ARTICLE IN PRESS

Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

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

Least squares KNN-based weighted multiclass twin SVM

M. Tanveer [a,*], A. Sharma [a], P.N. Suganthan [b,*]

[a] Discipline of Mathematics, Indian Institute of Technology Indore, Simrol, Indore, 453552, India
[b] School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 6 June 2019
Revised 21 February 2020
Accepted 24 February 2020
Available online xxxx
Communicated by Swagatam Das

Keywords:
Twin-KSVC
Weight matrix
Imbalance data

ABSTRACT

K-nearest neighbor (KNN) based weighted multi-class twin support vector machines (KWMTSVM) is a novel multi-class classification method. In this paper, we propose a novel least squares version of KWMTSVM called LS-KWMTSVM by replacing the inequality constraints with equality constraints and minimized the slack variables using squares of 2-norm instead of conventional 1-norm. This simple modification leads to a very fast algorithm with much better results. The modified primal problems in the proposed LS-KWMTSVM solves only two systems of linear equations whereas two quadratic programming problems (QPPs) need to solve in KWMTSVM. The proposed LS-KWMTSVM, same as KWMTSVM, employed the weight matrix in the objective function to exploit the local information of the training samples. To exploit the inter class information, we use weight vectors in the constraints of the proposed LS-KWMTSVM. If any component of vectors is zero then the corresponding constraint is redundant and thus we can avoid it. Elimination of redundant constraints and solving a system of linear equations instead of QPPs makes the proposed LS-KWMTSVM more robust and faster than KWMTSVM. The proposed LS-KWMTSVM, commensurate as the KWMTSVM, all the training data points into a “1-versus-1-versus-rest” structure, and thus our LS-KWMTSVM generate ternary output $\{-1, 0, +1\}$ which helps to deal with imbalance datasets. Numerical experiments on several UCI and KEEL imbalance datasets(with high imbalance ratio) clearly indicate that the proposed LS-KWMTSVM has better classification accuracy compared with other baseline methods but with remarkably less computational time.

1. Introduction

Support vector machine (SVM) is one of the most powerful kernel-based tools of machine learning used for classification and regression problems [4,5]. It has been applied in several real-world problems including electroencephalogram signal classification [27], remote sensing [24], diagnosis of Alzheimer’s disease [36], face detection [20]. SVM obtains a unique (global) optimal separation by solving two quadratic programming problem (QPP) which maximizing the margin among the two classes. The final optimal hyperplane is selected to be the “middle one” between the supporting hyperplanes. The main challenge of SVM formulation is its high computational complexity of order $\mathcal{O}(\ell^3)$, where ℓ is the total number of training data points in classification problem. This hindrance of SVM algorithms restricts it to apply on large datasets. To impoverish the high computational complexity problem, Jayadeva et al. [14] proposed a novel twin support vector machine (TSVM), which is similar in spirit to generalized eigenvalue proximal support vector machines (GEPSVM) [21] that obtains two non-parallel hyperplanes by solving two smaller size QPPs. The formulation of SVM requires all data points, however, data points in TSVM are distributed in such a way that one class gives the constraints to the other class and vice versa. The idea of dividing a large size QPP into two smaller size QPPs in TSVM makes the algorithm approximately four times faster than standard SVM. Due to the low computational complexity, TSVM becomes one of the most popular classification techniques for various applications. In the last one decade, many variants and extensions of TSVM [15,16,25,26,29–32,39,42,46] have been proposed. Kumar and Gopal [19] proposed least squares twin support vector machines (LSTSVM) as a way to reduce QPPs in TSVM with a linear system of equations by using a squared loss function instead of hinge loss function. Tanveer et al. [34] proposed a novel robust energy-based LSTSVM (RELS-TSVM) for classification which is the top-ranked classifier according to the recent comprehensive evaluation [33]. Datta et al. [7] proposed a novel classifier for imbalanced data classification with equal or unequal misclassification costs. Recently, Datta et al. [8] proposed a robust multiobjective

* Corresponding authors.
E-mail addresses: mtanveer@iiti.ac.in (M. Tanveer), msc1603141001@iiti.ac.in (A. Sharma), epnsugan@ntu.edu.sg (P.N. Suganthan).

https://doi.org/10.1016/j.neucom.2020.02.132
0925-2312/

Please cite this article as: M. Tanveer, A. Sharma and P. N. Suganthan, Least squares KNN-based weighted multiclass twin SVM, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.132

classifier which handle the class imbalance with pareto optimality. Recently, Tanveer et al. [37] proposed a novel general TSVM with pinball loss (Pin-GTSVM) by introducing pinball loss to the original TSVM. Pin-GTSVM is noise insensitive and better generalization performance to that of TSVM. To retain the sparsity in Pin-GTSVM, Tanveer et al. [38,35] proposed a novel sparse pinball twin SVM (SPTWSVM) which uses ϵ-insensitive zone pinball loss function. SPTWSVM is sparse, insensitive to outliers, better generalization performance and suitable for re-sampling. Recently, an efficient algorithm termed as reduced universum twin SVM (RUTSVM-CIL) [28] is proposed to solve the class imbalance learning. RUTSVM-CIL [28] is computationally efficient for large scale class imbalanced datasets.

The variants of SVM and TSVM discussed above are suitable for binary classification problems. However, multiclass classification problems are more prevalent in real life situations such as speech recognition, text classification and fault diagnosis. The classification problem of multiclass is usually solved by a decomposition and reconstruction procedure, when binary classification machines are intended. This decomposition-reconstruction strategy includes two approaches "one-versus-one" [3] and "one-versus-rest" [17]. For k-class classification problem, the first approach is to construct $\frac{k(k-1)}{2}$ binary SVM classifiers, each of which involves only two kinds of data points. Since the remaining data points are omitted in the training process, and thus we receive unfavourable results. While the second approach (one-versus-rest) constructs k binary classifiers, each classifier involves with all of the data points. "one-versus-all" approach leads to class disproportion problem and may produce dreadful results. Angulo et al. [2] proposed an effectual multiclass support vector machine classification algorithm termed as K-SVCR, which employs "one-versus-one-versus-rest" with ternary outputs $\{-1, 0, 1\}$, during the decomposition phase K-SVCR uses a mixed classification and regression support vector machines formulation. The K-SVCR constructs $\frac{k(k-1)}{2}$ binary SVM classifiers, each classifier is implemented with all data points, training avoids the risk of information loss and class distortion problem. By consolidating the formalistic advantage of K-SVCR and less computational complexity of TSVM, Xu et al. [44] extended K-SVCR for multi-class TSVM termed as Twin-KSVC for k-class classification problem. Twin-KSVC choose two focused kinds of samples from k classes to construct two non-parallel hyperplanes and the remaining samples are mapped into a region between the two non-parallel hyperplanes. The optimal separating hyperplanes are computed by resolving two smaller QPPs instead of solving a large QPP in the original K-SVCR. Following the works of Twin-KSVC and LST-KSVC, Nasiri et al. [23] recently proposed least squares twin multi-class classification support vector machine (LST-KSVC) which is extremely simple and fast algorithm. In Twin-KSVC and LST-KSVC algorithms, data points contribute the same weight for the construction of the hyperplanes, so that local information of training samples is omitted, and inter-class information is also not exploited. However, data points have different influences on the hyperplanes. To exploit the intra-class as well as inter-class information in multiclass classification problem, Xu et al. [43] proposed an algorithm called KNN-based weighted multiclass twin support vector machines (KWMTSVM). In KWMTSVM, K-nearest neighbor graph [44] is used to exploit the local information of the training samples and weight matrix D_1, D_2 are employed in the objective function, meanwhile weight vectors $F_{v_i} (i = 1, 2), H_v$ are introduced in constraints. If any component of F_{v_i}, H_v is zero then it entail that the consonant constraint is redundant.

Motivated by the works of [19,23,43], we propose a novel least squares KNN-based weighted multiclass twin support vector machines (LS-KWMTSVM). The proposed LS-KWMTSVM is endowed with the following attractive advantages:

- Unlike Twin-KSVC and KWMTSVM, the proposed LS-KWMTSVM solves two systems of linear equations which leads to extremely simple and fast algorithm. As a result, the proposed LS-KWMTSVM does not need any external optimizer.
- The Sherman–Morrison–Woodbury (SMW) formulation is employed to reduce the complexity of the nonlinear LS-KWMTSVM.
- The proposed LS-KWMTSVM evaluates all the training data points in a "1-versus-1-versus-rest" structure, so the proposed LS-KWMTSVM inaugurate ternary output $\{-1, 0, 1\}$ which helps to deal with imbalance datasets.
- The proposed LS-KWMTSVM, same as KWMTSVM, uses KNN graph approach to utilize the intra-class and inter-class information, different weight matrices are given to data points for the same class.
- Extensive numerical experimental are performed on UCI [22] and KEEL benchmark imbalance datasets [1], and their results are compared with three algorithms (Twin-KSVC [44], LST-KSVC [23] and KWMTSVM [43]). The comparative results clearly show the effectiveness and feasibility of the proposed LS-KWMTSVM for solving imbalance classification problems.

