83.93 ± 1.42	84.84 ± 1.00	81.22 ± 1.62	**88.86 ± 0.89**
Letter	Accuracy (%)	64.71 ± 0.91	67.86 ± 1.12	66.66 ± 1.25	64.12 ± 1.06	65.01 ± 1.26	65.00 ± 1.79	**68.42 ± 0.70**
	Macro-F_1(%)	57.62 ± 1.07	65.73 ± 1.21	64.78 ± 1.11	56.34 ± 1.27	60.74 ± 1.32	60.66 ± 1.56	**66.63 ± 0.77**
	MCC(%)	56.21 ± 1.12	64.68 ± 1.37	63.69 ± 1.57	55.34 ± 1.34	60.66 ± 1.43	60.33 ± 2.01	**65.00 ± 0.81**
Shuttle	Accuracy (%)	85.55 ± 0.93	**91.08 ± 0.78**	86.81 ± 0.77	86.75 ± 0.65	85.61 ± 0.93	85.41 ± 1.13	90.60 ± 0.23
	Macro-F_1(%)	70.62 ± 1.02	**77.89 ± 0.77**	73.58 ± 0.72	73.01 ± 0.60	71.33 ± 0.82	69.98 ± 0.95	76.56 ± 0.24
	MCC(%)	71.21 ± 1.10	**78.38 ± 0.81**	74.21 ± 1.48	73.56 ± 1.23	71.78 ± 1.73	70.21 ± 1.21	77.33 ± 0.44
Sensorless	Accuracy (%)	76.01 ± 1.14	81.88 ± 0.92	73.62 ± 1.12	80.45 ± 1.25	73.68 ± 1.05	74.01 ± 1.32	**83.11 ± 0.60**
	Macro-F_1(%)	74.82 ± 1.10	80.57 ± 1.03	71.87 ± 1.10	78.57 ± 1.10	72.03 ± 1.00	73.21 ± 1.24	**82.29 ± 0.59**
	MCC(%)	72.21 ± 1.33	77.95 ± 1.78	70.13 ± 2.27	75.68 ± 2.26	71.37 ± 2.15	71.87 ± 2.52	**80.05 ± 1.20**
Connect-4	Accuracy (%)	68.23 ± 0.96	72.36 ± 1.31	69.31 ± 0.91	70.15 ± 0.53	71.18 ± 0.69	67.01 ± 1.03	**73.30 ± 0.47**
	Macro-F_1(%)	39.64 ± 1.16	45.59 ± 0.88	40.51 ± 0.91	41.27 ± 0.66	42.07 ± 0.71	35.63 ± 0.96	**47.55 ± 0.56**
	MCC(%)	34.65 ± 1.27	40.93 ± 1.67	35.23 ± 1.62	36.78 ± 1.21	37.77 ± 1.28	32.11 ± 1.37	**42.33 ± 0.94**
Hand-poses	Accuracy (%)	68.33 ± 0.85	**76.83 ± 0.97**	74.58 ± 1.91	69.85 ± 1.08	70.78 ± 1.22	68.01 ± 2.00	76.12 ± 0.67
	Macro-F_1(%)	67.64 ± 1.32	**74.16 ± 1.79**	71.78 ± 1.78	69.57 ± 1.24	70.34 ± 1.85	66.66 ± 1.30	73.28 ± 1.11
	MCC(%)	62.65 ± 1.28	**68.38 ± 1.66**	66.71 ± 1.82	63.92 ± 1.39	65.18 ± 1.96	61.85 ± 1.58	67.26 ± 1.25
Acoustic	Accuracy (%)	65.21 ± 0.73	69.10 ± 0.57	68.88 ± 0.78	65.31 ± 0.88	66.08 ± 0.76	63.11 ± 1.32	**69.37 ± 0.51**
	Macro-F_1(%)	59.04 ± 1.01	61.86 ± 0.58	61.53 ± 0.83	59.16 ± 0.95	60.33 ± 0.79	58.66 ± 1.30	**62.55 ± 0.52**
	MCC(%)	45.68 ± 1.03	48.38 ± 1.27	48.11 ± 1.12	45.87 ± 1.25	46.56 ± 1.07	41.37 ± 1.65	**49.31 ± 1.01**
Covtype	Accuracy (%)	62.69 ± 1.54	–	68.43 ± 0.91	65.01 ± 1.15	65.28 ± 1.05	62.71 ± 1.37	**68.91 ± 0.75**
	Macro-F_1(%)	26.54 ± 1.33	–	32.01 ± 1.21	27.68 ± 1.22	30.31 ± 1.07	26.88 ± 1.56	**32.15 ± 1.02**
	MCC(%)	45.07 ± 1.55	–	48.80 ± 1.56	46.36 ± 1.38	47.33 ± 1.32	45.13 ± 1.61	**49.11 ± 1.12**

