


Robust energy-based least squares twin support vector machines

Mohammad Tanveer¹  · Mohammad Asif Khan² · Shen-Shyang Ho¹

Published online: 4 February 2016
© Springer Science+Business Media New York 2016

Abstract Twin support vector machine (TSVM), least squares TSVM (LST SVM) and energy-based LST SVM (ELS-TSVM) satisfy only empirical risk minimization principle. Moreover, the matrices in their formulations are always positive semi-definite. To overcome these problems, we propose in this paper a robust energy-based least squares twin support vector machine algorithm, called RELS-TSVM for short. Unlike TSVM, LST SVM and ELS-TSVM, our RELS-TSVM maximizes the margin with a positive definite matrix formulation and implements the structural risk minimization principle which embodies the marrow of statistical learning theory. Furthermore, RELS-TSVM utilizes energy parameters to reduce the effect of noise and outliers. Experimental results on several synthetic and real-world benchmark datasets show that RELS-TSVM not only yields better classification performance but also has a lower training time compared to ELS-TSVM, LSPTSVM, LST SVM, TBSVM and TSVM.

Keywords Machine learning · Support vector machines · Twin support vector machines · Least squares twin support vector machines

1 Introduction

Support vector machines (SVMs) [2, 4, 5, 34, 35], have already gained a great deal of attention due to their good generalization ability on high dimensional data. There are three key elements which make SVMs successful: (i) maximizing the margin around the separating hyperplane between two classes that leads to solving a convex quadratic programming problem (QPP), (ii) dual theory makes introducing the kernel function possible, (iii) and kernel trick is applied to solve nonlinear case. One of the main challenges for SVM is the large computational complexity of QPP. This drawback restricts the application of SVM to large-scale problems. To reduce the computational complexity of SVM, various algorithms with comparable classification abilities have been proposed, including SVM^{light} [13], SMO [25], Chunking algorithm [4], LIBSVM [3], Lagrangian SVM (LSVM) [19], Reduced SVM (RSVM) [17], Smooth SVM (SSVM) [18], Proximal SVM [8], LPSVR [31] and others.

Recently, research on nonparallel hyperplane classifiers has been an interesting trend. Unlike the standard SVM, which uses a single hyperplane, some recently proposed approaches, such as the generalized eigenvalue proximal support vector machine (GEP SVM) [20] and twin support vector machine (TSVM) [12], use two nonparallel hyperplanes. Experimental results show that the nonparallel hyperplanes can effectively improve the performance over SVM [12, 20]. Due to its strong generalization ability, some scholars proposed variants of TSVM [1, 11, 14, 15, 21, 24, 27, 28, 30, 32, 33, 36, 37]. Specifically, least squares twin

✉ Mohammad Tanveer
tanveergouri@gmail.com

Mohammad Asif Khan
m.asif.y12@lnmiit.ac.in

Shen-Shyang Ho
ssho@ntu.edu.sg

¹ School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore, 639798, Singapore

² Department of Electronics and Communication Engineering, The LNM Institute of Information Technology, Jaipur 302 031 India

support vector machine (LSTSVM) [14] has been proposed as a way to replace the convex QPPs in TSVM with a convex linear system by using the squared loss function instead of the hinge one, leading to very fast training speed. However, LSTSVM is sensitive to noise and outliers due to the construction of the constraints of LSTSVM that require the hyperplane to be at distance of exactly 1 from the points of the other class. Recently, Nasiri et al. [23] proposed an energy-based model of LSTSVM (ELS-TSVM) by introducing an energy term for each hyperplane to reduce the effect of noise and outliers. ELS-TSVM not only considers the different energy for each class, it also handles unbalanced datasets [23]. Different from TSVM, LSTSVM and ELS-TSVM, Shao et al. [28, 29] introduces an extra regularization term to each objective function in twin bounded support vector machine (TBSVM) and least squares recursive projection twin support vector machine (LSPTSVM), ensuring the optimization problems are positive definite and resulting in better generalization ability.

In this paper, we present an improved version of ELS-TSVM [23], called robust energy-based least squares twin support vector machines (RELS-TSVM). Our RELS-TSVM possesses the following attractive advantages:

- Unlike TSVM, LSTSVM and ELS-TSVM, our RELS-TSVM introduces regularization term to each objective function with the idea of maximizing the margin. Furthermore, the structural risk minimization principle is implemented in our formulation due to this extra term which embodies the marrow of statistical learning theory.
- Similar to ELS-TSVM, our RELS-TSVM also introduces an energy for each hyperplane to reduce the effect of noise and outliers which makes our algorithm more robust.
- Our RELS-TSVM solves two systems of linear equations rather than solving two quadratic programming problems (QPPs) in TSVM and TBSVM, and one large QPP in SVM, which makes the learning speed of RELS-TSVM faster than TBSVM, TSVM and SVM.
- The decision function of our RELS-TSVM is obtained directly from the primal problems. However, perpendicular distance is calculated to obtain the decision function in LSPTSVM, LSTSVM, TBSVM and TSVM.
- Our RELS-TSVM does not require any special optimizer.

Numerical experiments on several benchmark datasets show that our RELS-TSVM gains better classification ability with less training time in comparison with TSVM, TBSVM, LSTSVM, LSPTSVM and ELS-TSVM.

The rest of this paper is organized as follows. Section 2 provides a brief introduction to TSVM, LSTSVM and ELS-TSVM formulations. Section 3 describes the detail

of RELS-TSVM, including linear and nonlinear versions. Numerical experiments are performed and their results are compared with TSVM, TBSVM, LSTSVM, LSPTSVM and ELS-TSVM in Section 4. Finally, we conclude our work in Section 5.

2 Background

In this section, we give a brief outline of TSVM, LSTSVM and ELS-TSVM formulations. For a more detailed description, the interested readers can refer to [12, 14, 23].