The rest of paper is organized as follows. A brief introduction of Twin-KSVC and KWMTSVM are presented in Section 2. Section 3 describes the detail of the proposed LS-KWMTSVM algorithm. Section 4 discusses the computational complexity and in-depth analysis of proposed algorithm. In Section 5, we compare the proposed LS-KWMTSVM to Twin-KSVC and KWMTSVM. Numerical experiments on twelve UCI and KEEL benchmark datasets are accompanied to evaluate the effectiveness of the proposed LS-KWMTSVM algorithm in Section 6. Conclusions are presented in Section 7.

2. Related works

In this section, a brief framework of Twin-KSVC and KWMTSVM are presented. For more details, the interested readers are referred to [43,44].

2.1. Twin multi-class classification support vector machines (Twin-KSVC)

A new variant of multi-class algorithm, called K-SVCR, was affianced in [2] for the k-class classification problem, which originated better results as it appraise all the data points into the "1-versus-1-versus-rest" structure with outputs $\{-1, 0, 1\}$. By integrating both the structural advantage of K-SVCR and the speed's advantage of TSVM, Xu et al. [44] proposed a novel multi-class classification algorithm, called Twin-KSVC. The two variety of data points extracted from k-classes are scrutinize as the focused classes, and Twin-KSVC obtains two non-parallel hyperplanes corresponding to two focused classes. The remain samples are epitomize into a region between the above optimal hyperplanes. Let the matrix $A \in \mathbb{R}^{\ell_1 \times n}$ represents the data samples of class +1, $B \in \mathbb{R}^{\ell_2 \times n}$ is corresponding to the class -1 and $C \in \mathbb{R}^{\ell_3 \times n}$ represents the remaining data points which are labeled 0. The nonlinear Twin-KSVC tries to find two nonlinear kernel generated surfaces in the input space defined as:

$$K(x^T, D_*^T)u_+ + b_+ = 0 \text{ and } K(x^T, D_*^T)u_- + b_- = 0, \qquad (1)$$

where $D_* = [A; B; C]; u_+, u_- \in \mathbb{R}^n$ and K is an arbitrary kernel function. They can be obtained by resolving the following pair of QPPs:

Please cite this article as: M. Tanveer, A. Sharma and P. N. Suganthan, Least squares KNN-based weighted multiclass twin SVM, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.132

$$\min_{u_+,b_+,\xi_1,\eta_1} \frac{1}{2}\|K(A,D_*^T)u_+ + e_1b_+\|^2 + c_1e_2^T\xi_1 + c_2e_3^T\eta_1$$
$$s.t. \quad -(K(B,D_*^T)u_+ + e_2b_+) + \xi_1 \geqslant e_2, -(K(C,D_*^T)u_+ + e_3b_+) + \eta_1 \geqslant e_3(1-\epsilon), \quad \xi_1 \geq 0, \eta_1 \geq 0 \tag{2}$$

and

$$\min_{u_-,b_-,\xi_2,\eta_2} \frac{1}{2}\|K(B,D_*^T)u_- + e_2b_-\|^2 + c_3e_1^T\xi_2 + c_4e_3^T\eta_2$$
$$s.t. \quad (K(A,D_*^T)u_- + e_1b_-) + \xi_2 \geqslant e_1, \tag{3}$$
$$(K(C,D_*^T)u_- + e_3b_-) + \eta_2 \geqslant e_3(1-\epsilon), \xi_2 \geq 0, \eta_2 \geq 0,$$

where $c_i(i=1,2,3,4)$ are positive real penalty parameters, $e_i(i=1,2,3)$ are standard unit vectors of appropriate dimensions and ξ_i, $\eta_i(i=1,2)$ are slack variables.

By introducing the Lagrangian multipliers $\alpha \geq 0$ and $\beta \geq 0$, and using Karush–Kuhn–Tucker (K.K.T.) [18] necessary and sufficient optimality conditions, the dual formulation of (2) and (3) are as follows:

$$\max_{\rho} e_4^T\rho - \frac{1}{2}\rho^T N_1 (R^T R)^{-1} N_1^T \rho$$
$$s.t. \quad 0 \leqslant \rho \leqslant K_1 \tag{4}$$

and

$$\max_{\zeta} e_5^T\zeta - \frac{1}{2}\zeta^T N_2 (S^T S)^{-1} N_2^T \zeta$$
$$s.t. \quad 0 \leqslant \zeta \leqslant K_2, \tag{5}$$

where $R = [K(A,D_*^T)\ e_1], S = [K(B,D_*^T)\ e_2]$ and $M = [K(C,D_*^T)\ e_3]$. $N_1 = [S; M]; N_2 = [R; M]; K_1 = [c_1e_2; c_2e_3]; K_2 = [c_3e_1; c_4e_3]; \quad \rho = [\alpha; \beta]$; $\zeta = [\gamma; \delta]; e_4 = [e_2; e_3(1-\epsilon)]$ and $e_5 = [e_1; e_3(1-\epsilon)]$. The kernel generated surfaces in Eq. (1) can be obtained from the solution of QPPs (4) and (5) as given below:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -[R^T R + \delta I]^{-1}[S^T\alpha + M^T\beta]$$

and

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = [S^T S + \delta I]^{-1}[R^T\gamma + M^T\sigma],$$

where $\delta I(\delta > 0)$ is a regularization term [40] used to avoid the ill-conditioning of matrices R^TR and S^TS. In Twin-KSVC class label of new testing point x, determines by the following decision function [44]

$$f(x) = \begin{cases} -1, & K(x^T, D_*^T)u_- + e_2b_- < 1-\epsilon \\ 1, & K(x^T, D_*^T)u_+ + e_1b_+ > -1+\epsilon \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

In this way Twin-KSVC constructs $k(k-1)/2$ classifiers for k-classes. One of the focused classes acquire a vote for x, according to the condition satisfied by it. Then x is assigned to the class, which gets highest votes.

2.2. KNN-based weighted multi-class twin support vector machines (KWMTSVM)

The data points in Twin-KSVC algorithm contributes the same weights when constructing the hyperplanes. Due to this, the local information of data points is omitted and inter-class information is also not exploited. However, they have contrasting impact on the hyperplanes. In KNN-based weighted multi-class twin support vector machines (KWMTSVM) algorithm [43], K-nearest neighbor method [6] is used to obtain the information of intra-class and inter-class, For each sample x_k in class +1, define two sets: $Neb_s(x_k)$ and $Neb_d(x_k)$, where $Neb_s(x_k)$ contains its K neighbors in class v1, while $Neb_d(x_k)$ contains its K neighbors in class -1.

$$Neb_s(x_k) = \{x_k^j | \text{ if } x_k^j \text{ and } x_k \text{ belong to the same class}, 0 \leqslant j \leqslant m_1\}, \tag{7}$$

$$Neb_d(x_k) = \{x_k^j | \text{ if } x_k^j \text{ and } x_k \text{ belong to the different class}, 0 \leqslant j \leqslant m_2\}, \tag{8}$$

where $Neb_s(x_k)$ represents a set of m_1-nearest neighbors of x_k in class +1, and $Neb_d(x_k)$ represents a set of m_2-nearest neighbors x_k in class -1. For two adjacent matrices of class +1, define M_s and M_d as follows [43]:

$$M_{s,ij} = \begin{cases} 1, & \text{if } x_j \in Neb_s(x_i) \text{ or } x_i \in Neb_s(x_j) \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

and

$$M_{d,ij} = \begin{cases} 1, & \text{if } x_j \in Neb_d(x_i) \text{ or } x_i \in Neb_d(x_j) \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