Table 4
Training time (seconds) of each comparing algorithm on the benchmark datasets. The best values are highlighted in bold. The '-'means the experimental results are missing, because the training time is too long.

Data	MSVM	RMSVM	RLSSVM	MLG-LSC	MSDLSR	ReLSR	RLSSVC
Cora OS	4.14 ± 0.66	462.83 ± 2.31	4.85 ± 0.21	1.79 ± 0.10	41.43 ± 0.70	53.97 ± 0.81	**0.46 ± 0.01**
Coil20	0.16 ± 0.02	751.7 ± 2.63	0.03 ± 0.00	0.02 ± 0.00	0.26 ± 0.01	0.87 ± 0.02	**0.01 ± 0.00**
DNA	0.09 ± 0.01	39.92 ± 0.78	0.04 ± 0.00	0.02 ± 0.00	0.39 ± 0.01	0.88 ± 0.03	**0.01 ± 0.00**
Satimage	0.12 ± 0.01	100.1 ± 1.77	0.02 ± 0.00	0.02 ± 0.01	0.18 ± 0.01	1.37 ± 0.03	**0.01 ± 0.00**
Usps	56.35 ± 0.31	4524 ± 1.43	0.33 ± 0.01	0.24 ± 0.01	7.84 ± 0.19	10.68 ± 0.22	**0.23 ± 0.01**
Letter	9.52 ± 0.11	10436 ± 2.78	0.20 ± 0.01	0.05 ± 0.00	0.21 ± 0.02	12.43 ± 0.36	**0.03 ± 0.00**
Shuttle	0.77 ± 0.54	14957 ± 3.61	0.04 ± 0.00	0.03 ± 0.01	0.37 ± 0.02	17.65 ± 0.61	**0.02 ± 0.00**
Sensorless	25.81 ± 0.64	94851 ± 3.56	0.22 ± 0.02	0.14 ± 0.01	3.90 ± 0.12	29.99 ± 0.75	**0.13 ± 0.01**
Connect-4	16.81 ± 0.34	14316 ± 2.67	0.26 ± 0.02	0.21 ± 0.02	4.19 ± 0.22	17.22 ± 0.58	**0.19 ± 0.01**
Hand-poses	8.81 ± 0.24	15939 ± 2.31	0.05 ± 0.00	0.03 ± 0.00	0.49 ± 0.01	10.13 ± 0.27	**0.02 ± 0.00**
Acoustic	15.02 ± 0.24	14195 ± 2.17	0.38 ± 0.02	0.22 ± 0.01	6.82 ± 0.15	21.16 ± 0.58	**0.19 ± 0.01**
Covtype	47.35 ± 1.12	–	2.98 ± 0.12	1.26 ± 0.04	20.76 ± 0.67	192 ± 11.20	**1.23 ± 0.03**

with outliers, as it has higher test accuracy, requires less training time than the other methods.

6.3. Improvement of RLSSVC on the hard benchmark datasets

In this subsection, several experiments were conducted on the last four hard benchmark datasets to verify the effectiveness of the improvement of RLSSVC with metric learning and kernel trick.

6.3.1. RLSSVC with metric learning

MLG-LSC has adopted the metric learning in the second stage, so MSVM, RMSVM, RLSSVM, MSDLSR, ReLSR and RLSSVC are improved by introducing the metric learning. Specifically, for MSVM, RMSVM, RLSSVM, MSDLSR and ReLSR, the metric matrix $\boldsymbol{M}$ can be learned through (17) (where $\tilde{\boldsymbol{E}} = \boldsymbol{0}$), and each test instance is classified by (10), after obtaining the optimal projection matrix $\boldsymbol{W}$. For clarity, the improvements of these comparison methods are referred to ML-MSVM, ML-RMSVM, ML-RLSSVM, ML-MSDLSR, ML-ReLSR and ML-RLSSVC for short. The experimental results with average and standard deviation of these methods on the last four benchmark datasets with outliers(20%) are reported in Table 5.

