An Improved LST-KSVC Based on Energy Model

Qing Ai, Fei Li, Qingyun Gao, and Fei Zhao

Abstract—**Least squares twin support vector classification for K-class (LST-KSVC) [1] is an efficient multiclass classifier that incorporates least squares strategy into twin support vector classification for K-class (Twin-KSVC) [2]. Because of its excellent classification performance, LST-KSVC has been applied in many fields. However, the LST-KSVC has some drawbacks: (1) It only implements empirical risk minimization (ERM), which reduces its generalization performance. (2) It is sensitive to noise and outliers. (3) The inverse matrices need to be calculated, which is impossible for many large-scale engineering problems. (4) For the nonlinear case, the LST-KSVC needs to reconstruct primal problems using the approximate kernel-generated surface (AKGS) and does not directly use kernel tricks as in the support vector machine (SVM) [3]. To address these shortcomings, an improved LST-KSVC based on energy model, which is called ELST-KSVC, is proposed in this paper. First, a regularization term is introduced into LST-KSVC to implement structural risk minimization (SRM). Second, energy parameters are introduced into LST-KSVC to reduce the effect of noise and outliers. Third, the dual problems are reconstructed to avoid inverse matrices. Furthermore, the sequential minimal optimization (SMO) algorithm is used to efficiently train subclassifiers. Finally, ELST-KSVC can directly use kernel tricks for nonlinear cases. Experimental results show that the ELST-KSVC has better generalization performance and higher learning speed.**

Index Terms—**multiclass classification, LST-KSVC, Twin-KSVC, SMO.**

I. Introduction

TWIN support vector machine (TSVM) [4] is an improved version of SVM [5, 6]. Unlike SVM, which seeks two parallel hyperplanes, TSVM aims to search for two nonparallel hyperplanes. Each hyperplane is close to the corresponding class and away from the others. Since TSVM only needs to solve two smaller-scale quadratic programming problems (QPPs), its learning speed is 4 times that of SVM in theory. Due to its good classification accuracy and high learning speed, the TSVM has become an effective tool in many engineering fields. Many improvements have been proposed, such as A-TSVM [7], PIFTSVMs [8], ITWSVM-DC [9], SQN-PTSVM [10], TSVM-PI [11], CTSVM [12], $CL_{2,p}$-LSTSVM [13], RCTSVM [14], LPTSVM [15], AULSTSVM [16], FULSTSVM [17], and CIL-FART-IFTSVM [18].

Manuscript received April 15, 2022; revised October 16, 2022.

This work was supported in part by the Natural Science Foundation of Liaoning Province in China (2020-MS-281).

Q. Ai is an associate professor of the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051 China (corresponding author, e-mail: lyaiqing@126.com).

F. Li is a graduate student of the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051 China (e-mail: ustllifei@163.com).

Q. Gao is a graduate student of the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051 China (e-mail: ustlgqy@163.com).

F. Zhao is a graduate student of the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051 China (e-mail: ustlzhaofei@163.com).

TSVM and its improved algorithms can only solve the binary-class classification problem, but many practical problems involve multiclass classification [19–30], such as industrial fault diagnosis and computer-aided diagnosis of diseases. Under the framework of SVM and TSVM, the decomposition-reconstruction strategy is generally used to solve the multiclass classification problem. The 1-a-r and 1-a-1 are two representative methods. The main idea of the 1-a-1 method is to construct a subclassifier for each pair of focus classes. In light of the prediction results of all subclassifiers, a voting strategy is used to determine the class of the new sample. Since the 1-a-1 method only involves two classes of samples when constructing each subclassifier and the other samples are omitted, it is possible to obtain unfavourable results. The main idea of the 1-a-r method is to construct each subclassifier using the corresponding class as the positive class and the remaining classes as the negative class. In light of the distance from the new sample to each hyperplane, the nearest one is the class to which the sample belongs. Since the 1-a-r method easily leads to class imbalance when constructing each subclassifier, it may obtain poor classification results.

To overcome the disadvantages of 1-a-1 SVM [31] and 1-a-r SVM [32], K-SVCR [33] was proposed, where each subclassifier divides the training set into three subsets for each pair of focus classes: positive samples, negative samples and remaining samples. The K-SVCR avoids the sample information loss in the 1-a-1 SVM and class imbalance in the 1-a-r SVM. Thus it has better generalization performance. Compared with the 1-a-r SVM and 1-a-1 SVM, K-SVCR needs to solve more and larger subclassifiers and therefore has a lower learning speed. Based on K-SVCR and TSVM, Xu et al. proposed Twin-KSVC [2, 34–36]. The experimental results show that Twin-KSVC has a higher learning speed. Inspired by LS-TSVM [37], Nasiri et al. proposed a least squares version of Twin-KSVC, named LST-KSVC [3, 38]. The LST-KSVC replaces inequality constraints in Twin-KSVC with equality constraints and hinge loss with quadratic loss. Because LST-KSVC only needs to solve nonlinear equations, instead of QPPs, LST-KSVC has a higher learning speed. However, there exist some disadvantages in the LST-KSVC. (1) it only implements the ERM, instead of the SRM. (2) It is sensitive to noise and outliers. (3) Inverse matrices need to be calculated in the LST-KSVC, which is impossible for many large-scale engineering problems. (4) For the nonlinear case, the LST-KSVC needs to reconstruct primal problems using the AKGS and not directly use kernel tricks as in the SVM. To overcome the disadvantages of LST-KSVC, we propose an improved algorithm, named ELST-KSVC. First, a regularization term is introduced into LST-KSVC to implement the SRM. Second, energy parameters are introduced into LST-KSVC to reduce the effect of noise and outliers. Third, the dual problems are reconstructed to avoid inverse matrices. Furthermore, the SMO algorithm is used to train subclassifiers efficiently. Finally, the dual problems in ELST-KSVC can directly apply kernel tricks for nonlinear cases.

The remaining parts of this paper are organized as follows: the related works are reviewed in Section II. In Section III, we discuss the ELST-KSVC in detail, including the linear case, nonlinear case, decision rule, fast training algorithm and convergence analysis. The experimental results are presented in Section IV. Section V presents the conclusions.

II. Related works

Assume a multiclass classification training dataset $D = \{X_i \in R^{l_i \times d} | i = 1, 2, \cdots, K\}$, where K is the number of classes, l_i is the number of samples of the ith class, and the total number of samples $l = l_1 + \cdots + l_K$.

A. K-SVCR

K-SVCR is an effective multiclass classification SVM that constructs $K(K-1)/2$ hyperplanes. Each subclassifier in K-SVCR exploits the 1-a-1-a-r strategy to evaluate training samples, i.e., each subclassifier divides the training set into three subsets: positive samples, negative samples and remaining samples, and the corresponding outputs are +1, -1, and 0, respectively. In light of the prediction results of all subclassifiers (i.e., +1, -1, 0), a voting strategy is used to assign the class of the new sample.