When $M_{s,ij} = 1$ or $M_{d,ij} = 1$ is an undirectional edge between two points. To reduce the data points, redefine the weighed matrix $M_{d,ij}$ as follows:

$$f_j = \begin{cases} 1, & \text{if } \exists i, M_{d,ij} \neq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The nonlinear KWMTSVM seeks for two kernel generated surfaces given in Eq. (1) and can be obtained by solving the following pair of QPPs [43]:

$$\min_{u_+,b_+,\xi_1,\eta_1} \frac{1}{2}\|D_1(K(A,D_*^T)u_+ + e_1b_+)\|^2 + c_1e_2^T\xi_1 + c_2e_3^T\eta_1$$
$$s.t. \quad -F_1(K(B,D_*^T)u_+ + e_2b_+) + \xi_1 \geqslant F_{v_1}, -H_1(K(C,D_*^T)u_+ + e_3b_+) + \eta_1 \geqslant (1-\epsilon)H_v, \ \xi_1 \geq 0, \eta_1 \geq 0 \tag{12}$$

and

$$\min_{u_-,b_-,\xi_2,\eta_2} \frac{1}{2}\|(D_2K(B,D_*^T)u_- + e_2b_-)\|^2 + c_3e_1^T\xi_2 + c_4e_3^T\eta_2$$
$$s.t. \quad F_2(K(A,D_*^T)u_- + e_1b_-) + \xi_2 \geqslant F_{v_2}, \tag{13}$$
$$H_2(K(C,D_*^T)u_- + e_3b_-) + \eta_2 \geqslant (1-\epsilon)H_v,$$
$$\xi_2 \geq 0, \eta_2 \geq 0,$$

where $F_1 = diag(f_1, f_2, \ldots, f_{\ell_2}); H_1 = diag(h_1, h_2, \ldots, h_{\ell_3}); F_2 = diag(f_1, f_2, \ldots, f_{\ell_1}); H_2 = diag(h_1, h_2, \ldots, h_{\ell_3}); D_1 = diag(d_1, d_2, \ldots, d_{\ell_1})$ and $d_j = \sum_{i=1}^{\ell_1} M_{s,ij}$. F_{v_1} denotes the vector of diagonal elements of F_1, similarly F_{v_2} and H_v are defined. By introducing the Lagrange multipliers α, β, γ and σ, the dual formulation of Eqs. (12) and (13) are as follows:

$$\max_{\rho} e_4^T\rho - \frac{1}{2}\rho^T M_1^T (R^T D_1 R)^{-1} M_1 \rho$$
$$s.t. \quad 0 \leqslant \rho \leqslant K_1, \tag{14}$$

where $M_1 = [S^T F_1\ M^T H_1]$ and $e_4 = [F_1^T e_2;\ (1-\epsilon)H_1^T e_3]$.

$$\max_{\zeta} e_5^T\zeta - \frac{1}{2}\zeta^T M_2^T (S^T D_2 S)^{-1} M_2 \zeta$$
$$s.t. \quad 0 \leqslant \zeta \leqslant K_2, \tag{15}$$

where $M_2 = [R^T F_2\ M^T H_2]$ and $e_5 = [F_2^T e_1;\ (1-\epsilon)H_2^T e_3]$. Once QPPs in Eqs. (14) and (15) are resolved, we can obtain:

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -(R^T D_1 R + \delta I)^{-1}(S^T F_1 \alpha + M^T H_1 \beta)$$

and

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = (S^T D_2 S + \delta I)^{-1} (R^T F_2 \gamma + M^T H_2 \sigma),$$

where $\delta I(\delta > 0)$ is a regularization term used to avoid the ill-conditioning of matrices $R^T D_1 R$ and $S^T D_2 S$. The decision function of KWMTSVM is the same as the decision function of Twin-KSVC.

3. Proposed LS-KWMTSVM

Motivated by the works of [19,23,43], we propose a novel least squares KNN-based weighted multi-class twin support vector machines (LS-KWMTSVM). The proposed LS-KWMTSVM replaces the inequality constraints with equality constraints and utilizes the squared loss function instead of hinge loss function as in Twin-KSVC and KWMTSVM. Similar to KWMTSVM, the proposed LS-KWMTSVM also uses K-nearest neighbor graph approach to exploit the intra-class and inter-class information, different weight matrix M_s are given to training data points for the same class. The proposed LS-KWMTSVM solves two system of linear equations instead of solving QPPs in Twin-KSVC and KWMTSVM, which makes the proposed LS-KWMTSVM extremely simple and fast. The proposed algorithm does not require any special optimizer. Similar to K-SVCR, Twin-KSVC and KWMTSVM, the proposed LS-KWMTSVM appraise all the data samples into a "1-versus-1-versus-rest" structure, so the proposed algorithm inaugurate ternary outputs $\{-1, 0, +1\}$ which help to deal with imbalance datasets.

3.1. Linear LS-KWMTSVM

Let matrix $A \in \mathbb{R}^{\ell_1 \times n}$ represents the data points of class +1, $B \in \mathbb{R}^{\ell_2 \times n}$ represents the data points of class -1 and $C \in \mathbb{R}^{\ell_3 \times n}$ represents the remaining data points which are labeled 0. To classify the two focused classes, the linear LS-KWMTSVM tries to find two non-parallel hyperplanes defined as follows:

$$f^+(x) = w_+^T x + b_+ = 0 \text{ and } f^-(x) = w_-^T x + b_- = 0, \tag{16}$$

where $w_+, w_- \in \mathbb{R}^n$ and $b_+, b_- \in \mathbb{R}$. The linear LS-KWMTSVM assimilate of the following pair of optimi pzionatroblems:

$$\begin{aligned} \min_{w_+, b_+ \xi_1, \eta_1} \ & \tfrac{1}{2}\|D_1(Aw_+ + e_1 b_+)\|^2 + \tfrac{c_1}{2}\|\xi_1\|^2 + \tfrac{c_2}{2}\|\eta_1\|^2 \\ s.t. \ & -F_1(Bw_+ + e_2 b_+) + \xi_1 = F_{v_1}, \\ & -H_1(Cw_+ + e_3 b_+) + \eta_1 = (1-\epsilon)H_v \end{aligned} \tag{17}$$

and

$$\begin{aligned} \min_{w_-, b_- \xi_2, \eta_2} \ & \tfrac{1}{2}\|D_2(Bw_- + e_2 b_-)\|^2 + \tfrac{c_3}{2}\|\xi_2\|^2 + \tfrac{c_4}{2}\|\eta_2\|^2 \\ s.t. \ & F_2(Aw_- + e_1 b_-) + \xi_2 = F_{v_2}, H_2(Cw_- + e_3 b_-) + \eta_2 = (1-\epsilon)H_v, \end{aligned} \tag{18}$$

where $D_1, D_2, H_1, H_2, F_1, F_2, F_{v_2}, F_{v_1}$ and H_v are same as defined in KWMTSVM. ξ_1, η_1, ξ_2 and η_2 are the slack variables. By substituting the values of ξ_1 and η_1 into the QPP (17), we obtain:

$$\begin{aligned} in_{w_+, b_+} \frac{1}{2}\|D_1(Aw_+ + e_1 b_+)\|^2 + \frac{c_1}{2}\|F_{v_1} + F_1(Bw_+ + e_2 b_+)\|^2 \\ + \frac{c_2}{2}\|(1-\epsilon)H_v + H_1(Cw_+ + e_3 b_+)\|^2. \end{aligned} \tag{19}$$

Using the K.K.T. optimality conditions [18], we obtain:

$$\begin{aligned} & A^T D_1^T D_1(Aw_+ + e_1 b_+) + c_1 B^T F_1^T (F_{v_1} + F_1(Bw_+ + e_2 b_+)) \\ & + c_2 C^T H_1^T ((1-\epsilon)H_v + H_1(Cw_+ + e_3 b_+)) = 0 \end{aligned} \tag{20}$$

and

$$\begin{aligned} & e_1^T D_1^T D_1(Aw_+ + e_1 b_+) + c_1 e_2^T F_1^T (F_{v_1} + F_1(Bw_+ + e_2 b_+)) \\ & + c_2 e_3^T H_1^T ((1-\epsilon)H_v + H_1(Cw_+ + e_3 b_+)) = 0. \end{aligned} \tag{21}$$

Rearranging Eqs. (20) and (21) in matrix form and resolving for w_+ and b_+ gives

$$\begin{aligned} & \begin{bmatrix} A^T \\ e_1^T \end{bmatrix} D_1^T D_1 [A e_1] \begin{bmatrix} w_+ \\ b_+ \end{bmatrix} + c_1 \begin{bmatrix} B^T \\ e_2^T \end{bmatrix} F_1^T \left(F_{v_1} + F_1 [B e_2] \begin{bmatrix} w_+ \\ b_+ \end{bmatrix} \right) \\ & + c_2 \begin{bmatrix} C^T \\ e_3^T \end{bmatrix} H_1^T \left(H_v(1-\epsilon) + H_1 [C e_3] \begin{bmatrix} w_+ \\ b_+ \end{bmatrix} \right) = 0 \end{aligned} \tag{22}$$

$$H^T D_1^T D_1 H z_+ + c_1 G^T F_1^T [F_{v_1} + F_1 G z_+] + c_2 T^T H_1^T [H_v(1-\epsilon) + H_1 T z_+] = 0,$$

where $H = [A\, e_1], G = [B\, e_2], T = [C\, e_3]$ and $z_+ = \begin{bmatrix} w_+ \\ b_+ \end{bmatrix}$.