2.1 Twin support vector machines (TSVM)

Suppose that all the data points in class +1 are denoted by a matrix $A \in R^{m_1 \times n}$, where the i^{th} row $A_i \in R^n$, and the matrix $B \in R^{m_2 \times n}$ represents the data points of class -1. Unlike SVM, the linear TSVM [12] seeks a pair of non-parallel hyperplanes

$$f_1(x) = w_1^t x + b_1 \quad \text{and} \quad f_2(x) = w_2^t x + b_2 \tag{1}$$

such that each hyperplane is close to the data points of one class and far from the data points of other class, where $w_1 \in R^n$, $w_2 \in R^n$, $b_1 \in R$ and $b_2 \in R$. The formulation of TSVM can be written as follows:

$$\begin{aligned} \min_{(w_1, b_1) \in R^{n+1}} \quad & \frac{1}{2} \|Aw_1 + e_2 b_1\|^2 + c_1 \|\xi_1\| \\ \text{s.t.} \quad & -(Bw_1 + e_1 b_1) + \xi_1 \geq e_1, \quad \xi_1 \geq 0 \end{aligned} \tag{2}$$

and

$$\begin{aligned} \min_{(w_2, b_2) \in R^{n+1}} \quad & \frac{1}{2} \|Bw_2 + e_1 b_2\|^2 + c_2 \|\xi_2\| \\ \text{s.t.} \quad & (Aw_2 + e_2 b_2) + \xi_2 \geq e_2, \quad \xi_2 \geq 0 \end{aligned} \tag{3}$$

respectively, where c_1, c_2 are positive parameters and e_1, e_2 are vectors of one of appropriate dimensions. The idea in TSVM is to solve two QPPs (2) and (3), each of the QPPs in the TSVM pair is a typical SVM formulation, except that not all data points appear in the constraints of either problem [12].

In order to derive the corresponding dual problems, TSVM assumes that the matrices $G^t G$ and $H^t H$ are nonsingular, where $G = [A \ e_2]$ and $H = [B \ e_1]$ are augmented matrices of sizes $m_1 \times (n+1)$ and $m_2 \times (n+1)$, respectively. Under this extra condition, the dual problems are

$$\begin{aligned} \max_{\alpha \in R^{m_2}} \quad & e_1^t \alpha - \frac{1}{2} \alpha^t H (G^t G)^{-1} H^t \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 \end{aligned} \tag{4}$$

and

$$\begin{aligned} \max_{\gamma \in R^{m_1}} \quad & e_2^t \gamma - \frac{1}{2} \gamma^t G (H^t H)^{-1} G^t \gamma \\ \text{s.t.} \quad & 0 \leq \gamma \leq c_2 \end{aligned} \tag{5}$$

respectively.

In order to deal with the case when $G^t G$ or $H^t H$ is singular and avoid the possible ill conditioning, the inverse matrices $(G^t G)^{-1}$ and $(H^t H)^{-1}$ are approximately replaced by $(G^t G + \delta I)^{-1}$ and $(H^t H + \delta I)^{-1}$, respectively, where δ is a very small positive scalar and I is an identity matrix of appropriate dimensions. Thus, the above dual problems are modified as:

$$\begin{aligned} \max_{\alpha \in R^{m_2}} \quad & e_1^t \alpha - \frac{1}{2} \alpha^t H (G^t G + \delta I)^{-1} H^t \alpha \\ \text{s.t.} \quad & 0 \leq \alpha \leq c_1 \end{aligned} \tag{6}$$

and

$$\begin{aligned} \max_{\gamma \in R^{m_1}} \quad & e_2^t \gamma - \frac{1}{2} \gamma^t G (H^t H + \delta I)^{-1} G^t \gamma \\ \text{s.t.} \quad & 0 \leq \gamma \leq c_2 \end{aligned} \tag{7}$$

respectively.

Thus, the nonparallel proximal hyperplanes are obtained from the solution α and γ of (6) and (7) by

$$\begin{aligned} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} &= -(G^t G + \delta I)^{-1} H^t \alpha \quad \text{and} \quad \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} \\ &= (H^t H + \delta I)^{-1} G^t \gamma. \end{aligned} \tag{8}$$

The dual problems for (4) and (5) are derived and solved in [12].

A new sample $x \in R^n$ is assigned to a class i ($i = +1, -1$) by comparing the following perpendicular distance measure of it from the two hyperplanes (1):

$$\text{Class } i = \arg \min_{i=1,2} \frac{|x^t w_i + b_i|}{\|w_i\|}. \tag{9}$$

Experimental results show that the performance of TSVM is better than the conventional SVM and GEPSVM on UCI machine learning datasets. The case of nonlinear kernels is handled similar to linear kernels [12].

2.2 Least squares twin support vector machines (LSTSVM)

Similar to TSVM, least squares TSVM (LSTSVM) [14] also seeks a pair of non-parallel hyperplanes (1). It assigns the training points to the closer one of two non-parallel proximal hyperplanes and pushes them apart from the distance of 1. LSTSVM is an extremely fast and simple algorithm that requires only solution of a system of linear equations for generating both linear and non-linear classifiers. By replacing the inequality constraints with equality constraints and

taking the squares of 2-norm of slack variables instead of 1-norm, the primal problems of LSTSVM can be expressed as

$$\begin{aligned} \min_{(w_1, b_1) \in R^{n+1}} \quad & \frac{1}{2} \|Aw_1 + e_2 b_1\|^2 + \frac{c_1}{2} \|\xi_1\|^2 \\ \text{s.t.} \quad & -(Bw_1 + e_1 b_1) + \xi_1 = e_1, \end{aligned} \tag{10}$$

$$\begin{aligned} \min_{(w_2, b_2) \in R^{n+1}} \quad & \frac{1}{2} \|Bw_2 + e_1 b_2\|^2 + \frac{c_2}{2} \|\xi_2\|^2 \\ \text{s.t.} \quad & (Aw_2 + e_2 b_2) + \xi_2 = e_2. \end{aligned} \tag{11}$$

The linear LSTSVM completely solves the classification problem with just two matrix inverses of much smaller dimension of order $(n + 1) \times (n + 1)$ [14]. Once we get the solutions of (10) and (11), the two nonparallel hyperplanes are obtained by solving two systems of linear equations:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -[c_1 Q^t Q + P^t P]^{-1} c_1 Q^t e_1, \tag{12}$$

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = [c_2 P^t P + Q^t Q]^{-1} c_2 P^t e_2, \tag{13}$$

where c_1 and c_2 are positive penalty parameters, $P = [A \ e]$ and $Q = [B \ e]$.