For the training samples of the ith and jth classes, K-SVCR aims to construct an optimal hyperplane

$$w_{ij}^T x + b_{ij} = 0, \tag{1}$$

where $b_{ij} \in R$ and $w_{ij} \in R^d$ are the bias and normal vector, respectively.

To obtain the hyperplane, K-SVCR solves the following QPP:

$$\begin{array}{l} \min \frac{1}{2} \|w_{ij}\|^2 + c_1(e_i^T \gamma_{ij} + e_j^T \gamma_{ij*}) + c_2 \bar{e}^T(\lambda_{ij} + \lambda_{ij*}), \\ s.t.\ (X_i w_{ij} + e_i b_{ij}) + \gamma_{ij} \geq e_i, \\ \quad -(X_j w_{ij} + e_j b_{ij}) + \gamma_{ij*} \geq e_j, \\ \quad -\bar{e}\sigma - \lambda_{ij*} < \bar{X} w_{ij} + \bar{e} b_{ij} \leq \bar{e}\sigma + \lambda_{ij}, \\ \quad \gamma_{ij} \geq 0e_i, \gamma_{ij*} \geq 0e_j, \lambda_{ij} \geq 0\bar{e}, \lambda_{ij*} \geq 0\bar{e}, \end{array} \tag{2}$$

where $\bar{X} = D - X_i - X_j$, $\gamma_{ij}, \gamma_{ij*}$, λ_{ij} and λ_{ij*} are slack variables, $\sigma \in [0, 1)$ is the preset bandwidth parameter, and e_i, e_j and $\bar{e}$ are all 1 vectors of appropriate dimensions.

For a new sample x, if subclassifier $g^{ij}(x) = w_{ij}^T x + b_{ij} > \sigma$, a vote is given to the ith class; if subclassifier $g^{ij}(x) < -\sigma$, a vote is given to the jth class; if these conditions are not met, a vote is decreased for the ith and jth classes. K-SVCR counts the votes of all subclassifiers, and the new sample is assigned to the label that has the most votes.

B. Twin-KSVC

Twin-KSVC is an improvement of K-SVCR, which integrates TSVM and the 1-a-1-a-r strategy. Twin-KSVC constructs $K(K-1)/2$ pairs of subclassifiers. For the ith and jth classes, Twin-KSVC aims to seek a pair of nonparallel hyperplanes as follows:

$$w_i^T x + b_i = 0 \text{ and } w_j^T x + b_j = 0, \tag{3}$$

where $b_i(b_j) \in R$ and $w_i(w_j) \in R^d$ are the bias and normal vector of the $ith(jth)$ class, respectively.

To obtain the above nonparallel hyperplanes, Twin-KSVC solves the following QPPs:

$$\begin{array}{l} \min \frac{1}{2} \|X_i w_i + e_i b_i\|^2 + c_1 e_j^T \gamma_i + c_2 \bar{e}^T \lambda_i, \\ s.t.\ -(X_j w_i + e_j b_i) + \gamma_i \geq e_j, \\ \quad -(\bar{X} w_i + \bar{e} b_i) + \lambda_i \geq \bar{e}(1-\sigma), \\ \quad \gamma_i \geq 0e_j, \lambda_i \geq 0\bar{e}, \end{array} \tag{4}$$

and

$$\begin{array}{l} \min \frac{1}{2} \|X_j w_j + e_j b_j\|^2 + c_3 e_i^T \gamma_j + c_4 \bar{e}^T \lambda_j, \\ s.t.\ (X_j w_j + e_i b_j) + \gamma_j \geq e_i, \\ \quad -(\bar{X} w_j + \bar{e} b_j) + \lambda_j \geq \bar{e}(1-\sigma), \\ \quad \gamma_j \geq 0e_i, \lambda_j \geq 0e, \end{array} \tag{5}$$

where $\bar{X} = D - X_i - X_j$, γ_i, γ_j, λ_i and λ_j are slack variables.

For a new sample x, if subclassifier $g^i(x) = w_i^T x + b_i > -1 + \sigma$, a vote is given to the ith class; if subclassifier $g^j(x) = w_j^T x + b_j < 1 - \sigma$, a vote is given to the jth class; if these conditions are not met, a vote is decreased for the ith and jth classes. Twin-KSVC counts the votes of all subclassifiers, and the new sample is assigned to the label that has the most votes.

C. LST-KSVC

LST-KSVC is an improvement of Twin-KSVC, which integrates LS-TSVM and the 1-a-1-a-r strategy. Different from Twin-KSVC, LST-KSVC replaces inequality constraints with equality constraints and hinge loss with quadratic loss, such that it only needs to solve nonlinear equations rather than QPPs. To obtain the above nonparallel hyperplanes, LST-KSVC solves the following QPPs:

$$\begin{array}{l} \min \frac{1}{2} \|X_i w_i + e_i b_i\|^2 + \frac{c_1}{2} \gamma_i^T \gamma_i + \frac{c_2}{2} \lambda_i^T \lambda_i, \\ s.t.\ -(X_j w_i + e_j b_i) + \gamma_i = e_j, \\ \quad -(\bar{X} w_i + \bar{e} b_i) + \lambda_i = \bar{e}(1-\sigma), \end{array} \tag{6}$$

and

$$\begin{array}{l} \min \frac{1}{2} \|X_j w_j + e_j b_j\|^2 + \frac{c_3}{2} \gamma_j^T \gamma_j + \frac{c_4}{2} \lambda_j^T \lambda_j, \\ s.t.\ (X_j w_j + e_i b_j) + \gamma_j = e_i, \\ \quad -(\bar{X} w_j + \bar{e} b_j) + \lambda_j = \bar{e}(1-\sigma), \end{array} \tag{7}$$

For a new sample, similar to Twin-KSVC, a voting strategy is used to assign the final class.

III. ELST-KSVC

As discussed in the previous sections, LST-KSVC is a valuable extension of Twin-KSVC. The advantage of LST-KSVC compared with Twin-KSVC is that LST-KSVC has a higher learning speed. In this section, to address the disadvantages of LST-KSVC, we propose ELST-KSVC.

A. Linear Case

Due to the fact that LST-KSVC, only implements the ERM rather than the SRM and is sensitive to noise and outliers, we introduce a regularization term into LST-KSVC to implement the SRM and energy parameters into LST-KSVC to reduce the effect of noise and outliers. The primal problems are modified as follows:

$$\begin{array}{l}\min \frac{c_5}{2}\left\|w_i\right\|^2+\frac{1}{2}\gamma_i^T\gamma_i+\frac{c_1}{2}\lambda_i^T\lambda_i+\frac{c_2}{2}\theta_i^T\theta_i,\\ s.t.\ X_iw_i+e_ib_i=\gamma_i,\\ \quad -(X_jw_i+e_jb_i)+\lambda_i=E_1e_j,\\ \quad -(\bar{X}w_i+\bar{e}b_i)+\theta_i=(E_1-\sigma)\,\bar{e},\end{array} \tag{8}$$

and

$$\begin{array}{l}\min \frac{c_6}{2}\left\|w_j\right\|^2+\frac{1}{2}\gamma_j^T\gamma_j+\frac{c_3}{2}\lambda_j^T\lambda_j+\frac{c_4}{2}\theta_j^T\theta_j,\\ s.t.\ X_jw_j+e_jb_j=\gamma_j,\\ \quad (X_iw_j+e_ib_j)+\lambda_j=E_2e_i,\\ \quad -(\bar{X}w_j+\bar{e}b_j)+\theta_j=(E_2-\sigma)\,\bar{e},\end{array} \tag{9}$$

where $c_k(k=1,2,\cdots,6)$ are penalty parameters and $\lambda_i(\lambda_j)$ and $\theta_i(\theta_j)$ are slack variables.