$$\begin{aligned} \begin{bmatrix} w_+ \\ b+ \end{bmatrix} = & -(H^T D_1^T D_1 H + c_1 G^T F_1^T F_1 G + c_2 T^T H_1^T H_1 T)^{-1} (c_1 G^T F_1^T F_{v_1} \\ & + c_2 T^T H_1^T H_v(1-\epsilon)). \end{aligned} \tag{23}$$

Similarly, the solution of (18) is given by:

$$\begin{aligned} \begin{bmatrix} w_- \\ b_- \end{bmatrix} = & (G^T D_2^T D_2 G + c_3 H^T F_2^T F_2 H + c_4 T^T H_2^T H_2 T)^{-1} (c_3 H^T F_2^T F_{v_2} \\ & + c_4 T^T H_2^T H_v(1-\epsilon)). \end{aligned} \tag{24}$$

The optimal hyperplanes of Eq. (16) are obtained. The proposed LS-KWMTSVM gives the solution of classification problem with just inverse of two matrices of smaller sizes rather than solving two QPPs in KWMTSVM.

3.2. Nonlinear LS-KWMTSVM

In order to enhance the proposed linear LS-KWMTSVM to nonlinear LS-KWMTSVM, we scrutinize the following kernel generated surfaces instead of hyperplanes:

$$K(x^T, D_*^T)u_+ + b_+ = 0 \text{ and } K(x^T, D_*^T)u_- + b_- = 0, \tag{25}$$

where $D_* = [A; B; C]; u_+, u_- \in \mathbb{R}^n$ and K is an appropriate kernel function.

The nonlinear LS-KWMTSVM assimilate of the following pair of optimization problems:

$$\begin{aligned} \min_{u_+, b_+, \xi_1, \eta_1} \ & \tfrac{1}{2}\|D_1(K(A, D_*^T)u_+ + e_1 b_+)\|^2 + \tfrac{c_1}{2}\|\xi_1\|^2 + \tfrac{c_2}{2}\|\eta_1\|^2 \\ s.t. \ & -F_1(K(B, D_*^T)u_+ + e_2 b_+) + \xi_1 = F_{v_1}, -H_1(K(C, D_*^T)u_+ \\ & + e_3 b_+) + \eta_1 = (1-\epsilon)H_v \end{aligned} \tag{26}$$

and

$$\begin{aligned} \min_{u_-, b_-, \xi_2, \eta_2} \ & \tfrac{1}{2}\|D_2(K(B, D_*^T)u_- + e_2 b_-)\|^2 + \tfrac{c_3}{2}\|\xi_2\|^2 + \tfrac{c_4}{2}\|\eta_2\|^2 \\ s.t. \ & F_2(K(A, D_*^T)u_- + e_1 b_-) + \xi_2 = F_{v_2}, H_2(K(C, D_*^T)u_- + e_3 b_-) \\ & + \eta_2 = (1-\epsilon)H_v. \end{aligned} \tag{27}$$

Analogous to linear case, we obtain the solutions of Eqs. (26) and (27) as follows:

$$\begin{aligned} \begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = & -(R^T D_1^T D_1 R + c_1 S^T F_1^T F_1 S + c_2 M^T H_1^T H_1 M)^{-1} (c_1 S^T F_1^T F_{v_1} \\ & + c_2 M^T H_1^T H_v(1-\epsilon)) \end{aligned} \tag{28}$$

and

$$\begin{aligned} \begin{bmatrix} u_- \\ b_- \end{bmatrix} = & (S^T D_2^T D_2 S + c_3 R^T F_2^T F_2 R + c_4 M^T H_2^T H_2 M)^{-1} (c_3 R^T F_2^T F_{v_2} \\ & + c_4 M^T H_2^T H_v(1-\epsilon)), \end{aligned} \tag{29}$$

Please cite this article as: M. Tanveer, A. Sharma and P. N. Suganthan, Least squares KNN-based weighted multiclass twin SVM, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.132

where $R = [K(A, D_*^T)\ e_1]; S = [K(B, D_*^T)\ e_2]; M = [K(C, D_*^T)\ e_3]$ and ϵ is a real positive parameter. We notice that for the solution of nonlinear case, we require inverse of two matrices of size $(\ell+1)\times(\ell+1)$. To depreciate the computational cost, we use *Sherman–Morrison–Wood bury* (SMW) [11] formula to replicate Eqs. (28) and (29)

$$\begin{bmatrix} u_+ \\ b_+ \end{bmatrix} = -\left(Z - Z(F_1S)^T\left(\frac{I}{c_1} + (F_1S)^T Z (F_1S)^T\right)^{-1} F_1SZ\right) \times \left(c_1 S^T F_1 F_{v_1} + c_2 M^T H_1^T H_v (1-\epsilon)\right), \quad (30)$$

$$\begin{bmatrix} u_- \\ b_- \end{bmatrix} = \left(F_* - F_*(F_2R)^T\left(\frac{I}{c_3} + (F_2R)^T F_* (F_2R)^T\right)^{-1} F_2RF_*\right) \times \left(c_1 R^T F_2 F_{v_2} + c_2 M^T H_2^T H_v (1-\epsilon)\right), \quad (31)$$

where $Z = (R^T D_1^T D_1 R + c_2 M^T H_1^T H_1 M)^{-1}$ and $F_* = (S^T D_2^T D_2 S + c_4 M^T H_1^T H_1 M)^{-1}$.

Again by using the SMW formula we obtain:

$$Z = \frac{1}{c_2}\left(Y - Y(F_2R)^T (c_2 I + F_2RY(F_2R)^T)^{-1} F_2RY\right),$$

$$F_* = \frac{1}{c_4}\left(Y_* - Y_*(F_1S)^T (c_4 I + F_1SY_*(F_1S)^T)^{-1} F_1SY_*\right),$$

where $Y = (M^T H_1^T H_1 M)^{-1}$ and $Y_* = (M^T H_2^T H_2 M)^{-1}$. To avoid the case when Y and Y_* are ill-conditioned, we add a regularization term δI, where $\delta > 0$ then

$$Y = \frac{1}{\delta}(I - (H_1M)^T (\delta I + M^T H_1^T H_1 M)^{-1} H_1 M). \quad (32)$$

$$Y_* = \frac{1}{\delta}(I - (H_2M)^T (\delta I + M^T H_2^T H_2 M)^{-1} H_2 M). \quad (33)$$

Advantage of SMW formula is to compute three inverse matrices Z, F_*, Y and Y_* of smaller dimensions.

The decision function of the proposed LS-KWMTSVM is same as the decision function (6) of Twin-KSVC [44].

The class label of new testing data point x is decided by majority voting. For a new data sample, if inequality $w_+^T x + e_1 b_+ > -1 + \epsilon$ is convinced, then a vote is acquire by positive (+1) class. On the contrary, if the inequality $w_-^T x + e_2 b_- < 1 - \epsilon$ is satisfied, then a vote is aggregated by negative (-1) class, and 0 vote acquire by other classes. If the above in-equations are not fulfilled, we give a vote -1 to positive and negative class, and rest class obtain zero vote. Finally, we establish $k(k-1)/2$ classifiers for k-classes and calculate the total votes acquires by each class, then the testing data sample x will be nominated to the class that aggregates the highest votes.

4. Algorithm analysis

The proposed LS-KWMTSVM algorithm seeks two non-parallel hyperplanes by solving a system of linear equations.

- We give different weights to the data points of the focused class +1 in Eq. (16) by using the KNN graph [45]. If a data sample in the focused class +1 has more KNNs, then we give more weight to it.
- Constraint Reduction: We introduce $F_1 = diag(f_1, f_2, \ldots, f_{l_2})$ where $f_i = 1$ or 0 in Eq. (16). Similarly, $H_1 = diag(h_1, h_2, \ldots, h_{l_3})$ where $h_i = 1$ or 0. If any data sample of focused class associated to the KNN of another focused class i.e., $f_i = 1$ or $h_i = 1$ otherwise the corresponding constraint is redundant.
- The proposed algorithm exploits the local information of intra-class and inter-class by using the KNN method and imbalance problem is resolved by using the "1-versus-1-versus -rest" approach.
- Computational Complexity: It is well known that computational complexity of SVM is $\mathcal{O}(\ell^3)$ [5] where ℓ is the number of data points. Computational complexity of TSVM is $\mathcal{O}(2\times(\frac{\ell}{2})^3)$ because TSVM solves two problems, each of which is roughly equal size matrices of order $(\frac{\ell}{2})$.