In Table 5, it can be observed that the results are better for the improvements of the six methods based on metric learning. This fact demonstrates the virtues of the metric learning for solving

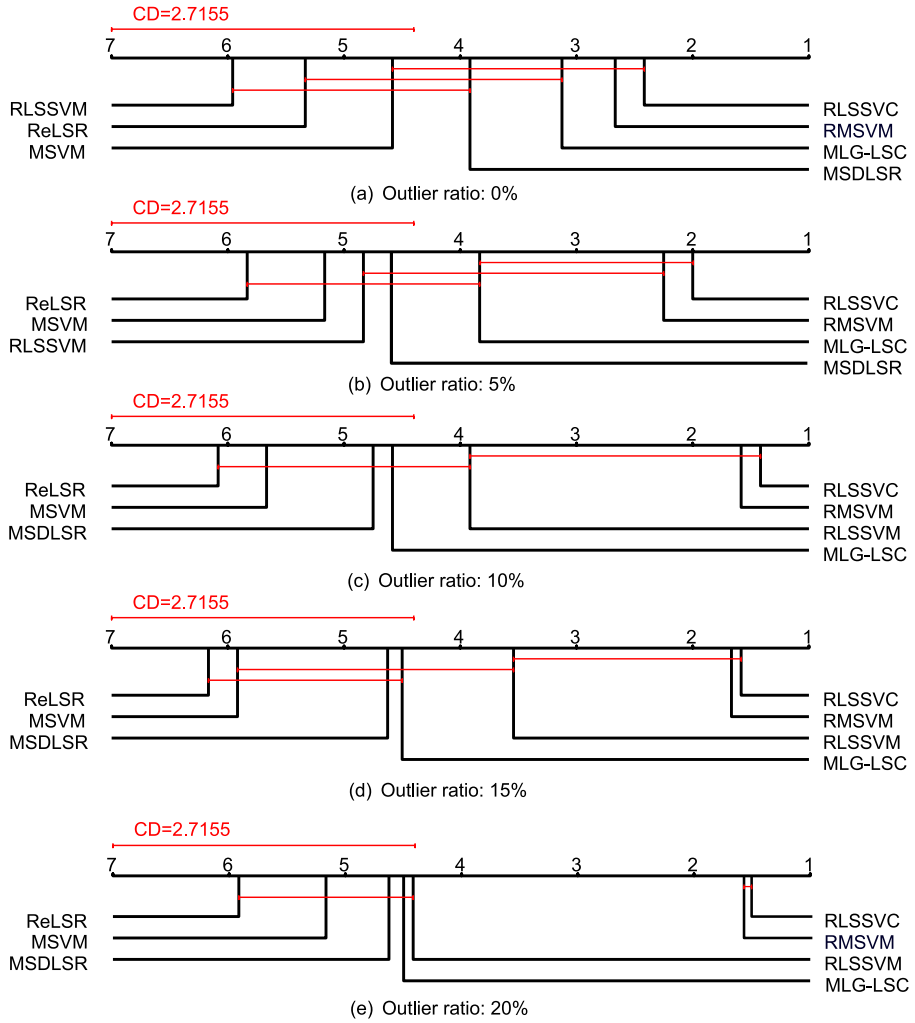

Fig. 6. CD diagrams of the five comparison models on the twelve benchmark datasets with different degrees of outliers. It is clear that RLSSVC achieves the statistically superior performance on the datasets with the different outlier ratios.

Table 5
Experimental results of each comparing algorithm with metric learning (mean ± standard deviation) on the last four hard benchmark datasets with outliers (20%). The best values are highlighted in bold. The '-'means the experimental results are missing, because the training time is too long.