2.3 Energy-based least squares twin support vector machines (ELS-TSVM)

Least squares twin support vector machines (LSTSVM) are sensitive to noise and outliers in the training dataset. Recently, Nasiri et al. [23] proposed a novel energy-based model of LSTSVM (ELS-TSVM) by introducing an energy term for each hyperplane to reduce the effect of noise and outliers.

The linear ELS-TSVM comprises of the following pair of minimization problems:

$$\begin{aligned} \min_{(w_1, b_1) \in R^{n+1}} \quad & \frac{1}{2} \|Aw_1 + eb_1\|^2 + \frac{c_1}{2} \xi_1^t \xi_1 \\ \text{s.t.} \quad & -(Bw_1 + eb_1) + \xi_1 = E_1, \end{aligned} \tag{14}$$

$$\begin{aligned} \min_{(w_2, b_2) \in R^{n+1}} \quad & \frac{1}{2} \|Bw_2 + eb_2\|^2 + \frac{c_2}{2} \xi_2^t \xi_2 \\ \text{s.t.} \quad & (Aw_2 + eb_2) + \xi_2 = E_2, \end{aligned} \tag{15}$$

where c_1 and c_2 are positive parameters, E_1 and E_2 are energy parameters of the hyperplanes. Let us first discuss ELS-TSVM with LSTSVM.

- The constraints of ELS-TSVM and LSTSVM are different. The constraints of LSTSVM require the hyperplane to be at a distance of exactly 1 from points of other class that makes LSTSVM be sensitive to outliers. On the other hand, ELS-TSVM introduces an energy term for each hyperplane and different energy parameters are

selected according to prior knowledge or grid search method to reduce the effect of noise and outliers.

- The decision function of ELS-TSVM is obtained directly from the primal problems. However, the perpendicular distance is calculated to obtain the decision function in LSTSV M.
- Both ELS-TSVM and LSTSV M are least squares version of TSVM to replace the convex QPPs in TSVM with a convex linear systems. This makes ELS-TSVM and LSTSV M algorithms extremely fast with generalization performance better than TSVM.

On substituting the equality constraints into the objective function, QPP (14) becomes:

$$L_1 = \frac{1}{2} \| Aw_1 + eb_1 \|^2 + \frac{c_1}{2} \| Bw_1 + eb_1 + E_1 \|^2. \tag{16}$$

Setting the gradient of (16) with respect to w_1 and b_1 to zero gives the solution of QPP (14) as follows:

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = -[c_1 Q^t Q + P^t P]^{-1} c_1 Q^t E_1, \tag{17}$$

where $P = [A \ e]$ and $Q = [B \ e]$.

In an exactly similar way the solution of QPP (15) can be obtained as follows:

$$\begin{bmatrix} w_2 \\ b_2 \end{bmatrix} = [c_2 P^t P + Q^t Q]^{-1} c_2 P^t E_2. \tag{18}$$

A new sample x_i is assigned to a class $i (i = +1, -1)$, depending on the following decision function:

$$f(x_i) = \begin{cases} +1 & \text{if } \left| \frac{x_i w_1 + eb_1}{x_i w_2 + eb_2} \right| \leq 1 \\ -1 & \text{if } \left| \frac{x_i w_1 + eb_1}{x_i w_2 + eb_2} \right| > 1 \end{cases} \tag{19}$$

where $|\cdot|$ is the absolute value.

The solutions of (17) and (18) demand the computation of the inverse matrices $[c_1 Q^t Q + P^t P]$ and $[c_2 P^t P + Q^t Q]$ of order $(n + 1)$ respectively, and thus may not be well conditioned in some situations. To overcome this difficulty, a regularization term δI is introduced so that the matrices become positive definite with $\delta > 0$ being chosen to be very small.

For a detailed study on ELS-TSVM, the interested reader is referred to [23].

3 Robust energy-based least squares twin support vector machines (RELS-TSVM)

The constraints of LSTSV M require the hyperplane to be at a distance of exactly 1 from data points of the other class. This makes LSTSV M sensitive to outliers. To address this problem, ELS-TSVM introduces an energy term for each

hyperplane to reduce the effect of noise and outliers. However, each of them involves the empirical risk minimization principle, which easily leads to the overfitting problem, and reduces the prediction accuracies of classifiers. To overcome this difficulty, we add an extra regularization term to each objective function and present a new algorithm called robust energy-based least squares twin support vector machines (RELS-TSVM) for classification problems which makes our algorithm robust to noise and outliers.

3.1 Linear RELS-TSVM

By introducing the regularization terms to the framework of ELS-TSVM, the linear RELS-TSVM comprises of the following pair of minimization problems:

$$\begin{aligned} \min_{(w_1, b_1) \in R^{n+1}} & \frac{1}{2} \| Aw_1 + eb_1 \|^2 + \frac{c_1}{2} \xi_1^t \xi_1 + \frac{c_3}{2} \left\| \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \right\|^2 \\ \text{s.t.} & -(Bw_1 + eb_1) + \xi_1 = E_1, \end{aligned} \tag{20}$$

$$\begin{aligned} \min_{(w_2, b_2) \in R^{n+1}} & \frac{1}{2} \| Bw_2 + eb_2 \|^2 + \frac{c_2}{2} \xi_2^t \xi_2 + \frac{c_4}{2} \left\| \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} \right\|^2 \\ \text{s.t.} & (Aw_2 + eb_2) + \xi_2 = E_2, \end{aligned} \tag{21}$$

where c_1, c_2, c_3 and c_4 are positive parameters, E_1 and E_2 are energy parameters of the hyperplanes.