In a 3-class classification problem, assume that each class have approximately $\ell/3$ data points. The data points of 3rd class involved twice in the constraints of the K-SVCR, thus there are $\frac{4\ell}{3}$ constraints. Therefore, computational complexity of K-SVCR is $\mathcal{O}(\frac{4\ell}{3})^3$. In Twin-KSVC, the data points of 3rd class are used only once hence computational complexity of Twin-KSVC is $\mathcal{O}(2\times(\frac{2\ell}{3})^3)$.

If LS-KWMTSVM has no redundant constraint then LS-KWMTSVM has the identical construction as Twin-KSVC, so it have the similar computational complexity. If LS-KWMTSVM have the some redundant constraint then computational complexity of LS-KWMTSVM is less than $\mathcal{O}(\frac{16\ell^3}{27})$. In LS-KWMTSVM, KNN- graph needs $\ell^2(\log(\ell))$ steps to compute the weight matrices for each data

Table 1
Imbalance ratio (IR) of datasets [1].

Dataset	No. of samples	No. of attributes	No. of classes	Imbalance ratio (IR)
Iris	150	4	3	2
Teaching	151	5	4	2.08
Wine	178	13	3	1.5
Hayes	132	5	3	1.7
Glass	214	9	6	8.44
Lenses	24	4	3	5
Contraceptive	1473	9	3	1.89
Zoo	101	16	7	19.2
Cleveland	297	13	5	2.09
Tae	150	5	3	2.06
Seeds	210	7	3	2
Newthyroid	215	5	3	4.84
Car	1728	6	4	24.04
Balance	625	4	3	5.88
Ecoli	336	7	5	71.5
Vertebal	310	6	3	12.75
Soyabean	47	35	4	11.52
Dermatology	358	34	6	16.9

Please cite this article as: M. Tanveer, A. Sharma and P. N. Suganthan, Least squares KNN-based weighted multiclass twin SVM, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.132

Table 2
Performance comparison of proposed LS-KWMTSVM with Twin-KSVC, LST-KSVC and KWMTSVM using Gaussian kernel.

Dataset	Twin-KSVC		LST-KSVC		KWMTSVM		LS-KWMTSVM	
	Accuracy(%)($c_1 = c_3, c_2 = c_4, \mu$)	Time(s)	Accuracy(%)($c_1 = c_3, c_2 = c_4, \mu$)	Time(s)	Accuracy(%)($c_1 = c_3, c_2 = c_4, \mu$)	Time(s)	Accuracy(%)($c_1 = c_3, c_2 = c_4, \mu$)	Time(s)
Iris (150×4×3)	**88.89** (0.031,2,8)	0.31	86.66 (8,0.031,0.5)	0.08	82.22 (0.125,0.5,2,0.2)	0.33	**88.89** (8,0.031,0.25)	0.130
Teaching (151×5×3)	43.47 (0.125,0.31,8)	0.30	**60.86** (32,8,0.062)	0.08	54.34 (0.5,0.125,0.125)	0.34	58.69 (8,2,0.0625)	0.132
Wine (178×13×3)	92.45 (0.031,8,1024)	0.34	**94.33** (32,1,64)	0.09	92.45 (2,0.031,1024)	0.36	93.33 (32,1,128)	0.157
Hayes (132×5×3)	66.67 (0.031,0.5,2)	0.30	**76.19** (32,0.031,1)	0.07	66.67 (0.125,0.031,4)	0.31	**76.19** (32,0.031,1)	0.11
Glass (214×9×6)	57.97 (0.031,1,16)	0.95	60 (8,0.125,0.0625)	0.45	59.42 (2,0.031,8)	1.14	**65.21** (32,0.125,0.125)	0.67
Lense (24×4×3)	75 (0.031,0.031,16)	0.26	75 (32,0.031,8)	0.04	75 (0.031,0.031,16)	0.28	**87.5** (8,8,4)	0.07
Contraceptive (210×9×3)	**40** (32,32,1024)	0.41	32.30 (2,2,2)	0.115	35 (1,0.031,4)	0.43	**40** (2,0.031,1)	0.22
Zoo (100×16×7)	61.29 (0.031,0.031,2)	0.66	93.54 (0.031,0.5,2)	0.19	61.29 (32,32,1024)	0.71	**96.77** (1,0.125,0.7)	0.37
Cleveland (297×13×5)	52.87 (0.125,0.125,256)	1.11	50.57 (32,0.031,2)	0.46	50.57 (0.5,0.031,256)	1.22	**55.17** (8,0.031,8)	0.72
Tae (150×5×3)	44.44 (0.5,0.5,0.5)	0.32	48.89 (32,0.031,0.004)	0.08	53.33 (0.125,8,0.125)	0.35	**57.77** (8,0.125,0.0078)	0.14
Seeds (210×7×3)	93.33 (1,0.5,0.5)	0.38	93.33 (32,0.125,1)	0.09	**96.66** (0.5,0.125,1)	0.42	93.33 (32,0.031,4)	0.165
Newthyroid (215×5×3)	**98.46** (1,0.5,4)	0.36	95.384 (0.125,8,0.125)	0.10	95.3 (0.125,8,0.125)	0.44	96.38 (2,0.031,0.125)	0.16
Car (1728×6×4)	73.37 (0.031,0.031,2)	6.57	**94.13** (0.031,0.5,2)	0.37	92.15 (32,32,1024)	7.71	**94.13** (1,0.125,0.7)	0.87
Balance (625×4×3)	92.85 (32,32,1024)	2.12	96.55 (2,2,2)	0.17	96.55 (1,0.031,4)	3.02	**97.17** (2,0.031,1)	0.37
Ecoli (366×7×5)	88.57 (0.031,0.031,16)	0.62	90.87 (32,0.031,8)	0.12	90.87 (0.031,0.031,16)	0.67	**92.75** (8,8,4)	0.20
Vertebral (310×6×3)	75.05 (0.031,0.5,2)	0.57	70.87 (32,0.031,1)	0.12	**83.81** (0.125,0.031,4)	0.64	82.42 (32,0.031,1)	0.11
Soyabean (47×35×4)	**100** (0.031,2,8)	0.27	**100** (8,0.031,0.5)	0.06	**100** (0.125,0.5,2,0.2)	0.33	**100** (8,0.031,0.25)	0.09
Dermatology (358×34×6)	89.23 (0.125,0.125,256)	0.63	**92.23** (32,0.031,2)	0.14	85.23 (0.5,0.031,256)	0.67	**92.23** (8,0.031,8)	0.16

Please cite this article as: M. Tanveer, A. Sharma and P. N. Suganthan, Least squares KNN-based weighted multiclass twin SVM, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.132

point. Thus, total computational complexity is approximately $\mathcal{O}(\frac{16\ell^3}{27} + \ell^2(\log(\ell))$.

5. Discussion on LS-KWMTSVM

In this section, we discuss the differences between Twin-KSVC, KWMTSVM, LST-KSVC and the proposed LS-KWMTSVM.

5.1. LS-KWMTSVM vs. Twin-KSVC

Twin-KSVC and proposed LS-KWMTSVM employ "one-versus-one-versus-rest" with ternary outputs $\{-1, 0, 1\}$, during the decomposition phase both algorithms use a mixed classification and regression support vector machines formulation and construct $\frac{k(k-1)}{2}$ binary SVM classifiers.

- Twin-KSVC and proposed LS-KWMTSVM implemented with all data points, training avoids the risk of information loss and class distortion problem.
- In Twin-KSVC, data points contribute the same weight for the construction of the hyperplanes, so that local information of training samples is omitted, and inter-class information is not exploited. While proposed LS-KWMTSVM, uses KNN graph approach to utilize the intra-class and inter-class information, different weight matrices are given to data points for the same class.
- In the proposed LS-KWMTSVM, the solution of two primal problems is reduced to solving only two system of linear equations whereas Twin-KSVC needs to solve two QPPs.

5.2. LS-KWMTSVM vs. KWMTSVM

- KWMTSVM and proposed LS-KWMTSVM both employ "one-versus-one-versus-rest" with ternary outputs $\{-1, 0, 1\}$.
- KWMTSVM and proposed LS-KWMTSVM use K-nearest neighbor graph to exploit the local information of the training samples and weight matrix D_1, D_2 are employed in the objective function.
- Weight vectors $F_{v_i}(i = 1, 2), H_v$ are introduced in constraints of both algorithm. If any component of F_{v_i}, H_v is zero then it entail that the consonant constraint is redundant.
- In the proposed LS-KWMTSVM, the solution of two primal problems is reduced to solving only two system of linear equations whereas KWTSVM needs to solve two QPPs.