Different from (6) and (7) in the LST-KSVC, the objective functions in (8) and (9) add regularization terms ($\|w_i\|^2$ and $\|w_j\|^2$) such that ELST-KSVC implements the SRM. In addition, (8) and (9) introduce the energy parameters (E_1 and E_2) in the constraints such that ELST-KSVC is robust for noise and outliers.

The Lagrangian of (8) is given by:

$$\begin{array}{l}L=\frac{c_5}{2}\left\|w_i\right\|^2+\frac{1}{2}\gamma_i^T\gamma_i+\frac{c_1}{2}\lambda_i^T\lambda_i+\frac{c_2}{2}\theta_i^T\theta_i\\ \quad +\alpha^T\left(\gamma_i-X_iw_i-e_ib_i\right)\\ \quad +p^T\left(\lambda_i-(X_jw_i+e_jb_i)-E_1e_j\right)\\ \quad +q^T(\theta_i-(\bar{X}w_i+\bar{e}b_i)+\theta_i-(E_1-\sigma)\,\bar{e}),\end{array} \tag{10}$$

where α, p and q are Lagrangian multiplier vectors. According to Karush-Kuhn-Tucker (KKT) conditions, we can obtain

$$\frac{\partial L}{\partial w_i}=c_5w_i-X_i{}^T\alpha-X_j{}^Tp-\bar{X}^Tq=0, \tag{11}$$

$$\frac{\partial L}{\partial b_i}=-e_i^T\alpha-e_j^Tp-\bar{e}^Tq=0, \tag{12}$$

$$\frac{\partial L}{\partial \gamma_i}=\gamma_i+\alpha=0, \tag{13}$$

$$\frac{\partial L}{\partial \lambda_i}=c_1\lambda_i+p=0, \tag{14}$$

$$\frac{\partial L}{\partial \theta_i}=c_2\theta_i+q=0. \tag{15}$$

From (11) and (12), we can obtain

$$w_i=\frac{1}{c_5}\begin{pmatrix}X_i{}^T & X_j{}^T & \bar{X}^T\end{pmatrix}\begin{pmatrix}\alpha\\ p\\ q\end{pmatrix}, \tag{16}$$

$$\begin{pmatrix}e_i^T & e_j^T & \bar{e}^T\end{pmatrix}\begin{pmatrix}\alpha\\ p\\ q\end{pmatrix}=0. \tag{17}$$

According to (13)-(17), we can obtain the following dual optimization problem of (8):

$$\begin{array}{l}\max -\frac{1}{2}\begin{pmatrix}\alpha^T & p^T & q^T\end{pmatrix}\bar{Q}\begin{pmatrix}\alpha\\ p\\ q\end{pmatrix}\\ \quad -c_5\begin{pmatrix}0e_i^T & E_1e_j^T & (E_1-\sigma)\bar{e}^T\end{pmatrix}\begin{pmatrix}\alpha\\ p\\ q\end{pmatrix},\\ s.t.\ \begin{pmatrix}e_i^T & e_j^T & \bar{e}^T\end{pmatrix}\begin{pmatrix}\alpha\\ p\\ q\end{pmatrix}=0,\end{array} \tag{18}$$

where

$$\bar{Q}=\begin{pmatrix}X_iX_i{}^T+c_5I & X_iX_j{}^T & X_i\bar{X}^T\\ X_jX_i{}^T & X_jX_j{}^T+\frac{c_5}{c_1}I & X_j\bar{X}^T\\ \bar{X}^TX_i{}^T & \bar{X}^TX_j{}^T & \bar{X}^T\bar{X}^T+\frac{c_5}{c_2}I\end{pmatrix}. \tag{19}$$

The bias b_i can be calculated by

$$b_i=\frac{1}{l}\sum_{k=1}^{l}[-\frac{1}{c_5}\bar{Q}\begin{pmatrix}\alpha\\ p\\ q\end{pmatrix}-\begin{pmatrix}0e_i\\ E_1e_j\\ (E_1-\sigma)\,\bar{e}^T\end{pmatrix}]_k. \tag{20}$$

Similarly, we can obtain the following dual optimization problem of (9):

$$\begin{array}{l}\max -\frac{1}{2}\begin{pmatrix}\beta^T & m^T & n^T\end{pmatrix}\hat{Q}\begin{pmatrix}\beta\\ m\\ n\end{pmatrix}\\ \quad -c_6\begin{pmatrix}0e_j^T & E_2e_i^T & (E_2-\sigma)\bar{e}^T\end{pmatrix}\begin{pmatrix}\beta\\ m\\ n\end{pmatrix},\\ s.t.\ \begin{pmatrix}e_j^T & e_i^T & \bar{e}^T\end{pmatrix}\begin{pmatrix}\beta\\ m\\ n\end{pmatrix}=0,\end{array} \tag{21}$$

where β, m and n are Lagrangian multiplier vectors, and

$$\hat{Q}=\begin{pmatrix}X_jX_j{}^T+c_6I & X_jX_i{}^T & X_j\bar{X}^T\\ X_iX_j{}^T & X_iX_i{}^T+\frac{c_6}{c_3}I & AC^T\\ \bar{X}X_j{}^T & \bar{X}X_i{}^T & \bar{X}\bar{X}^T+\frac{c_6}{c_4}I\end{pmatrix}. \tag{22}$$

w_j and b_j can be calculated by

$$w_j=-\frac{1}{c_6}\begin{pmatrix}X_j{}^T & X_i{}^T & \bar{X}^T\end{pmatrix}\begin{pmatrix}\beta\\ m\\ n\end{pmatrix} \tag{23}$$

and

$$b_j=\frac{1}{l}\sum_{k=1}^{l}[-\frac{1}{c_6}\hat{Q}\begin{pmatrix}\beta\\ m\\ n\end{pmatrix}-\begin{pmatrix}0e_j\\ E_2e_i\\ (E_2-\sigma)\,\bar{e}^T\end{pmatrix}]_k. \tag{24}$$