Data	Evaluation criterion	ML-MSVM	ML-RMSVM	ML-RLSSVM	ML-MSDLSR	ML-ReLSR	ML-RLSSVC
Connect-4	Accuracy (%)	71.27 ± 0.73	73.18 ± 0.68	71.14 ± 0.95	72.19 ± 0.67	70.81 ± 1.21	**74.11 ± 0.55**
	Macro-F_1(%)	41.33 ± 0.83	48.27 ± 0.77	41.26 ± 0.99	43.31 ± 0.78	37.55 ± 1.12	**49.35 ± 0.61**
	MCC(%)	37.78 ± 1.12	42.35 ± 1.14	37.33 ± 1.24	40.23 ± 1.17	35.46 ± 1.33	**44.22 ± 0.72**
Hand-poses	Accuracy (%)	74.54 ± 1.27	**83.11 ± 1.32**	81.97 ± 1.22	78.57 ± 1.06	73.14 ± 1.37	82.78 ± 0.95
	Macro-F_1(%)	72.91 ± 1.15	**80.17 ± 1.45**	77.78 ± 1.17	75.86 ± 1.23	70.29 ± 1.54	79.01 ± 1.02
	MCC(%)	65.49 ± 1.36	**74.55 ± 1.57**	71.24 ± 1.23	69.45 ± 1.35	64.82 ± 1.51	73.92 ± 1.08
Acoustic	Accuracy (%)	67.54 ± 1.27	71.78 ± 0.68	70.16 ± 0.62	67.91 ± 0.65	64.18 ± 1.01	**72.16 ± 0.55**
	Macro-F_1(%)	62.26 ± 0.92	65.17 ± 0.87	64.01 ± 0.66	62.32 ± 0.68	60.44 ± 1.03	**66.85 ± 0.57**
	MCC(%)	46.93 ± 1.18	50.55 ± 1.24	49.66 ± 0.71	47.79 ± 0.87	42.25 ± 1.22	**51.58 ± 0.62**
Covtype	Accuracy (%)	64.44 ± 1.43	–	69.78 ± 1.33	66.55 ± 1.21	64.62 ± 1.39	**70.32 ± 1.12**
	Macro-F_1(%)	28.31 ± 1.36	–	33.83 ± 1.41	32.24 ± 1.23	27.77 ± 1.51	**34.87 ± 1.16**
	MCC(%)	47.72 ± 1.27	–	51.43 ± 1.52	50.13 ± 1.30	47.98 ± 1.58	**52.31 ± 1.22**

the multiclass classification problems. The results in Table 5 show that ML-RLSSVC outperforms the other methods on the most datasets. Therefore, our proposed RLSSVC has good expansibility.

6.3.2. Scalable kernel RLSSVC

In this subsection, the robustness of the scalable kernel RLSSVC is verified in dealing with hard datasets with outliers. In all the experiments, the original instance $\boldsymbol{x}$ is first transformed to $\tilde{\phi}(\boldsymbol{x})$ using the Nyström method, and then used as the input to the seven compared models discussed in Section 6.2. For clarity, we refer to the scalable kernel versions of the models discussed in Section 6.2 as NMSVM, NRMSVM, NRLSSVM, NMLG-LSC, NMSDLSR, NReLSR and NRLSSVC, respectively. The Gauss kernel $\boldsymbol{k}(\boldsymbol{x},\boldsymbol{z}) = \exp(-,\|\boldsymbol{x}-\boldsymbol{z}\|^2)$ is used as the kernel function. The kernel spread parameter γ was chosen roughly by the five-fold cross-validation with $\gamma \in \{2^{-9},\ldots,2^3\}$. The number r of landmark points is set to 700.

Table 6
Experimental results of each scalable kernel version of comparing algorithm (mean ± standard deviation) on the last four hard benchmark datasets with outliers (20%). The best values are highlighted in bold. The '–' means the experimental results are missing, because the training time is too long.