5.3. LS-KWMTSVM vs. LST-KSVC

- LST-KSVC and proposed LS-KWMTSVM solve two system of linear equation, which are used to obtain a pair of nonparallel optimal hyperplanes for two focused classes.
- LST-KSVC and proposed LS-KWMTSVM both employ "one-versus-one-versus-rest" with ternary outputs $\{-1, 0, 1\}$.
- Similar to Twin-KSVC, in LST-KSVC data points contribute the same weight for the construction of the hyperplanes, so that local information of training samples is omitted, and inter-class information is not exploited. While proposed LS-KWMTSVM, uses KNN graph approach to utilize the intra-class and inter-class information, different weight matrices are given to data points for the same class.
- From Eq. 17, we observe proposed LS-KWMTSVM requires

$$\min_{w_+, b_+ \xi_1, \eta_1} \frac{1}{2}\|D_1(Aw_+ + e_1 b_+)\|^2 + \frac{c_1}{2}\|\xi_1\|^2 + \frac{c_2}{2}\|\eta_1\|^2$$
$$s.t. -F_1(Bw_+ + e_2 b_+) + \xi_1 = F_{v_1},$$
$$-H_1(Cw_+ + e_3 b_+) + \eta_1 = (1-\epsilon)H_v$$

  when D_1, H_1, F_1, are identity matrix and F_{v_1}, H_v are unit vectors, above equation is recast into primal problem of LST-KSVC.
- LST-KSVC can be regarded as special case of LS-KWMTSVM.

6. Numerical experiments

Recently, imbalanced dataset [12] problem has appealed more attention in the field of classification. Imbalance problem occurs when significant irregularity between the probability distribution of datasets in various classes. Most of the classifiers generally confined on imbalanced datasets, which is due to the biased probabilities distribution of the datasets in different classes, given by the imbalance ratio (IR) [41].

We manifest the performance of four algorithms i.e., KWMTSVM [43], LST-KSVC [23], Twin KSVC [44] and the proposed LS-KWMTSVM algorithm. We conduct experiments on eighteen imbalance datasets taken from UCI machine learning repository [22] and KEEL repository [1]. The class imbalance ratios are shown in Table 1. The datasets are iris, zoo, wine, hayes-roth, glass, lenses, contraceptive, teaching evaluation, cleave land, tae, seeds, newthyroid, car, balance, ecoli, vertebal, soyabean and dermatology. In Table 2, total number of samples, attributes and number of classes are denoted by sign "$\cdot \times \cdot \times \cdot$" below the dataset name. For example Iris dataset contains 150 samples and each sample consist of the four attributes classified to three classes is denoted by "$150 \times 4 \times 3$". We evaluate the performance of the proposed LS-KWMTSVM on running time and classification accuracy aspects. We use 10-fold cross-validation in our experiments to compare the performance of the three algorithms. In 10-fold cross validation, the dataset is randomly split into ten subsets, nine of them are used for training and one is used for testing. This process is repeated ten times and performance measure is taken as the average of ten tested results. These four algorithms are implemented in MATLAB 2010b. Gaussian kernel function $K(x, y) = \exp^{-(\|x-y\|^2/\mu^2)}$ is considered as it is often applied and yields great generalization performance, where μ is a parameter.

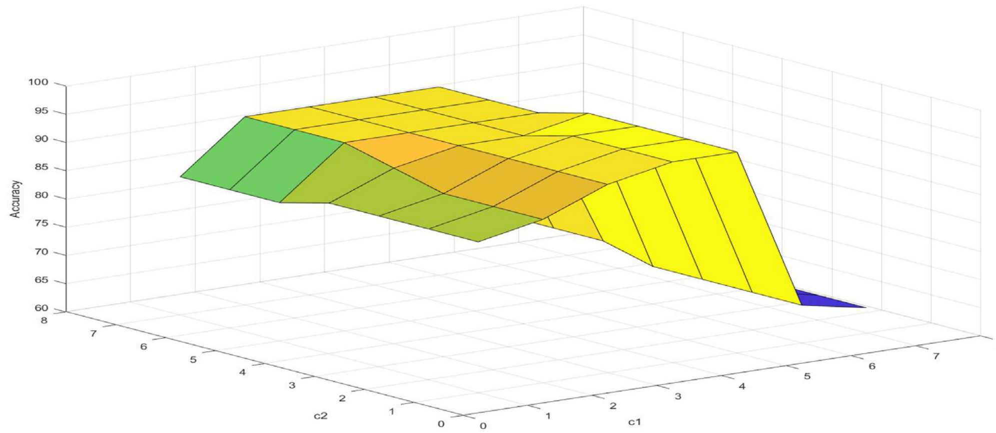
6.1. Parameter selection

It is clear that the performance of the algorithms depend on the choices of parameters. In our experiments, optimal parameters are obtained by the grid search method [13]. For all algorithms, penalty parameters $c_i(i = 1, 2, 3, 4)$ are selected from the set $\{2^j | j = -5, -4, \cdots, 4, 5\}$. The Gaussian kernel parameter μ is elected over the range $\{2^j \mid j = -10, -9, \cdots, 9, 10\}$. Parameter ϵ is prescribed to a small value 0.2. To scale down the computational cost of parameter selection, we prescribed $c_1 = c_3$ and $c_2 = c_4$ for all algorithms.

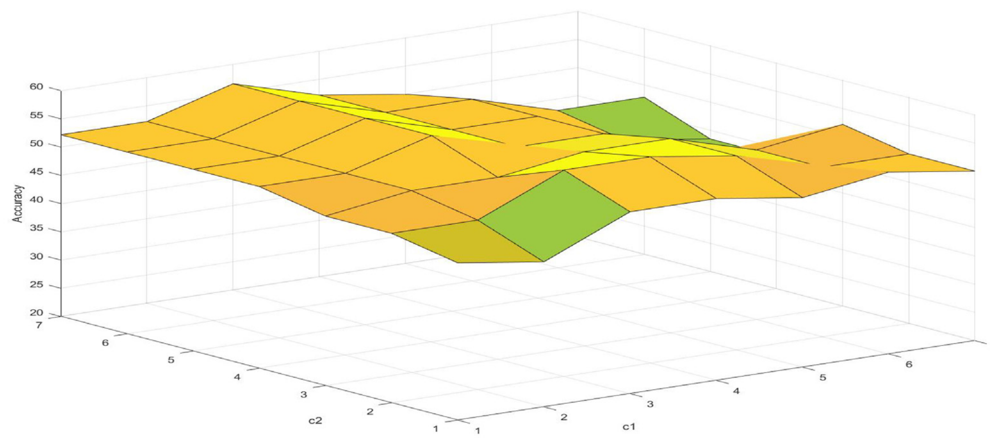
6.2. Result comparison and discussion

We compare the proposed LS-KWMTSVM with Twin-KSVC [44], LST-KSVC[23] and KWMTSVM [45]. Table 2 illustrate the experimental results. We scrutinize that the proposed LS-KWMTSVM outperforms on most of the datasets in the perspective of prediction accuracy. In the proposed algorithm, we solve the system of linear equations which makes the computation speed fast. Fig. (a) manifests the impact of penalty parameters (c_1, c_2) on the performance of LS-KWMTSVM with the optimal value of kernel parameter for the dataset Iris.

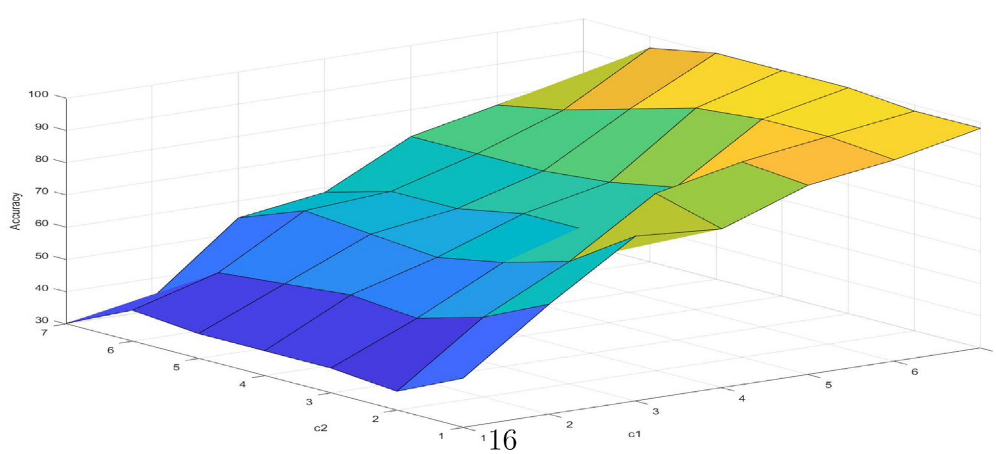
Please cite this article as: M. Tanveer, A. Sharma and P. N. Suganthan, Least squares KNN-based weighted multiclass twin SVM, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.132



(a) Iris



(b) Teaching



(c) Wine

Please cite this article as: M. Tanveer, A. Sharma and P. N. Suganthan, Least squares KNN-based weighted multiclass twin SVM, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.132

Table 3
Average rank on accuracy of four algorithms on twelve benchmark datasets.