B. Nonlinear Case

For the nonlinear case, we directly introduce the mapping ϕ from R^d to a high dimensional space H instead of the AKGS in LST-KSVC. The primal problems in nonlinear ELST-KSVC are as follows:

$$\begin{array}{l}\min \frac{c_5}{2}\left\|w_i\right\|^2+\frac{1}{2}\gamma_i^T\gamma_i+\frac{c_1}{2}\lambda_i^T\lambda_i+\frac{c_2}{2}\theta_i^T\theta_i,\\ s.t.\ \phi\left(X_i\right)w_i+e_ib_i=\gamma_i,\\ \quad -(\phi\left(X_j\right)w_i+e_jb_i)+\lambda_i=E_1e_j,\\ \quad -(\phi\left(\bar{X}\right)w_i+\bar{e}b_i)+\theta_i=(E_1-\sigma)\,\bar{e},\end{array} \tag{25}$$

and

$$\begin{array}{l}\min \frac{c_6}{2}\left\|w_j\right\|^2+\frac{1}{2}\gamma_j^T\gamma_j+\frac{c_3}{2}\lambda_j^T\lambda_j+\frac{c_4}{2}\theta_j^T\theta_j,\\ s.t.\ \phi(X_j)w_j+e_jb_j=\gamma_j,\\ \quad (\phi(X_i)w_j+e_ib_j)+\lambda_j=E_2e_i,\\ \quad -(\phi(\bar{X})w_j+\bar{e}b_j)+\theta_j=(E_2-\sigma)\,\bar{e},\end{array} \tag{26}$$

Similarly, we can obtain the following dual optimization problems of (25) and (26):

$$\begin{aligned}&\max -\tfrac{1}{2}\begin{pmatrix}\alpha^T & p^T & q^T\end{pmatrix}\bar{Q}\begin{pmatrix}\alpha\\ p\\ q\end{pmatrix}\\ &\qquad -c_5\begin{pmatrix}0e_1^T & E_1e_2^T & (E_1-\sigma)e_3^T\end{pmatrix}\begin{pmatrix}\alpha\\ p\\ q\end{pmatrix},\\ &s.t.\ \begin{pmatrix}e_1^T & e_2^T & e_3^T\end{pmatrix}\begin{pmatrix}\alpha\\ p\\ q\end{pmatrix}=0,\end{aligned} \tag{27}$$

and

$$\begin{aligned}&\max -\tfrac{1}{2}\begin{pmatrix}\beta^T & m^T & n^T\end{pmatrix}\hat{Q}\begin{pmatrix}\beta\\ m\\ n\end{pmatrix}\\ &\qquad -c_6\begin{pmatrix}0e_2^T & E_2e_1^T & (E_2-\sigma)e_3^T\end{pmatrix}\begin{pmatrix}\beta\\ m\\ n\end{pmatrix},\\ &s.t.\ \begin{pmatrix}e_2^T & e_1^T & e_3^T\end{pmatrix}\begin{pmatrix}\beta\\ m\\ n\end{pmatrix}=0,\end{aligned} \tag{28}$$

where

$$\bar{Q}=\begin{pmatrix}K\left(X_i,X_i^T\right)+c_5I & K\left(X_i,X_j^T\right) & K\left(X_i,\bar{X}^T\right)\\ K\left(X_j,X_i^T\right) & K\left(X_j,X_j^T\right)+\frac{c_5}{c_1}I & K\left(X_j,\bar{X}^T\right)\\ K\left(\bar{X},X_i^T\right) & K\left(\bar{X},X_j^T\right) & K\left(\bar{X},\bar{X}^T\right)+\frac{c_5}{c_2}I\end{pmatrix} \tag{29}$$

and

$$\hat{Q}=\begin{pmatrix}K\left(X_j,X_j^T\right)+c_6I & K\left(X_j,X_i^T\right) & K\left(X_j,\bar{X}^T\right)\\ K\left(X_i,X_j^T\right) & K\left(X_i,X_i^T\right)+\frac{c_6}{c_3}I & K\left(X_i,\bar{X}^T\right)\\ K\left(\bar{X},X_j^T\right) & K\left(\bar{X},X_i^T\right) & K\left(\bar{X},\bar{X}^T\right)+\frac{c_6}{c_4}I\end{pmatrix}. \tag{30}$$

Obviously, (27) and (28) can degenerate to (18) and (21) by using a linear kernel.

C. Decision Rule

For each pair of subclassifiers in ELST-KSVC, a new sample x is labelled by

$$f(x)=\begin{cases}1, & \text{if } \frac{1}{c_5}\left(K\left(x,X_i^T\right)\alpha+K\left(x,X_j^T\right)p+K\left(x,\bar{X}^T\right)q\right)\\ & \quad +b_i>E_1+\sigma\\ -1, & \text{if } -\frac{1}{c_6}\left(K\left(x,X_j^T\right)\beta+K\left(x,X_i^T\right)m+K\left(x,\bar{X}^T\right)n\right).\\ & \quad +b_j<E_2-\sigma\\ 0, & \text{otherwise}\end{cases} \tag{31}$$

We use a voting strategy to assign the final label of x. If subclassifier $g^i(x)=w_i{}^T\cdot x+b_i>E_1+\sigma$, a vote is given to the ith class; if subclassifier $g^j(x)=w_j{}^T\cdot x+b_j<E_2-\sigma$, a vote is given to the jth class; if these conditions are not met, a vote is decreased for the ith and jth classes. All subclassifiers are counted, and x is assigned to the class that has the most votes.

D. Fast Solvers

LST-KSVC and its improved algorithms cannot solve large-scale problems. To handle large-scale problems, in this subsection, we use the SMO algorithm to solve (27) and (28). The optimization problems (27) and (28) can be rewritten in a unified form as follows:

$$\begin{aligned}&\max\left\{D(x)=-\tfrac{1}{2}x^TQx+p^Tx\right\},\\ &s.t.\ e^Tx=0,\end{aligned} \tag{32}$$

where Q is a positive definite matrix. For instance, the unified form (32) can be converted to (27) when $Q=Q$, $p=-c_5(0e_1^TE_1e_2^T(E_1-\sigma)e_3^T)^T$, while the unified form (32) can be converted to (28) when $Q=\hat{Q}$, $p=-c_6(0e_2^TE_2e_1^T(E_2-\sigma)e_3^T)^T$.

The Lagrangian of (32) is given by:

$$L=-\frac{1}{2}x^TQx+p^Tx+\eta e^Tx. \tag{33}$$

Define

$$F=-Qx+p. \tag{34}$$

The KKT condition for (32) is

$$F+\eta e=0. \tag{35}$$

The QPP (32) is solved by the following SMO algorithm:

Step 1: Set $k=0$, $x^k=0e$ and $F^k=p$.

Step 2: If the stop criterion is satisfied, stop; otherwise, $i_1=\arg\max_i(F_i^k)$ and $i_2=\arg\min_i(F_i^k)$.