3.1.1 Discussion on RELS-TSVM

It is well known that the classical SVM implements the structural risk minimization principle. However, ELS-TSVM only implements empirical risk minimization which makes it less robust. To overcome this problem, we introduce a regularization term to each objective function with the idea of maximizing the margin, ensuring the optimization problems in our RELS-TSVM are positive definite and implements the structural risk minimization principle. Similar to ELSTSV M, our RELS-TSVM also uses an energy term for each hyperplane to reduce the effect of noise and outliers which makes our algorithm more robust than LSTSV M and TSVM. The decision function similar to ELS-TSVM is obtained directly from the primal problems. However, the perpendicular distance is calculated to obtain the decision function in LSTSV M and TSVM. Our RELS-TSVM solves two systems of linear equations rather than solving two quadratic programming problems (QPPs) in TSVM and one large QPP in SVM, which makes the learning speed of RELS-TSVM faster than TSVM and SVM. We also extend our numerical experiments for nonlinear kernel. It is worthwhile to note that our RELS-TSVM does not require any special optimizer.

On substituting the equality constraints into the objective function, QPP (20) becomes:

$$L_1 = \frac{1}{2} \|Aw_1 + eb_1\|^2 + \frac{c_1}{2} \|Bw_1 + eb_1 + E_1\|^2 + \frac{c_3}{2} \left\| \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \right\|^2. \quad (22)$$

By taking the partial derivatives with respect to w_1 and b_1 , we get

$$A^t(Aw_1 + eb_1) + c_1 B^t(E_1 + Bw_1 + eb_1) + c_3 w_1 = 0, \quad (23)$$

$$e^t(Aw_1 + eb_1) + c_1 e^t(E_1 + Bw_1 + eb_1) + c_3 b_1 = 0. \quad (24)$$

On combining (23) and (24), we obtain

$$\begin{bmatrix} A^t \\ e^t \end{bmatrix} \left(\begin{bmatrix} A & e \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \right) + c_1 \begin{bmatrix} B^t \\ e^t \end{bmatrix} \left(E_1 + \begin{bmatrix} B & e \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \right) + c_3 \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} = 0. \quad (25)$$

Let $z_1 = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$, $P = [A \ e]$ and $Q = [B \ e]$, the solution becomes:

$$z_1 = -(c_1 Q^t Q + P^t P + c_3 I)^{-1} c_1 Q^t E_1. \quad (26)$$

In a similar way, one obtains the solution of QPP (21) as follows:

$$z_2 = (c_2 P^t P + Q^t Q + c_4 I)^{-1} c_2 P^t E_2. \quad (27)$$

Once two vectors z_1 and z_2 are obtained, the training stage of linear RELS-TSVM is completed. The label of an unknown data point x_i is obtained as in (19).

Remark 1 It should be pointed out that both $(c_1 Q^t Q + P^t P + c_3 I)$ and $(c_2 P^t P + Q^t Q + c_4 I)$ are positive definite matrices due to the extra regularization term, which makes our RELS-TSVM more robust and stable than that of LSTSVM and ELS-TSVM.

3.2 Nonlinear RELS-TSVM

In order to extend our results to nonlinear case, we consider the following kernel-generated surfaces:

$$K(x^t, C^t)w_1 + b_1 = 0 \text{ and } K(x^t, C^t)w_2 + b_2 = 0, \quad (28)$$

where $C = [A \ ; \ B]$ and K is an appropriately chosen kernel.

Similar to the linear case, the above two kernel-generated surfaces are obtained through the following two QPPs:

$$\begin{aligned} \min_{(w_1, b_1) \in R^{m+1}} \quad & \frac{1}{2} \|K(A, C^t)w_1 + eb_1\|^2 + \frac{c_1}{2} \xi_1^t \xi_1 + \frac{c_3}{2} \left\| \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} \right\|^2 \\ \text{s.t.} \quad & -(K(B, C^t)w_1 + eb_1) + \xi_1 = E_1, \end{aligned} \quad (29)$$

$$\begin{aligned} \min_{(w_2, b_2) \in R^{m+1}} \quad & \frac{1}{2} \|K(B, C^t)w_2 + eb_2\|^2 + \frac{c_2}{2} \xi_2^t \xi_2 + \frac{c_4}{2} \left\| \begin{bmatrix} w_2 \\ b_2 \end{bmatrix} \right\|^2 \\ \text{s.t.} \quad & (K(A, C^t)w_2 + eb_2) + \xi_2 = E_2, \end{aligned} \quad (30)$$

where $K(A, C^t)$ and $K(B, C^t)$ are kernel matrices of sizes $m_1 \times m$ and $m_2 \times m$ respectively, where $m = m_1 + m_2$.

Similar to the linear case, the solutions of (29) and (30) are

$$z_1 = -(c_1 N^t N + M^t M + c_3 I)^{-1} c_1 N^t E_1 \quad (31)$$

$$z_2 = (c_2 M^t M + N^t N + c_4 I)^{-1} c_2 M^t E_2 \quad (32)$$

respectively, where $N = [K(B, C^t) \ e]$ and $M = [K(A, C^t) \ e]$.

Note that both matrices $(c_1 N^t N + M^t M + c_3 I)$ and $(c_2 M^t M + N^t N + c_4 I)$ are positive definite, which makes our nonlinear RELS-TSVM more robust and stable than that of LSTSVM and ELS-TSVM.