Dataset	Twin-KSVC	LST-KSVC	KWMTSVM	LS- KWMTSVM
Iris	1.5	3	4	1.5
Teaching	4	1	3	2
Wine	3.5	1	3.5	2
Hayes	3.5	1.5	3.5	1.5
Glass	4	2	3	1
Lenses	3	3	3	1
Contraceptive	1.5	4	3	1.5
Zoo	3.5	2	3.5	1
Cleveland	2	3.5	3.5	1
Tae	4	3	2	1
Seeds	3	3	1	3
Newthyroid	1	3	4	2
Car good	4	1.5	3	1.5
Balance	4	2.5	2.5	1
Ecoli	4	2.5	2.5	1
Vertebral	3	4	1	2
Soyabean	2.5	2.5	2.5	2.5
Dermatology	3	1	4	2
Average Rank	3.05	2.44	2.91	1.58

One can observe from Fig. (a) that c_1 has more impact on the predictive accuracy of the proposed LS-KWMTSVM as compared to c_2. As value of c_1 increase accuracy also increase linearly. Fig. (b) manifests the impact of penalty parameters (c_1, c_2) on the performance of LS-KWMTSVM with the optimal value of kernel parameter for Teaching dataset. It is observed from Fig. (b) that variation in the value of c_1 and c_2 does not much effect the variation of predictive accuracy. Predicative accuracy of proposed algorithm is approximately constant. Fig. (c) manifests the impact of penalty parameters (c_1, c_2) on the performance of LS-KWMTSVM with the optimal value of kernel parameter for Wine dataset. Fig. (c) manifests that c_1 have more impact on the predictive accuracy as compared to c_2. For large value of c_1, the performance of LS-KWMTSVM suddenly degrades.

6.3. Statistical analysis

To analyze the statistical implication of the proposed LS-KWMTSVM in contrast to Twin-KSVC [44], LST-KSVC [23] and KWMTSVM [45], we use Friedman test with corresponding post hoc tests [9,10] for the 4 algorithms and 18 benchmark datasets. Friedman test is considered to be simple, robust, non-parametric and safe test for comparison of different classifiers over multiple datasets. It ranks the algorithm for each dataset separately, the best performing algorithm getting the rank 1, second one is ranked 2 and so on. In the case of ties, average ranks are assigned. Table 3 expose the average rank of algorithms on accuracy with Gaussian kernel function.

Under the null hypothesis, the Friedman statistics is distributed according to $\mathcal{X}_F^2$ with $(t-1)$ degree of freedom as follows: [9]:

$$\mathcal{X}_F^2 = \frac{12N}{t(t+1)}\left[\sum_j R_j^2 - \frac{t(t+1)^2}{4}\right],$$

$$F_F = \frac{(N-1)\mathcal{X}_F^2}{N(t-1) - \mathcal{X}_F^2},$$

$$\mathcal{X}_F^2 = \frac{12 \times 18}{4(4+1)}\left[3.05^2 + 2.44^2 + 2.91^2 + 1.58^2 - \frac{4 \times 5^2}{4}\right] = 13.18.,$$

$$F_F = \frac{(18-1) \times 13.18}{18 \times (4-1) - 13.18} = 5.48,$$

The rank of j^{th} algorithm on the i^{th} dataset is denoted by r_i^j out of N datasets, where $R_j = \frac{1}{N}\sum_j r_i^j$. F_F is the F-distribution with $(t-1)(N-1) = (4-1, (4-1) \times (18-1)) = (3, 51)$ degrees of freedom, where t is the number of algorithms and N number of datasets. The critical values of F(3, 51) at significance level $\alpha = (0.025, 0.05, 0.1)$ are 3.38, 2.78 and 2.19 respectively. Since F_F value 5.48 of the proposed algorithm is larger than the critical values i.e, the average rank of the proposed algorithm is much lower than other algorithms. One can conclude that the proposed algorithm LS-KWMTSVM is significantly better than Twin-KSVC, LST-KSVC and KWMTSVM.

7. Conclusion

In this paper, a novel least squares KNN-based weighted multi-class twin support vector machine (LS-KWMTSVM) is proposed by adopting the equality constraints instead of inequality constraints and minimized the slack variables using squares of 2-norm instead of conventional 1-norm. This simple modification leads to a very fast algorithm with much better results. The modified primal problems in the proposed LS-KWMTSVM solves only two systems of linear equations whereas two QPPs need to solve in KWMTSVM. The proposed LS-KWMTSVM, equivalent to the KWMTSVM, employed the weight matrix in the objective function to exploit the local information of the training samples. To exploit the inter class information, we use weight vectors in the constraints of the proposed LS-KWMTSVM. If any component of vectors is zero then the corresponding constraint is redundant and thus we can avoid it. Elimination of redundant constraints and solving a system of linear equations instead of QPPs makes the proposed LS-KWMTSVM more robust and faster than KWMTSVM. The proposed LS-KWMTSVM, equivalent to the KWMTSVM, appraise all the training data points into a "1-versus-1-versus-rest" structure, and thus the proposed LS-KWMTSVM inaugurate ternary output $\{-1, 0, +1\}$ which helps to deal with imbalance datasets. Numerical experiments on several UCI benchmark datasets clearly indicate that the proposed LS-KWMTSVM has better accuracy in classification to that of KWMTSVM but with remarkably less computational time. It should be pointed out that there are several parameters in the proposed LS-KWMTSVM, so parameter selection is an important problem and need to address in future.

Please cite this article as: M. Tanveer, A. Sharma and P. N. Suganthan, Least squares KNN-based weighted multiclass twin SVM, Neurocomputing, https://doi.org/10.1016/j.neucom.2020.02.132

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 Science and Engineering Research Board (SERB) under Early Career Research Award Grant No. ECR/2017/000053 and Ramanujan fellowship Grant No. SB/S2/RJN-001/2016. This work was also supported by Council of Scientific & Industrial Research (CSIR), New Delhi, INDIA under Extra Mural Research (EMR) Scheme Grant No. 22(0751)/17/ EMR-II. We gratefully acknowledge the Indian Institute of Technology Indore for providing facilities and support.