Step 3: Solve the following suboptimization problem

$$\begin{aligned}&z^{opt}=\\ &\arg\max_z\left\{-\tfrac{1}{2}\begin{bmatrix}-z\\ z\end{bmatrix}^T\begin{bmatrix}Q_{i_1i_1}Q_{i_1i_2}\\ Q_{i_2i_1}Q_{i_2i_2}\end{bmatrix}\begin{bmatrix}-z\\ z\end{bmatrix}+\begin{bmatrix}-z\\ z\end{bmatrix}^T\begin{bmatrix}F_{i_1}^k\\ F_{i_2}^k\end{bmatrix}\right\}\\ &\quad=\frac{F_{i_2}^k-F_{i_1}^k}{Q_{i_1i_1}+Q_{i_2i_2}-2Q_{i_1i_2}}.\end{aligned} \tag{36}$$

Step 4: Set $x_{i_1}^{k+1}=x_{i_1}^k-z^{opt}$, $x_{i_2}^{k+1}=x_{i_2}^k+z^{opt}$, $F_i^{k+1}=F_i^k+z^{opt}Q_{ii_1}-z^{opt}Q_{ii_2}(i\in 1,\cdots,l)$ and $k=k+1$. Goto Step 2.

E. Convergence Analysis

Lemma 1: $\left\{x^k\right\}$ converges to the global solution of (32).

Proof: From (36), we can obtain

$$\begin{aligned}&D\left(x^{k+1}\right)-D\left(x^k\right)=\\ &\max_z\left\{-\tfrac{1}{2}\begin{bmatrix}-z\\ z\end{bmatrix}^T\begin{bmatrix}Q_{i_1i_1}Q_{i_1i_2}\\ Q_{i_2i_1}Q_{i_2i_2}\end{bmatrix}\begin{bmatrix}-z\\ z\end{bmatrix}+\begin{bmatrix}-z\\ z\end{bmatrix}^T\begin{bmatrix}F_{i_1}^k\\ F_{i_2}^k\end{bmatrix}\right\}\\ &\quad=\frac{\left(z^{opt}\right)^2\left(Q_{i_1i_1}+Q_{i_2i_2}-2Q_{i_1i_2}\right)}{2}.\end{aligned} \tag{37}$$

We can observe that $Q_{i_1i_1}+Q_{i_2i_2}-2Q_{i_1i_2}>C$,

$$C=\begin{cases}2c_5 & \text{if } x_{i_1}\subseteq X_i \text{ and } x_{i_2}\subseteq X_i\\ 2\frac{c_5}{c_1} & \text{if } x_{i_1}\subseteq X_j \text{ and } x_{i_2}\subseteq X_j\\ 2\frac{c_5}{c_2} & \text{if } x_{i_1}\subseteq \bar{X} \text{ and } x_{i_2}\subseteq \bar{X}\\ c_5+\frac{c_5}{c_1}, & \text{if } x_{i_1}\subseteq X_i \text{ and } x_{i_2}\subseteq X_j\\ c_5+\frac{c_5}{c_2}, & \text{if } x_{i_1}\subseteq X_i \text{ and } x_{i_2}\subseteq \bar{X}\\ \frac{c_5}{c_1}+\frac{c_5}{c_2}, & \text{if } x_{i_1}\subseteq X_j \text{ and } x_{i_2}\subseteq \bar{X}\end{cases}, \text{ when } Q=\bar{Q}.$$

From $Q_{i_1i_1}+Q_{i_2i_2}-2Q_{i_1i_2}>C$ and $\left(z^{opt}\right)^2=\frac{\left\|x^{k+1}-x^k\right\|_2^2}{2}$, (37) implies that D is increased at each step, i.e.,

$$D\left(x^{k+1}\right)-D\left(x^k\right)\geq\frac{C\left\|x^{k+1}-x^k\right\|_2^2}{4} \tag{38}$$

According to the Wolf duality [39], D is bounded above. Thus, $\left\{D\left(x^k\right)\right\}$ is a convergent sequence. From (38), we can obtain that $\left\{x^{k+1}-x^k\right\}$ converges to 0.

Because D is a positive definite quadratic form, the set $\left\{x \mid D(x)\geq D\left(x^0\right)\right\}$ is a compact set. Because $\left\{x^k\right\}$ lies in $\left\{x \mid D(x) \geq D\left(x^0\right)\right\}$, it is a bounded sequence. Let $\{x^{n_k}\}_{k\in\mathbb{N}}$ be a convergent subsequence of $\left\{x^k\right\}$ and $\bar{x}$ be the limit point of subsequence.

For $\forall\ k \geq 0$, let $\mu(k) = \arg\max_i\left(F_i\left(x^k\right)\right)$ and $\nu(k) = \arg\min_i\left(F_i\left(x^k\right)\right)$ at the kth step. Since there is only a finite number of variables, for $\forall\ k \geq 0$, there exists at least a pair of variables $i_1 = \mu(n_k)$ and $i_2 = \nu(n_k)$ in this subsequence. Because F_{i_1} and F_{i_2} are continuous functions of x, we have

$$F_{i_1}(\bar{x}) - F_{i_2}(\bar{x}) = lim_{k\to\infty}\left(F_{i_1}(x^{n_k}) - F_{i_2}(x^{n_k})\right). \quad (39)$$

(39) can be decomposed into $F_{i_1}(\bar{x}) - F_{i_2}(\bar{x}) = lim_{k\to\infty}\left(\Upsilon_1(k) + \Upsilon_2(k)\Upsilon_3(k)\right)$, where $\Upsilon_1(k) = F_{i_1}(x^{n_k}) - F_{i_1}\left(x^{n_k+1}\right)$, $\Upsilon_2(k) = F_{i_1}\left(x^{n_k+1}\right) - F_{i_2}\left(x^{n_k+1}\right)$ and $\Upsilon_3(k) = F_{i_2}\left(x^{n_k+1}\right) - F_{i_2}(x^{n_k})$.

Since $\left\{x^{k+1} - x^k\right\}$ converges to 0, $lim_{k\to\infty}\Upsilon_1(k) = 0$ and $lim_{k\to\infty}\Upsilon_3(k) = 0$. From step 4 of the SMO algorithm, we obtain $lim_{k\to\infty}\Upsilon_2(k) = 0$. Thus, $F_{i_1}(\bar{x}) - F_{i_2}(\bar{x}) = 0$. For $\forall i, j \in \{1, \cdots, l\}$,

$$\begin{aligned} &\left(F_i(\bar{x}) - F_j(\bar{x})\right)^2 \\ &= lim_{k\to\infty}\left(F_i(x^{n_k}) - F_j(x^{n_k})\right)^2 \leq \\ &\frac{Q_{ii}+Q_{jj}-2Q_{ij}}{Q_{i_1i_1}+Q_{i_2i_2}-2Q_{i_1i_2}}\left(lim_{k\to\infty}(F_{i_1}(x^{n_k}) - F_{i_2}(x^{n_k}))\right)^2 \\ &= \frac{Q_{ii}+Q_{jj}-2Q_{ij}}{Q_{i_1i_1}+Q_{i_2i_2}-2Q_{i_1i_2}}\left(F_{i_1}(\bar{x}) - F_{i_2}(\bar{x})\right) = 0. \end{aligned} \quad (40)$$

From (40), we observe that $F_1(\bar{x}) = F_2(\bar{x}) = \ldots = F_l(\bar{x})$. According to the KKT conditions, $\bar{x}$ is the global optimal solution of (32). Since D is strictly convex, $\left\{x^k\right\}$ converges to $\bar{x}$.