Further, it can be noted that the solution of nonlinear RELS-TSVM requires inversion of matrix size $(m + 1) \times (m + 1)$ twice. Therefore, to reduce the computation cost, the Sherman-Morrison-Woodbury (SMW) formula [9] is used to approximate (31) and (32) as

$$z_1 = - \left(S - SN^t \left(\frac{I}{c_1} + NSN^t \right)^{-1} NS \right) \times c_1 N^t E_1 \quad (33)$$

$$z_2 = \left(T - TM^t \left(\frac{I}{c_2} + MTM^t \right)^{-1} MT \right) \times c_2 M^t E_2 \quad (34)$$

where $S = (M^t M + c_3 I)^{-1}$ and $T = (N^t N + c_4 I)^{-1}$.

In order to reduce the dimensionality in our nonlinear RELS-TSVM, a reduced kernel technique [17] may also be useful if the size of the training data becomes very large.

Unlike ELS-TSVM, our RELS-TSVM need not care about matrix singularity. It is worthwhile to note that c_3 and c_4 are used as penalty parameters rather than perturbation terms.

Once the two vectors z_1 and z_2 are obtained, the training stage of nonlinear RELS-TSVM is completed. The label of an unknown data point x_i is assigned to class i ($i = +1, -1$), depending on the following decision function.

$$f(x_i) = \begin{cases} +1 & \text{if } \left| \frac{K(x_i, C^t)w_1 + eb_1}{K(x_i, C^t)w_2 + eb_2} \right| \leq 1 \\ -1 & \text{if } \left| \frac{K(x_i, C^t)w_1 + eb_1}{K(x_i, C^t)w_2 + eb_2} \right| > 1 \end{cases} \quad (35)$$

where $|\cdot|$ is the absolute value.

4 Numerical experiments

In this section, we present experimental results on several synthetic and real-world benchmark datasets to analyze the classification accuracies and computational efficiencies of

Table 1 Performance comparison of RELS-TSVM with ELS-TSVM, LSPTSVM, TBSVM and TSVM for linear kernel

Datasets (Train size, Test size)	TSVM Acc.(%), Time(s) ($c_1=c_2$)	TBSVM Acc.(%), Time(s) ($c_1=c_2, c_3=c_4$)	LSTSVM Acc.(%), Time(s) ($c_1=c_2$)	LSPTSVM Acc.(%), Time(s) ($c_1=c_2, c_3=c_4$)	ELS-TSVM Acc.(%), Time(s) ($c_1=c_2, E_1, E_2$)	RELS-TSVM Acc.(%), Time(s) ($c_1=c_2, c_3=c_4, E_1, E_2$)
Cross Planes (80 × 2, 70 × 2)	95.71, 0.0748 (2 ⁻⁵)	97.14 , 0.0012 (2 ⁻⁵ , 2 ⁻⁴)	91.42, 0.0006 (2 ⁻⁵)	97.14 , 0.007 (2 ⁻⁵ , 2 ⁻⁵)	94.29, 0.0021 (2 ⁻⁵ , 0.6, 0.7)	95.71, 0.0009 (2 ⁻⁵ , 2 ⁻³ , 0.6, 0.8)