References

[1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic &, Soft. Comput. 17 (2011).
[2] C. Angulo, X. Parra, A. Catal, K-SVCR: A support vector machine for multi-class classification, Neurocomputing 55 (2003) 57–77.
[3] L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, L. Jackel, Y. LeCun, U. Muller, E. Sackinger, P. Simard, Comparison of classifier methods: a case study in handwritten digit recognition, in: International Conference on Pattern Recognition, IEEE Computer Society Press, 1994, pp. 77–82.
[4] P. Bradely, O. Mangasarian, Massive data discrimination via linear support vector machine, Optim. Method Softw. 13 (2000) 1–10.
[5] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 273–297.
[6] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[7] S. Datt, S. Das, Near-bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs, Neural Networks 70 (2015) 39–52.
[8] S. Datt, S. Das, Multiobjective support vector machines: Handling class imbalance with pareto optimality, IEEE Trans. Neural Networks Learn. Syst. 30 (2019) 1–25.
[9] J. Demsar, Statistical comparisons of classifiers over multiple datasets, J. Mach. Learn. Res. 7 (2006) 1–30.
[10] S. Garcia, A. Fernandez, J. Luengo, F. Herrera, Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining experimental analysis of power, Inf. Sci. 180 (2010) 2044–2064.
[11] G. Golub, C. Van, Matrix Computations, third edition., John Hopkins University Press, Baltimore, London, 2012.
[12] E. Gracia, H. He, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (2009) 1263–1284.
[13] C. Hsu, C. Lin, A comparision of methods for multiclass support vector machine, IEEE Trans. Neural Networks 13 (2002) 415–425.
[14] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (5) (2007) 905–910.
[15] R. Khemchandani, P. Saigal, S. Chandra, Angle-based twin support vector machine, Ann. Oper. Res. 269 (2018) 387–417.
[16] R. Khemchandani, S. Sharma, Robust least squares twin support vector machine for human activity recognition, Appl. Soft Comput. 47 (2016) 33–46.
[17] U. Krebel, Pairwise classification and support vector machines, in: Advances in kernel methods, MIT Press, Cambridge, MA (1999) 255–268..
[18] H.W. Kuhn, A.W. Tucker, Nonlinear programming, Proceedings of 2nd Berkeley Symposium, University of California Press, Berkeley, 1951, pp. 481–492.
[19] M. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Syst. Appl. 36 (2009) 7535–7543.
[20] Y.Q. Li, C.T. Guan, Joint feature re-extraction and classification using an iterative semi-supervised support vector machine algorithm, Mach. Learn. 71 (1) (2008) 33–53.
[21] O. Mangasarian, E. Wild, Multisurface proximal support vector classification via generalized eigenvalues, IEEE Trans. Pattern Anal. Mach. Intell. 28 (1) (2006) 69–74.
[22] P. Murphy, D. Aha, Uci repository of machine learning databases, http://www.ics.uci.edu/ mlearn (1992)..
[23] J. Nasiri, N. Charkari, S. Jalili, Least squares twin multi-class classification support vector machine, Pattern Recogn. 48 (2015) 984–992.
[24] M. Pal, P. Mather, Support vector machines for classification in remote sensing, Int. J. Remote Sens. 26 (2005) 1007–1011.
[25] X. Peng, TSVR: An efficient twin support vector machine for regression, Neural Network 23 (3) (2010) 365–372.
[26] X. Peng, L. Kong, D. Chen, Improvements on twin parametric-margin support vector machine, Neurocomputing 151 (2015) 857–863.
[27] B. Richhariya, M. Tanveer, EEG signal classification using universum support vector machine, Expert Syst. Appl. 106 (2018) 169–182.
[28] B. Richhariya, M. Tanveer, A reduced universum twin support vector machine for class imbalance learning, Pattern Recogn. 107150 (2020).
[29] Y.H. Shao, W.J. Chen, J.J. Zhang, Z. Wang, N.Y. Deng, An efficient weighted lagrangian twin support vector machine for imbalanced data classification, Pattern Recogn. 47 (2014) 3158–3167.
[30] Y.H. Shao, C.N. Li, M.Z. Liu, Z. Wang, N.Y. Deng, Sparse lq-norm least squares support vector machine with feature selection, Pattern Recogn. 78 (2018) 167–181.
[31] J. Tang, Y. Tian, X. Liu, D. Li, J. Lv, G. Kou, Improved multi-view privileged support vector machine, Neural Networks 106 (2018) 96–109.
[32] M. Tanveer, Newton method for implicit Lagrangian twin support vector machines, Int. J. Mach. Learn. Cybern. 6 (2015) 1029–1040.
[33] M. Tanveer, C. Gautam, P.N. Suganthan, Comprehensive evaluation of twin svm based classifiers on uci datasets, Appl. Soft Comput. 83 (2019) 105617.
[34] M. Tanveer, M.A. Khan, S.S. Ho, Robust energy-based least squares twin support vector machines, Appl. Intell. 45 (1) (2016) 174–186.
[35] M. Tanveer, T. Rajani, M. Ganaie, Improved sparse pinball twin svm, in: 2019 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE..
[36] M. Tanveer, B. Richhariya, R.U. Khan, A.H. Rashid, P. Khanna, M. Prasad, C.T. Lin, Machine learning techniques for the diagnosis of Alzheimer's disease: a review, ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) 16 (1s) (2020) 1–35, https://doi.org/10.1145/3344998.
[37] M. Tanveer, A. Sharma, P.N. Suganthan, General twin support vector machine with pinball loss function, Inf. Sci. 494 (2019) 311–327.
[38] M. Tanveer, A. Tiwari, R. Choudhary, S. Jalan, Sparse pinball twin support vector machines, Appl. Soft Comput. 78 (2019) 164–175.
[39] Y. Tian, Z. Qi, X. Ju, Y. Shi, X. Liu, Nonparallel support vector machines for pattern classification, IEEE Trans. Cybern. 44 (2014) 1067–1079.
[40] A. Tikhonov, V. Arsenin, Solution of Ill Posed Problems, John Wiley and Sons, 1997.
[41] R.M.V. García, J.S. Sánchez, On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, Knowledge Based Systems 25 (2012) 13–21.
[42] Z. Wang, Y.H. Shao, L. Bai, N.Y. Deng, Twin support vector machine for clustering, IEEE Trans. Neural Networks Learn. Syst. 26 (2015) 2583–2588.
[43] Y. Xu, K-nearest neighbor-based weighted twin support vector machine, Neurocomputing 205 (2016) 430–438.
[44] Y. Xu, R. Guo, L. Wang, A twin multi-class classification support vector machine, Cognitive Comput. 5 (2013) 580–588.
[45] Y. Xu, L. Wang, K-nearest neighbor-based weighted twin support vector regression, Appl. Intell. 41 (2014) 299–309.
[46] Y. Xu, L. Wang, P. Zhong, A rough margin-based?-twin support vector machine, Neural Comput. Appl. 21 (6) (2012) 1307–1317.

M. Tanveer is Associate Professor and Ramanujan Fellow at the Discipline of Mathematics of the Indian Institute of Technology, Indore. Prior to that, he spent one year as a Postdoctoral Research Fellow at the Rolls-Royce@NTU Corporate Lab of the Nanyang Technological University, Singapore. His research interests include machine learning, deep learning, optimization, biomedical engineering and applications to Alzheimer's disease and dementias. He has published over 40 referred journal papers of international repute. He is the recipient of the 2016 DST-Ramanujan Fellowship in Mathematical Sciences and 2017 SERB-Early Career Research Award in Engineering Sciences which are the prestigious awards of INDIA at early career level. He is a Senior Member of IEEE, editorial review board member of Applied Intelligence, Springer, Guest Editor of ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), Multimedia Tools and Applications, Springer, Applied Soft Computing, Elsevier and Associate Editor for IEEE SMC 2019. He has also co-edited one book in Springer on machine intelligence and signal analysis. He has also been lead organizer/general chair and invited/-plenary/keynote speaker in many international conferences and Symposiums. He was the Co-Chair of Special Session Proposal in IEEE SSCI 2018. Tanveer is currently the Principal Investigator of 06 major research projects and Co-Principal Investigator of 01 major research project funded by Government of India including Department of Science and Technology (DST), Science & Engineering Research Board (SERB), Council of Scientific & Industrial Research (CSIR) and SPARC – Ministry of Human Resource Development (MHRD). Homepage: http://iiti.ac.in/people/mtanveer/

Anshul Sharma received his M. Sc. degree in Mathematics from the Indian Institute of Technology Indore, India. His research interests include machine learning, support vector machine, and optimization.

Ponnuthurai Nagaratnam Suganthan (or P N Suganthan) received the B.A degree, Postgraduate Certificate and M.A degree in Electrical and Information Engineering from the University of Cambridge, UK in 1990, 1992 and 1994, respectively. After completing his PhD research in 1995, he served as a pre-doctoral Research Assistant in the Dept of Electrical Engineering, University of Sydney in 1995–96 and a lecturer in the Dept of Computer Science and Electrical Engineering, University of Queensland in 1996-99. He moved to NTU in 1999. He is an Editorial Board Member of the Evolutionary Computation Journal, MIT Press (2013-2018). He is an associate editor of the IEEE Trans on Cybernetics (2012 -), IEEE Trans on Evolutionary Computation (2005 -), Information Sciences (Elsevier) (2009 -), Pattern Recognition (Elsevier) (2001 -), Applied Soft Computing (2018-) and Int. J. of Swarm Intelligence Research (2009 -) Journals. He is a founding co-editor-in-chief of Swarm and Evolutionary Computation (2010 -), an SCI Indexed Elsevier Journal. His co-authored SaDE paper (published in April 2009) won the "IEEE Trans. on Evolutionary Computation outstanding paper award" in 2012. His former PhD student, Dr Jane Jing Liang, won the IEEE CIS Outstanding PhD dissertation award, in 2014. His research interests include swarm and evolutionary algorithms, pattern recognition, big data, deep learning and applications of swarm, evolutionary & machine learning algorithms. He was selected as one of the highly cited researchers by Thomson Reuters in 2015, 2016, 2017, and 2018 in computer science. He served as the General Chair of the IEEE SSCI 2013. He has been a member of the IEEE since 1991 and Fellow since 2015. He was an elected AdCom member of the IEEE Computational Intelligence Society (CIS) in 2014-2016. He is an IEEE CIS distinguished lecturer (2018–2020).