IV. Experiments

To investigate the effectiveness of ELST-KSVC, we compared ELST-KSVC with the state-of-the-art multiclass algorithms on multiple datasets.

A. Experimental Configuration

1) Datasets: We performed experiments on eight multiclass benchmark datasets. The experimental datasets are listed in Table IV, where #Sample, #Feature and #Label represent the number of samples, the number of features and the number of labels, respectively.

2) Comparative Algorithms:

1) K-SVCR [33]: This is a classical multiclass classification algorithm that combines the 1-a-1-a-r strategy and SVM. It has three parameters, namely, c_1, c_2 and σ, where c_1 and c_2 are selected from $\left\{2^{-8}, 2^{-7}, \ldots, 2^7\right\}$, and σ is selected from $\{0,\ 0.1,\ 0.2\}$. In our experiments, we set $c_1 = c_2$.
2) Twin-KSVC [2]: This is an improvement of K-SVCR, which combines the 1-a-1-a-r strategy and TSVM. It has five parameters, c_1, c_2, c_3, c_4 and σ, where c_1, c_2, c_3 and c_4 are selected from $\left\{2^{-8}, 2^{-7}, \ldots, 2^7\right\}$, and σ is selected from $\{0,\ 0.1,\ 0.2\}$. In our experiments, we set $c_1 = c_3$ and $c_2 = c_4$.
3) LST-KSVC [1]: This is the least squares version of Twin-KSVC. It has five parameters, c_1, c_2, c_3, c_4 and σ, where c_1, c_2, c_3 and c_4 are selected from $\left\{2^{-8}, 2^{-7}, \ldots, 2^7\right\}$, and σ is selected from $\{0,\ 0.1,\ 0.2\}$. In our experiments, we set $c_1 = c_3$ and $c_2 = c_4$.
4) Improvement on LST-KSVC (ILST-KSVC) [40]: This is an improvement of LST-KSVC. It has seven parameters, c_1, c_2, c_3, c_4, c_5, c_6 and σ, where c_1, c_2, c_3, c_4, c_5 and c_6 are selected from $\left\{2^{-8}, 2^{-7}, \ldots, 2^7\right\}$, and σ is selected from $\{0,\ 0.1,\ 0.2\}$. In our experiments, we set $c_1 = c_3$, $c_2 = c_4$ and $c_5 = c_6$.
5) ELST-KSVC: The penalty parameters c_1, c_2, c_3, c_4, c_5 and c_6 are selected from $\left\{2^{-8}, 2^{-7}, \ldots, 2^7\right\}$. The preset bandwidth parameter σ is selected from $\{0,\ 0.1,\ 0.2\}$. The energy parameters E_1 and E_2 are selected from $\{0.6,\ 0.7,\ \ldots,\ 1\}$. In our experiments, we set $c_1 = c_3$, $c_2 = c_4$, $c_5 = c_6$ and $E_1 = E_2$.

In the experiment, the Gaussian kernel function $K(x_i,\ x_j) = e^{-\frac{\|x_i - x_j\|^2}{\gamma^2}}$ was employed, where the parameter γ was selected from $\left\{2^{-8}, 2^{-7}, \ldots, 2^7\right\}$.

B. Comparison of Experimental Results

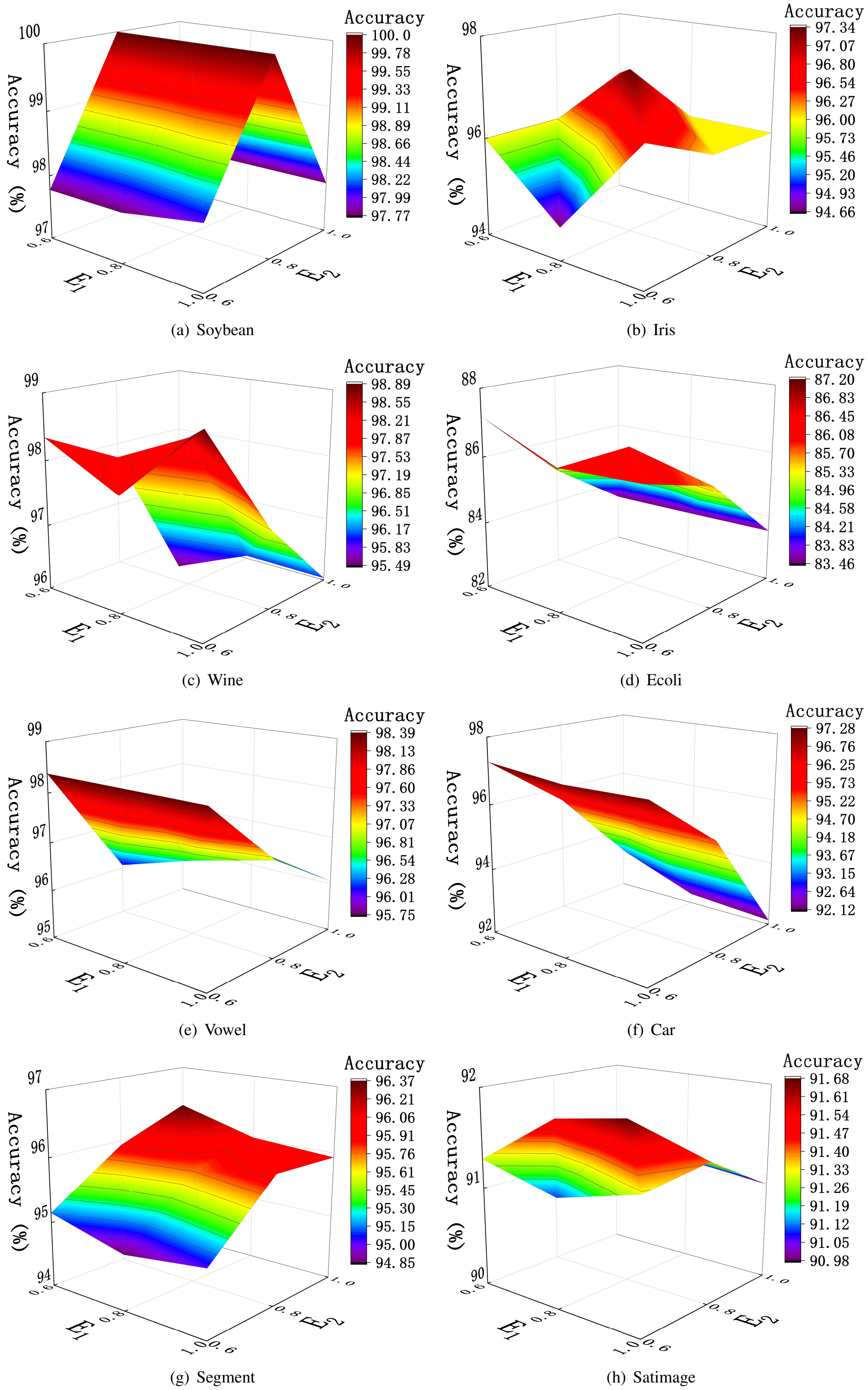
Experiments were conducted on a server with an Intel Xeon processor (2.5 GHz) and 32 GB RAM.