Data	Evaluation criterion	NMSVM	NRMSVM	NRLSSVM	NMLG-LSC	NMSDLSR	NReLSR	NRLSSVC
Connect-4	Accuracy (%)	75.49 ± 0.74	76.66 ± 0.87	74.14 ± 0.58	75.08 ± 0.65	76.19 ± 0.75	73.81 ± 0.71	**77.51 ± 0.45**
	Macro-F_1(%)	49.38 ± 0.91	50.24 ± 0.83	46.67 ± 0.72	48.04 ± 0.62	49.81 ± 0.68	42.30 ± 0.77	**51.16 ± 0.61**
	MCC(%)	45.65 ± 0.88	46.33 ± 0.79	42.86 ± 0.71	44.12 ± 0.70	45.55 ± 0.73	40.88 ± 0.81	**47.62 ± 0.68**
Hand-poses	Accuracy (%)	84.79 ± 1.21	**91.76 ± 1.04**	88.97 ± 1.31	84.12 ± 0.97	86.57 ± 1.17	83.14 ± 1.76	90.53 ± 0.88
	Macro-F_1(%)	81.41 ± 1.36	**88.71 ± 1.47**	86.78 ± 1.52	82.49 ± 1.15	85.86 ± 1.35	80.29 ± 1.65	87.01 ± 0.94
	MCC(%)	75.85 ± 1.33	**82.07 ± 1.59**	79.24 ± 1.63	76.38 ± 1.28	77.45 ± 1.42	74.82 ± 1.50	80.92 ± 1.03
Acoustic	Accuracy (%)	73.59 ± 1.19	77.36 ± 0.92	76.16 ± 0.81	73.57 ± 0.91	73.91 ± 0.87	72.18 ± 1.46	**78.16 ± 0.61**
	Macro-F_1(%)	70.76 ± 1.17	72.24 ± 0.91	72.01 ± 0.85	70.35 ± 0.96	71.32 ± 0.93	69.44 ± 1.42	**73.73 ± 0.63**
	MCC(%)	55.67 ± 1.16	58.87 ± 1.37	58.33 ± 1.21	55.11 ± 1.33	56.46 ± 1.55	52.25 ± 1.71	**60.03 ± 1.05**
Covtype	Accuracy (%)	68.59 ± 1.19	–	75.13 ± 1.27	70.25 ± 1.37	71.55 ± 1.20	68.35 ± 1.44	**76.53 ± 1.03**
	Macro-F_1(%)	46.32 ± 1.36	–	50.77 ± 1.55	47.32 ± 1.58	49.24 ± 1.19	45.77 ± 1.67	**51.23 ± 1.46**
	MCC(%)	56.67 ± 1.46	–	61.92 ± 1.63	58.29 ± 1.73	60.88 ± 1.31	55.98 ± 1.79	**62.02 ± 1.57**

The last four datasets, described in Table 2, are employed to evaluate the robustness of the proposed scalable kernel RLSSVC. The accuracy, F_1^{Macro} and MCC on all the four datasets (700 landmark points) with the outliers at 20% are reported in Table 6. Table 6 clearly show that the NRLSSVC with the variance reduction yields better classification results than the other compared models on the most datasets, with respect to both F_1-measure scores and Matthews correlation coefficient. This validate that our NRLSSVC is effective for solving the nonlinear classification problems.

7. Conclusion

In this paper, a novel robust least squares support vector classifier (RLSSVC), minimizing the variance and mean of the modeling errors for each class, is proposed. The theoretical analysis shows that the variance of the modeling errors of RLSSVC is smaller than that of RLSSVR in dealing with the binary classification problems. According to the validity of the RLSSVC for solving the binary classification problems, RLSSVC is then generalized for solving the multiclass classification problems. The robustness analysis provides a theoretical guarantee for the robustness of RLSSVC, which delivers that RLSSVC assigns the smaller weights for the training instances with the larger errors, while the larger weights for the training instances with the smaller errors. Experimental results show that the proposed RLSSVC achieves the better classification effect with the lower computational costs.

CRediT authorship contribution statement

Jiajun Ma: Conceptualization, Methodology, Data curation, Writing - original draft. **Shuisheng Zhou:** Theoretical guidance, Experimental review. **Dong Li:** Language editing, Grammar proofreading.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant 61772020.