Ripley (250 × 2, 1000 × 2)	85.40, 0.1372 (2 ³)	84.30, 0.0098 (2 ³ , 2 ²)	85.90, 0.0046 (2 ⁻⁵)	82.40, 0.0046 (2 ⁰ , 2 ⁰)	81.8, 0.0198 (2 ⁻⁵ , 1, 0.7)	86.00 , 0.0086 (2 ³ , 2 ⁻³ , 1, 0.9)
Heart-c (177 × 13, 120 × 13)	69.17, 0.1530 (2 ⁰)	71.67 , 0.0032 (2 ⁷ , 2 ⁴)	67.50, 0.0013 (2 ⁻¹)	57.50, 0.010 (2 ⁻³ , 2 ⁰)	65.00, 0.0035 (2 ⁰ , 0.8, 1)	70.83, 0.0028 (2 ⁰ , 2 ⁰ , 0.7, 0.8)
Ionosphere (246 × 33, 105 × 33)	84.76, 0.1554 (2 ³)	85.71, 0.0126 (2 ⁻⁶ , 2 ⁵)	85.71, 0.0142 (2 ⁻⁵)	88.57, 0.0113 (2 ⁻⁵ , 2 ⁻³)	85.71, 0.0169 (2 ⁻³ , 1, 0.7)	88.87 , 0.0166 (2 ⁻⁵ , 2 ⁻⁵ , 0.8, 0.6)
Heart-statlog (200 × 13, 70 × 13)	82.85, 0.0696 (2 ⁻³)	80.32, 0.0046 (2 ⁻⁸ , 2 ⁵)	82.85, 0.0012 (2 ⁻³)	82.85, 0.0008 (2 ¹ , 2 ²)	81.43, 0.0084 (2 ⁻⁵ , 0.7, 0.6)	84.29 , 0.0027 (2 ⁻⁵ , 2 ³ , 0.9, 1)
Bupa Liver (241 × 6, 104 × 6)	64.42, 0.0231 (2 ⁻⁵)	68.26 , 0.0029 (2 ⁻⁷ , 2 ³)	64.42, 0.0012 (2 ⁻¹)	67.30, 0.0010 (2 ¹ , 2 ⁻⁵)	62.5, 0.0044 (2 ⁻⁵ , 0.7, 0.9)	66.34, 0.0033 (2 ⁻⁵ , 2 ⁻⁵ , 0.8, 0.7)
WPBC (137 × 34, 57 × 34)	78.94, 0.0666 (2 ⁻⁵)	80.71 , 0.0340 (2 ⁻⁷ , 2 ⁻⁵)	66.67, 0.0013 (2 ⁻¹)	77.19, 0.011 (2 ⁻⁵ , 2 ⁻³)	71.93, 0.0032 (2 ⁻³ , 0.8, 0.6)	80.71 , 0.0024 (2 ⁻¹ , 2 ⁻³ , 0.8, 0.9)
Pima-Indians (537 × 8, 231 × 8)	79.22, 0.4732 (2 ⁰)	79.22, 0.0153 (2 ³ , 2 ⁷)	78.78, 0.0034 (2 ⁻³)	80.09 , 0.0029 (2 ⁰ , 2 ¹)	79.22, 0.0149 (2 ⁻³ , 0.8, 1.0)	80.09 , 0.0149 (2 ⁻³ , 2 ⁻⁵ , 0.8, 1.0)
Cleve (177 × 13, 120 × 13)	80.00, 0.0302 (2 ⁻⁵)	78.33, 0.0031 (2 ⁻³ , 2 ⁴)	81.67, 0.0013 (2 ⁻⁵)	83.33, 0.0016 (2 ⁻⁵ , 2 ⁰)	78.33, 0.0033 (2 ⁻⁵ , 0.7, 0.9)	84.17 , 0.0026 (2 ⁻⁵ , 2 ⁻¹ , 0.6, 0.6)
German (800 × 24, 200 × 24)	77.50, 1.7374 (2 ⁰)	78.00, 0.0166 (2 ⁻³ , 2 ⁴)	79.00 , 0.0097 (2 ⁻³)	72.00, 0.0089 (2 ⁰ , 2 ⁻¹)	78.00, 0.0271 (2 ⁻³ , 0.8, 0.7)	79.00 , 0.0042 (2 ⁻¹ , 2 ³ , 1, 0.9)
Australian (540 × 14, 150 × 14)	85.33, 0.3146 (2 ⁻⁴)	85.33, 0.0039 (2 ⁷ , 2 ⁵)	86.67 , 0.0038 (2 ⁻⁵)	85.33, 0.0042 (2 ⁻⁵ , 2 ⁻³)	86.67 , 0.0143 (2 ¹ , 1, 0.7)	86.67 , 0.0033 (2 ¹ , 2 ⁻⁵ , 1, 0.7)
Transfusion (600 × 4, 148 × 4)	79.73, 0.2212 (2 ¹)	84.45, 0.0162 (2 ² , 2 ⁰)	83.78, 0.0027 (2 ⁻³)	83.10, 0.0176 (2 ⁻⁵ , 2 ⁻⁵)	83.78, 0.0073 (2 ⁰ , 1, 0.8)	86.49 , 0.0181 (2 ¹ , 2 ⁻¹ , 0.7, 1)
Sonar (150 × 60, 58 × 60)	65.51, 0.0744 (2 ⁻³)	70.69, 0.0028 (2 ⁻⁵ , 2 ⁶)	63.79, 0.0020 (2 ⁻³)	67.24, 0.0018 (2 ⁻³ , 2 ³)	68.97, 0.0067 (2 ⁻⁵ , 0.9, 1.0)	74.14 , 0.0032 (2 ⁻³ , 2 ³ , 0.6, 0.6)