The 5-fold cross-validation is employed to evaluate each algorithm. We report the experimental results in Table I, where the best result for each dataset is highlighted in bold. From Table I, we can observe that our ELST-KSVC is better than each comparative algorithm on all datasets in terms of running time and is superior to Twin-KSVC, LST-KSVC and ILST-KSVC on most datasets in terms of accuracy.

To evaluate these classifiers more systematically, we employed the Friedman test [41] to analyse the accuracy of these classifiers. $\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4}\right]$, where k is the number of comparative algorithms, N is the number of datasets, and R_j is the average rank of the jth classifier on all datasets. Friedman statistic $F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \sim F(k-1, (k-1)(N-1))$. In Table II, we show the accuracy rank of five algorithms on eight benchmark datasets. From Table II, we notice that the average rank of ELST-KSVC is better than that of Twin-KSVC, LST-KSVC and ILST-KSVC for accuracy. The Friedman statistics F_F of accuracy is 8.887, and the corresponding critical value is 2.714 at the significance level $\alpha = 0.05$. Because the Friedman statistic F_F is greater than the critical value at $\alpha = 0.05$, these algorithms are significantly different in terms of accuracy. We employed the Nemenyi test [41] to further analyse whether ELST-KSVC has better accuracy than other classifiers. The critical difference (CD $= q_\alpha\sqrt{\frac{k(k+1)}{6N}}$) was employed to compare the average rank difference of accuracy between ELST-KSVC and a comparative classifier. For the Nemenyi test, at $\alpha = 0.05$, we obtained $q_\alpha = 2.948$ and CD $= 2.157(k = 5,\ \ N = 8)$. We show the average rank difference of accuracy between ELST-KSVC and other algorithms in Table III. When the average rank difference of accuracy between two algorithms is within one CD, they cannot be considered to be significantly different. As presented in Table III, we notice that the difference between ELST-KSVC and other comparative algorithms is not significantly different.

According to the experimental results, we can draw the conclusions as follows:

1) Because ILST-KSVC implements the SRM, it is better than LST-KSVC and Twin-KSVC for accuracy.



(a) Soybean (b) Iris (c) Wine (d) Ecoli (e) Vowel (f) Car (g) Segment (h) Satimage

Fig. 1. The performance of ELST-KSVC changes on all evaluation metrics as the value of energy parameters increases on eight benchmark multiclass datasets.

TABLE I
EXPERIMENTAL RESULTS OF EACH COMPARATIVE ALGORITHMS WITH RBF KERNEL

Dataset	K-SVCR Acc ± std Time (s)	Twin-KSVC Acc ± std Time (s)	LST-KSVC Acc ± std Time (s)	ILST-KSVC Acc ± std Time (s)	ELST-KSVC Acc ± std Time (s)
Soybean	99.16 ± 1.89 0.074	96.18 ± 5.15 0.178	97.02 ± 6.27 0.020	96.98 ± 6.03 0.019	**99.56 ± 0.99** **0.010**
Iris	95.47 ± 3.81 0.132	92.00 ± 5.39 0.195	95.13 ± 4.34 0.049	**95.60 ± 3.70** 0.065	95.07 ± 4.58 **0.014**
Wine	**98.54 ± 1.62** 0.213	97.31 ± 2.53 0.217	96.18 ± 3.40 0.081	97.09 ± 3.57 0.076	97.76 ± 2.65 **0.037**
Ecoli	**88.56 ± 3.64** 3.163	86.56 ± 4.22 1.709	85.76 ± 3.41 0.486	85.93 ± 3.67 0.518	86.12 ± 3.38 **0.246**
Vowel	**98.75 ± 0.70** 185.742	93.88 ± 1.99 776.113	94.89 ± 1.84 30.616	96.73 ± 1.41 31.243	98.38 ± 1.04 **13.113**
Car	**98.43 ± 0.54** 43.895	97.48 ± 0.77 41.684	94.48 ± 1.25 11.909	97.19 ± 0.77 11.268	98.03 ± 0.86 **2.598**
Segment	**96.94 ± 0.79** 498.903	90.91 ± 0.95 316.283	95.56 ± 0.96 72.219	94.67 ± 1.03 73.666	95.01 ± 0.65 **30.215**
Satimage	**91.71 ± 1.06** 1753.600	85.31 ± 0.86 980.628	84.03 ± 0.74 311.736	88.88 ± 0.74 312.722	91.31 ± 0.83 **137.863**

TABLE II
THE RANK OF EACH COMPARATIVE ALGORITHM FOR ACCURACY

Dataset	K-SVCR	Twin-KSVC	LST-KSVC	ILST-KSVC	ELST-KSVC
Soybean	2	5	3	4	**1**
Iris	2	5	3	**1**	4
Wine	**1**	3	5	4	2
Ecoli	**1**	2	5	4	3
Vowel	**1**	5	4	3	2
Car	**1**	3	5	4	2
Segment	**1**	5	2	4	3
Satimage	**1**	4	5	3	2
Average	**1.25**	4	4	3.375	2.375

TABLE III
THE AVERAGE RANK DIFFERENCE OF ACCURACY BETWEEN ELST-KSVC AND A COMPARATIVE ALGORITHM

Algorithm	Difference	CD
K-SVCR	1.125	
Twin-KSVC	1.625	2.157
LST-KSVC	1.625	
ILST-KSVC	1	

TABLE IV
DESCRIPTION OF DATASETS

Dataset	#Sample	#Feature	#Label
Soybean	47	35	4
Iris	150	4	3
Wine	178	13	3
Ecoli	327	7	5
Vowel	990	11	11
Car	1728	6	4
Segment	2310	20	7
Satimage	4435	37	6

2) ELST-KSVC outperforms ILST-KSVC for accuracy, because ELST-KSVC not only implements the SRM principle, but also reduces the effect of noise and outliers effectively.
3) Because ELST-KSVC avoids inverse matrices, and uses SMO to solve, it is faster than other comparative algorithms on all datasets.

C. Parameter Sensitivity Analysis

Energy parameters E_1 and E_2 are used to reduce the effect of noise and outliers. The effects of E_1 and E_2 of our ELST-KSVC on the accuracy are investigated in this subsection. The experiments were conducted on eight benchmark datasets. We changed the values of E_1 and E_2 and set other parameters to fixed values in the experiments. Figure 1 shows how the accuracy of our ELST-KSVC changes as the values of E_1 and E_2 change. We observe that the values of E_1 and E_2 significantly affect the accuracy of ELST-KSVC.

V. CONCLUSION

In this paper, to overcome the disadvantages of LST-KSVC, we have proposed a novel algorithm named ELST-KSVC. First, our ELST-KSVC employs energy parameters to reduce the effect of noise and outliers, and the regularization term to implement the SRM. Second, ELST-KSVC reconstructs the dual optimization problems to avoid inverse matrices, and the SMO is used to improve the training speed for large-scale problems. Finally, ELST-KSVC directly uses the kernel tricks for nonlinear cases. The experimental results present that ELST-KSVC outperforms other multiclass classifiers.