References

[1] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300.
[2] Shiming Xiang, Feiping Nie, Gaofeng Meng, Chunhong Pan, Changshui Zhang, Discriminative least squares regression for multiclass classification and feature selection, IEEE Trans. Neural Netw. Learn. Syst. 23 (11) (2012) 1738–1754.
[3] Shai Shalev-Shwartz, Shai Ben-David, UnderstandIng Machine Learning: From Theory To Algorithms, Cambridge University press, 2014.
[4] Aymeric Dieuleveut, Nicolas Flammarion, Francis Bach Harder, Better, faster, stronger convergence rates for least-squares regression, J. Mach. Learn. Res. 18 (101) (2017) 1–51.
[5] J.A.K. Suykens, J. Vandewalle, Multiclass least squares support vector machines, in: International Joint Conference on Neural Networks, 1999, pp. 900–903.
[6] Erin L.Allwein, Robert E.Schapire, Yoram Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2000) 113–141.
[7] Divya Tomar, Sonali Agarwal, A comparison on multi-class classification methods based on least squares twin support vector machine, Knowl. Based Syst. 81 (2015) 131–147.
[8] Shuisheng Zhou, Sparse LSSVM in primal using Cholesky factorization for large-scale problems, IEEE Trans. Neural Netw. Learn. Syst. 27 (4) (2016) 783–795.
[9] Corinna Cortes, V.N. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[10] A. Rocha, S.K. Goldenstein, Multiclass from binary: Expanding one-versus-all, one-versus-one and ECOC-based approaches, IEEE Trans. Neural Netw. Learn. Syst. 25 (2) (2017) 289–302.
[11] Carl Brunner, Andreas Fischer, Klaus Luig, Thorsten Thies, Pairwise support vector machines and their application to large scale problems, J. Mach. Learn. Res. 13 (1) (2012) 2279–2292.
[12] M. Liu, D. Zhang, S. Chen, H. Xue, Joint binary classifier learning for ECOC-based multi-class classification, IEEE Trans. Pattern Anal. Mach. Intell. 38 (11) (2016) 2335–2341.
[13] Takashi Takenouchi, Shin Ishii, Binary classifiers ensemble based on Bregman divergence for multi-class classification, Neurocomputing 273 (2018) 424–434.
[14] Yoonkyung Lee, Yi Lin, Grace Wahbay, Multicategory support vector machines, theory, and application to the classication of microarray data and satellite radiance data, J. Amer. Statist. Assoc. 99 (465) (2004) 67–81.
[15] Koby Crammer, Yoram Singer, On the algorithmic implementation of multiclass kernel-based vector machines, J. Mach. Learn. Res. 2 (2) (2001) 265–292.
[16] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun, Large margin methods for structured and interdependent output variables, J. Mach. Learn. Res. 6 (2) (2006) 1453–1484.
[17] Chih Wei Hsu, Chih Jen Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. Learn. Syst. 13 (2) (2000) 415–425.
[18] Ürün Doğan, Tobias Glasmachers, Christian Igel, A unified view on multi-class support vector classification, J. Mach. Learn. Res. 17 (45) (2016) 1–32.
[19] X. Zhang, L. Wang, S. Xiang, C. Liu, Retargeted least squares regression algorithm, IEEE Trans. Neural Netw. Learn. Syst. 26 (9) (2015) 2206–2213.
[20] L. Wang, X. Zhang, C. Pan, MSDLSR: Margin scalable discriminative least squares regression for multicategory classification, IEEE Trans. Neural Netw. Learn. Syst. 27 (12) (2016) 2711–2717.
[21] Chuanxing Geng, Songcan Chen, Metric learning-guided least squares classifier learning, IEEE Trans. Neural Netw. Learn. Syst. 29 (12) (2018) 6409–6414.
[22] Kilian Q. Weinberger, Lawrence K. Saul, Distance metric learning for large margin nearest neighbor classification, J. Mach. Learn. Res. 10 (1) (2009) 207–244.