Bold type shows the best result

* Acc. denotes accuracy

Table 2 Performance comparison of RELS-TSVM with ELS-TSVM, LSTSV, TBSVM and TSVM for Gaussian kernel

Datasets (Train size, Test size)	TSVM		TBSVM		LSTSV		ELS-TSVM		RELS-TSVM	
	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc. (%)	Time (s)
Ripley (250 × 2, 1000 × 2)	88.00, 1.2213 (2 ⁻³ , 2 ⁻⁵)	83.50, 0.1874 (2 ⁻¹ , 2 ³ , 2 ⁵)	86.40, 0.19246 (2 ⁻¹ , 2 ⁻¹)	83.50, 0.1137 (2 ⁻⁵ , 0.7, 0.8, 2 ³)	88.30 , 0.1747 (2 ⁻⁵ , 2 ⁻⁵ , 0.6, 0.8, 2 ⁴)					
Heart-c (177 × 13, 120 × 13)	60.81, 0.0644 (2 ⁻⁵ , 2 ⁻⁴)	70.00 , 0.0087 (2 ⁻¹ , 2 ³ , 2 ⁻³)	61.67, 0.0057 (2 ⁻⁵ , 2 ⁻⁴)	61.67, 0.0081 (2 ¹ , 2 ⁰ , 0.6, 0.7, 2 ⁻³)	67.50, 0.0075 (2 ¹ , 2 ⁰ , 0.6, 0.7, 2 ⁻³)					
Heart-statlog (200 × 13, 70 × 13)	82.85, 0.4604 (2 ³ , 2 ⁵)	84.28, 0.0793 (2 ⁻⁸ , 2 ³ , 2 ²)	85.71 , 0.0797 (2 ⁻¹ , 2 ⁸)	78.57, 0.0847 (2 ⁻³ , 0.8, 1, 2 ²)	85.71 , 0.0817 (2 ⁻¹ , 2 ⁻³ , 0.6, 1, 2 ⁰)					
Ionosphere (246 × 33, 105 × 33)	92.38, 0.1554 (2 ⁻³ , 2 ³)	87.46, 0.0187 (2 ⁻⁷ , 2 ⁸ , 2 ⁰)	87.61, 0.0142 (2 ⁻⁵ , 2 ⁻⁴)	92.38, 0.0169 (2 ⁻³ , 0.8, 0.7, 2 ⁰)	96.19 , 0.0166 (2 ⁻⁵ , 2 ⁻³ , 0.8, 0.7, 2 ⁰)					
Bupa Liver (241 × 6, 104 × 6)	63.46, 0.2185 (2 ¹ , 2 ²)	70.19 , 0.0197 (2 ⁻² , 2 ⁴ , 2 ⁻²)	55.77, 0.0128 (2 ⁰ , 2 ⁻²)	66.35, 0.0188 (2 ⁻³ , 0.7, 0.6, 2 ¹)	69.23, 0.0188 (2 ⁻⁵ , 2 ⁻⁵ , 0.6, 0.6, 2 ⁻¹)					
Notes (306 × 16, 129 × 16)	96.90 , 0.1513 (2 ¹ , 2 ⁷)	96.90 , 0.0736 (2 ⁻⁶ , 2 ² , 2 ⁶)	96.12, 0.0224 (2 ⁻³ , 2 ⁸)	95.35, 0.0270 (2 ⁻⁵ , 0.6, 0.7, 2 ¹)	96.90 , 0.0258 (2 ⁻⁵ , 2 ⁻³ , 0.6, 0.8, 2 ⁰)					
WPBC (137 × 34, 57 × 34)	80.70 , 0.0685 (2 ⁻⁵ , 2 ²)	80.18, 0.0047 (2 ⁻⁴ , 2 ⁵ , 2 ⁵)	78.94, 0.0044 (2 ⁻⁵ , 2 ⁰)	75.44, 0.0063 (2 ⁻¹ , 0.9, 0.7, 2 ³)	80.70 , 0.0051 (2 ⁰ , 2 ¹ , 1, 0.7, 2 ⁻¹)					
Pima-Indians (537 × 8, 231 × 8)	77.48, 0.3717 (2 ⁻³ , 2 ⁰)	76.09, 0.0999 (2 ⁻³ , 2 ⁵ , 2 ⁻²)	64.07, 0.0929 (2 ⁻¹ , 2 ⁻¹)	78.35, 0.0989 (2 ⁻³ , 0.7, 0.7, 2 ³)	80.52 , 0.0979 (2 ⁰ , 2 ⁰ , 0.7, 0.6, 2 ⁻¹)					
German (800 × 24, 200 × 24)	62.50, 0.8049 (2 ⁻¹ , 2 ⁷)	69.50, 0.2146 (2 ⁻³ , 2 ⁷ , 2 ⁻³)	69.50, 0.2499 (2 ⁻³ , 2 ⁵)	78.00, 0.2884 (2 ⁻³ , 0.9, 0.8, 2 ⁵)	79.00 , 0.2611 (2 ⁻¹ , 2 ⁻¹ , 0.9, 0.7, 2 ⁴)					
Australian (540 × 14, 150 × 14)	77.33, 0.0797 (2 ⁻⁵ , 2 ⁴)	77.33, 0.0085 (2 ⁻¹ , 2 ³ , 2 ²)	56.67, 0.0069 (2 ⁻³ , 2 ¹)	76.00, 0.0083 (2 ⁻⁵ , 0.6, 0.8, 2 ⁴)	86.00 , 0.0079 (2 ³ , 2 ⁵ , 0.7, 0.6, 2 ⁻⁵)					
Haberman (200 × 3, 106 × 3)	75.47, 1.2747 (2 ⁻¹ , 2 ¹)	75.80, 0.2149 (2 ⁻¹ , 2 ³ , 2 ⁰)	77.35 , 0.2678 (2 ⁻¹ , 2 ⁴)	76.42, 0.3654 (2 ⁻³ , 0.8, 0.6, 2 ⁻²)	77.35 , 0.3579 (2 ⁻⁵ , 2 ⁻³ , 0.8, 0.6, 2 ⁻⁵)					
Transfusion (600 × 4, 148 × 4)	85.13, 0.2722 (2 ¹ , 2 ⁻¹)	76.42, 0.1117 (2 ⁻⁸ , 2 ⁶ , 2 ⁻³)	81.76, 0.1000 (2 ¹ , 2 ⁻³)	86.49, 0.1000 (2 ¹ , 0.7, 0.6, 2 ⁻³)	89.91 , 0.1004 (2 ³ , 2 ¹ , 0.6, 0.6, 2 ⁻⁵)					
WDBC (500 × 30, 69 × 30)	71.01, 0.1623 (2 ¹ , 2 ³)	79.71, 0.0085 (2 ⁻⁶ , 2 ⁵ , 2 ⁻²)	78.26, 0.0078 (2 ⁻⁵ , 2 ⁻⁴)	84.05, 0.0086 (2 ⁻³ , 0.7, 0.8, 2 ²)	85.51 , 0.0092 (2 ⁻³ , 2 ⁻⁵ , 0.8, 0.6, 2 ²)					
Splice (500 × 60, 2675 × 60)	86.02, 6.2216 (2 ⁻³ , 2 ⁶)	88.11, 0.1123 (2 ⁻⁴ , 2 ² , 2 ¹)	88.03, 0.1207 (2 ⁻⁵ , 2 ²)	88.00, 0.1273 (2 ⁻⁵ , 0.8, 0.6, 2 ²)	88.48 , 0.1281 (2 ⁻³ , 2 ⁻⁵ , 0.8, 0.6, 2 ²)					
CMC (1000 × 9, 473 × 9)	72.52, 0.1027 (2 ⁻³ , 2 ¹⁰)	73.95, 0.0247 (2 ⁻³ , 2 ³ , 2 ⁰)	74.63, 0.0098 (2 ⁵ , 2 ⁻²)	74.20, 0.0146 (2 ⁴ , 0.7, 0.7, 2 ⁰)	75.47 , 0.0107 (2 ⁻¹ , 2 ³ , 0.6, 0.6, 2 ⁰)					

Bold type shows the best result

our RELS-TSVM. We focus on the comparisons between our RELS-TSVM and some related classifiers, including ELS-TSVM [23], LSPTSVM [29], LSTSVM [14], TBSVM [28] and TSVM [12]. All the experiments are carried out in MATLAB R2010a on a PC with 2.27 GHz Intel(R) Xeon(R) processor and 3 GB of RAM. In the case of a non-linear kernel, the Gaussian kernel function is employed as it is often employed and yields great generalization performance. Then classification accuracy of each algorithm is computed using ten-fold cross-validation [7].