REFERENCES

[1] J. A. Nasiri and N. M. Charkari, "Least squares twin multi-class classification support vector machine," *Pattern Recognition*, vol. 48, no. 3, pp. 984–992, 2015.

[2] Y. Xu and R. Guo, "A twin multi-class classification support vector machine," *Computation*, vol. 5, no. 4, pp. 580–588, 2013.
[3] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[4] Jayadeva and R. Khemchandani, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905–910, 2007.
[5] W. L. Al-Yaseen, "Improving intrusion detection system by developing feature selection model based on firefly algorithm and support vector machine," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 534–540, 2019.
[6] I. K. Izzah and A. S. Girsang, "Modified TF-Assoc term weighting method for text classification on news dataset from twitter," *IAENG International Journal of Computer Science*, vol. 48, no. 1, pp. 142–151, 2021.
[7] Y. Li and H. Sun, "Domain adaptive twin support vector machine learning using privileged information," *Neurocomputing*, vol. 469, pp. 13–27, 2022.
[8] Z. Liang and L. Zhang, "Intuitionistic fuzzy twin support vector machines with the insensitive pinball loss," *Applied Soft Computing*, vol. 115, p. 108231, 2022.
[9] Y. An and H. Xue, "Indefinite twin support vector machine with DC functions programming," *Pattern Recognition*, vol. 121, p. 108195, 2022.
[10] S. Sharma and R. Rastogi, "Large-scale twin parametric support vector machine using pinball loss function," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 987–1003, 2021.
[11] Z. Che and B. Liu, "Twin support vector machines with privileged information," *Information Sciences*, vol. 573, pp. 141–153, 2021.
[12] X. H. Zheng and L. Zhang, "CTSVM: A robust twin support vector machine with correntropy-induced loss function for binary classification problems," *Information Sciences*, vol. 559, pp. 22–45, 2021.
[13] C. Yuan and L. M. Yang, "Capped $L_{2,p}$-norm metric based robust least squares twin support vector machine for pattern classification," *Neural Networks*, vol. 142, pp. 457–478, 2021.
[14] ——, "Correntropy-based metric for robust twin support vector machine," *Information Sciences*, vol. 545, pp. 82–101, 2021.
[15] M. Tanveer and A. Tiwari, "Large-scale pinball twin support vector machines," *Machine Learning*, early access.
[16] B. Richhariya and M. Tanveer, "An efficient angle-based universum least squares twin support vector machine for classification," *ACM Transactions on Internet Technology*, vol. 21, no. 3, p. 59, 2021.
[17] ——, "A fuzzy universum least squares twin support vector machine (fulstsvm)," *Neural Computing & Applications*, early access.
[18] S. Rezvani and X. Z. Wang, "Class imbalance learning using fuzzy art and intuitionistic fuzzy twin support vector machines," *Information Sciences*, vol. 578, pp. 659–682, 2021.
[19] J. Wang and F. Zhang, "Multi-class asd classification via label distribution learning with class-shared and class-specific decomposition," *Medical Image Analysis*, vol. 75, p. 102294, 2022.
[20] G. Zhao and X. Luo, "Study on principal component analysis and multi-class discrimination against the combination to identify easy-confused fur," *Journal of Industrial Textiles*, vol. 50, no. 6, pp. 794–811, 2021.
[21] G. Piir and S. Sild, "Binary and multi-class classification for androgen receptor agonists, antagonists and binders," *Chemosphere*, vol. 262, p. 128313, 2021.
[22] J. Li and G. Li, "Transformer fault diagnosis based on multi-class adaboost algorithm," *IEEE Access*, vol. 10, pp. 1522–1532, 2022.
[23] C. Kunselman and S. Sheikh, "Microstructure classification in the unsupervised context," *Acta Materialia*, vol. 223, p. 117434, 2022.
[24] M. Gour and S. Jain, "Uncertainty-aware convolutional neural network for covid-19 x-ray images classification," *Computers in Biology and Medicine*, vol. 140, p. 105047, 2022.
[25] D. J. Ho and D. V. K. Yarlagadda, "Deep multi-magnification networks for multi-class breast cancer image segmentation," *Computerized Medical Imaging and Graphics*, vol. 88, p. 101866, 2021.
[26] Y. Du and D. Han, "Raman spectroscopy-based adversarial network combined with svm for detection of foodborne pathogenic bacteria," *Talanta*, vol. 237, p. 122901, 2022.
[27] J. Chen and W. Yi, "FB-CGANet: filter bank channel group attention network for multi-class motor imagery classification," *Journal of Neural Engineering*, vol. 19, no. 1, p. 016011, 2022.
[28] B. Gao and H. Zhou, "Learning to discover multi-class attentional regions for multi-label image recognitionn," *IEEE Transactions on Image Processing*, vol. 30, pp. 5920–5932, 2021.
[29] R. Kumar and M. Gupta, "Intelligent audio signal processing for detecting rainforest species using deep learning," *Intelligent Automation and Soft Computing*, vol. 31, no. 2, pp. 693–706, 2022.
[30] R. V. V. Krishna and S. S. Kumar, "Color image segmentation using soft rough fuzzy-c-means and local binary pattern," *Intelligent Automation and Soft Computing*, vol. 26, no. 2, pp. 281–290, 2020.
[31] S. Ding and J. Zhang, "Survey on multi class twin support vector machines," *Ruan Jian Xue Bao/Journal of Sofware*, vol. 29, no. 1, pp. 89–108, 2018.
[32] J. Xie and K. Hone, "Extending twin support vector machine classifier for multi-category classification problems," *Intelligent Data Analysis*, vol. 17, no. 4, pp. 649–664, 2013.
[33] C. Angulo and X. Parra, "K-SVCR. a support vector machine for multi-class classification," *Neurocomputing*, vol. 55, no. 1, pp. 57–77, 2013.
[34] Q. Ai and A. Wang, "An improved Twin-KSVC with its applications," *Neural Computing and Applications*, vol. 31, no. 10, pp. 6615–6624, 2019.
[35] Y. Xu, "K-nearest neighbor-based weighted multi-class twin support vector machine," *Neurocomputing*, vol. 205, pp. 430–438, 2016.
[36] H. Wang and S. Lu, "Ramp loss for twin multi-class support vector classification," *International Journal of Systems Science*, vol. 51, no. 8, pp. 1448–1463, 2020.
[37] M. A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7535–7543, 2009.
[38] M. Tanveer and A. Sharma, "Least squares knn-based weighted multiclass twin svm," *Neurocomputing*, vol. 459, pp. 454–464, 2021.
[39] R. Fletcher, "Practical methods of optimization," *SIAM Review*, vol. 26, no. 10, pp. 143–144, 1984.
[40] M. D. de Lima and N. L. Costa, "Improvements on least squares twin multi-class classification support vector machine," *Neurocomputing*, vol. 313, pp. 96–205, 2018.
[41] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.