[23] Pourya Zadeh, Reshad Hosseini, Suvrit Sra, Geometric mean metric learning, in: International Conference on Machine Learning, pages, 2016, pp. 2464–2471.
[24] Bernhard Schölkopf, Learning with kernels: Support vector machines, regularization, optimization, and beyond, IEEE Trans. Neural Netw. Learn. Syst. 16 (3) (2005) 781.
[25] Philip M.L.ong, Rocco A.S.ervedio, Random classification noise defeats all convex potential boosters, Mach. Learn. 78 (3) (2010) 287–304.
[26] Benoît Frénay, Michel Verleysen, Classification in the presence of label noise: a survey, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2013) 845–869.
[27] Ruxin Wang, Tongliang Liu, Dacheng Tao, Multiclass learning with partially corrupted labels, IEEE Trans. Neural Netw. Learn. Syst. 29 (6) (2017) 2568–2580.
[28] J.A.K. Suykens, J. De Brabanter, L. Lukas, J. Vandewalle, Weighted Least Squares Support Vector Machines: robustness and sparse approximation, Neurocomputing 48 (1) (2002) 85–105.
[29] Tongliang Liu, Dacheng Tao, Classification with noisy labels by importance reweighting, IEEE Trans. Pattern Anal. Mach. Intell. 38 (3) (2016) 447–461.
[30] József Valyon, Gábor Horváth, A weighted generalized LS-SVM, Period. Polytech. Electr. Eng. 47 (2003) 229–252.
[31] Lü You, Jizhen Liu, Yaxin Qu, A new robust least squares support vector machine for regression with outliers, Procedia Eng. 15 (2011) 1355–1360.
[32] Seyda Ertekin, Leon Bottou, C.L.ee Giles, Nonconvex online support vector machines, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2) (2011) 368–381.
[33] Jiajun Ma, Shuisheng Zhou, Li Chen, Weiwei Wang, Zhuan Zhang, A sparse robust model for large scale multi-class classification based on K-SVCR, Pattern Recognit. Lett. 117 (1) (2019) 16–23.
[34] Kuaini Wang, Ping Zhong, Robust non-convex least squares loss function for regression with outliers, Knowl. Based Syst. 71 (1) (2014) 290–302.
[35] Xiaowei Yang, Liangjun Tan, H.E. Lifang, A robust least squares support vector machine for regression and classification with noise, Neurocomputing 140 (2014) 41–52.
[36] Li Chen, Shuisheng Zhou, Sparse algorithm for robust LSSVM in primal space, Neurocomputing 275 (2018) 2880–2891.
[37] Chong Zhang, Minh Pham, Sheng Fu, Yufeng Liu, Robust multicategory support vector machines using difference convex algorithm, Math. Program. 169 (1) (2018) 277–305.
[38] Thi Hoai Le An, Dinh Pham Tao, Solving a class of linearly constrained indefinite quadratic problems by DC algorithms, J. Global Optim. 11 (3) (1997) 253–285.
[39] Chuanfa Chen, Changqing Yan, Yanyan Li, A robust weighted least squares support vector regression based on least trimmed squares, Neurocomputing 168 (2015) 941–946.
[40] X. Lu, W. Liu, C. Zhou, M. Huang, Robust least-squares support vector machine with minimization of mean and variance of modeling error, IEEE Trans. Neural Netw. Learn. Syst. 29 (7) (2018) 2909–2920.
[41] G.H. Golub, C.F. Van Loan, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 1996.
[42] Bernhard Schölkopf, Ralf Herbrich, Alex J. Smola, A generalized representer theorem, in: International Conference on Computational Learning Theory, 2001, pp. 416–426.
[43] Christopher K.I.W.illiams, Matthias Seeger, Using the Nyström method to speed up kernel machines, in: Neural Information Processing Systems, 2001, pp. 682–688.
[44] A.J. Smola, B. Schölkopf, P. Langley, Sparse greedy matrix approximation for machine learning, in: International Conference on Machine Learning, 2000, pp. 911–918.
[45] Ali Rahimi, Benjamin Recht, Random features for large-scale kernel machines, in: Neural Information Processing Systems, 2008, pp. 1177–1184.
[46] Joseph D.Conklin, Applied logistic regression, Technometrics 44 (1) (2002) 81–82.
[47] Cho Jui Hsieh, Kai Wei Chang, Chih Jen Lin, Sathiya S. Keerthi, S. Sundararajan, A dual coordinate descent method for large-scale linear SVM, in: International Conference on Machine Learning, 2008, pp. 408–415.
[48] Andrew Kachites Mccallum, Kamal Nigam, Jason Rennie, Kristie Seymore, Automating the construction of internet portals with machine learning, Inf. Retr. 3 (2) (2000) 127–163.
[49] Samer A. Nene, Shree K. Nayar, Hiroshi Murase, Columbia Object Image Library (COIL-20), Technical Report CUCS-005-96, Department of Computer Science, Columbia University, 1996.
[50] Harikrishna Narasimhan, Harish Ramaswamy, Aadirupa Saha, Shivani Agarwal, Consistent multiclass algorithms for complex performance measures, in: International Conference on Machine Learning, 2015, pp. 2398–2407.
[51] J. Gorodkin, Comparing two K-category assignments by a K-category correlation coefficient, Comput. Biol. Chem. 28 (5) (2004) 367–374.
[52] Janez Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.