4.1 Parameters selection

An important problem is the parameter selection of these six algorithms. The classification performance depends heavily on the choices made regarding the kernel functions and its parameter values. We choose optimal values of the parameters by the grid search method [10]. For the six algorithms, the Gaussian kernel parameter μ is selected from the sets $\{2^i | i = -10, -9, \dots, 10\}$. The value of the parameters $c_{(1,2,3,4)}$ and $E_{(1,2)}$ for both linear as well as nonlinear kernel were selected from the sets $\{2^i | i = -5, -3, -1, 0, 1, 3, 5\}$ and $\{0.6, 0.7, 0.8, 0.9, 1\}$ respectively. To decrease the computational cost of the parameter selection, we set $c_1 = c_2$ for TSVM, LSTSVM and ELS-TSVM and $c_1 = c_2, c_3 = c_4$ for TBSVM, LSPTSVM and our RELS-TSVM.

4.2 Experimental results and discussion

The experimental results on several synthetic and benchmark datasets for linear and Gaussian kernel are summarized in Tables 1 and 2, respectively, where ‘‘Acc.’’ denotes the mean value of ten times testing results, and ‘‘Time’’ denotes the training time.

4.2.1 Synthetic datasets

We consider a simple two dimensional ‘‘Cross Planes’’ dataset as an example of synthetic dataset which was also

tested in [14, 20, 28, 32]. It was generated by perturbing points lying on two intersecting lines and the intersection point is not in the center. It is easy to see from Table 1 that the result of the proposed RELS-TSVM is more reasonable than that of ELS-TSVM. This clearly indicates that our RELS-TSVM can handle the ‘‘Cross Planes’’ dataset much better than ELS-TSVM. The second example is an artificially-generated Ripley’s synthetic dataset [26]. It is a two dimensional dataset which includes 250 patterns. One observes from Tables 1 and 2 that our RELS-TSVM obtains better classification performance with less training time than other algorithms.

4.2.2 UCI datasets

We performed numerical experiments to demonstrate the performance of our RELS-TSVM in comparison to TSVM, TBSVM, LSTSVM, LSPTSVM and ELS-TSVM on several publicly available benchmark datasets [22]. In all the real-world examples considered, each attribute of the original data is normalized as follows:

$$\bar{x}_{ij} = \frac{x_{ij} - x_j^{\min}}{x_j^{\max} - x_j^{\min}},$$

where x_{ij} is the (i,j)-th element of the input matrix A , \bar{x}_{ij} is its corresponding normalized value. $x_j^{\min} = \min_{i=1}^m(x_{ij})$ and $x_j^{\max} = \max_{i=1}^m(x_{ij})$ denote the minimum and maximum values, respectively, of the j-th column of A . Clearly, one observes from Table 1 that, in comparison to TSVM, TBSVM, LSTSVM, LSPTSVM and ELS-TSVM, our RELS-TSVM shows better generalization performance. For Heart-c dataset, our RELS-TSVM (accuracy 70.83 % time 0.0028 s) outperforms other five algorithms either in terms of time, accuracy or both i.e., TSVM (accuracy 69.17 % time 0.1530 s), TBSVM (accuracy 71.67 % time 0.0032 s), LSTSVM (accuracy 67.50 % 0.0013 s), LSPTSVM (accuracy 57.50 % time 0.010 s), and ELS-TSVM (accuracy 65.00 % 0.0035 s). In the same fashion, for Ionosphere dataset, the experimental

Table 3 Performance comparison of RELS-TSVM with ELS-TSVM

Datasets (Train size, Test size)	ELS-TSVM Acc.(%) ($c_1=c_2, E_1, E_2$)	RELS-TSVM Acc.(%) ($c_1=c_2, c_3=c_4, E_1, E_2$)
BCI Ia (268 × 5376, 293 × 5376)	66.89 ($2^3, 0.6, 1.0$)	81.11 ($2^4, 2^4, 0.7, 0.6$)
BCI Ib (200 × 8064, 180 × 8064)	48.89 ($2^3, 0.6, 0.8$)	53.41 ($2^3, 2^5, 0.6, 0.8$)

Bold type shows the best result

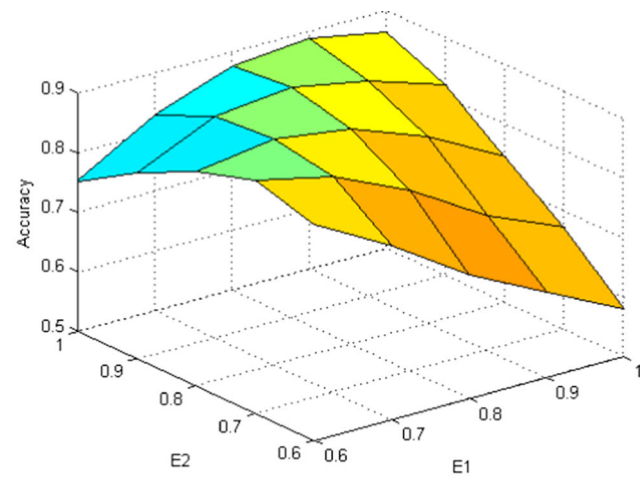
results by our RELS-TSVM (88.87 %) are higher than TSVM (84.76 %), TBSVM (85.71 %), LSTSVM (85.71 %), LSPTSVM (88.57 %) and ELS-TSVM (85.71 %). We obtained the similar conclusions for Heart-statlog, WPBC, Pima, Cleve, German, Australian, Transfusion and Sonar datasets.

Furthermore, one observes from Table 2 that our RELS-TSVM outperforms TSVM, TBSVM, LSTSVM and ELS-TSVM using Gaussian kernel. For Votes dataset, one sees from Table 2, the classification accuracies obtained by our RELS-TSVM is higher than LSTSVM, ELS-TSVM and is equal to TSVM and TBSVM but time complexity is still lower in our RELS-TSVM than TSVM and TBSVM. The empirical results further reveal that our proposed RELS-TSVM whose solutions are obtained by solving system of linear equations, is faster and more robust than others on most of the datasets. This clearly ranks our approach higher and more preferable. It is worthwhile to notice that the values of the parameters c_3 and c_4 affect the results significantly and these values are varying in our RELS-TSVM rather than small fixed positive scalar in TSVM, TBSVM, LSTSVM and ELS-TSVM.

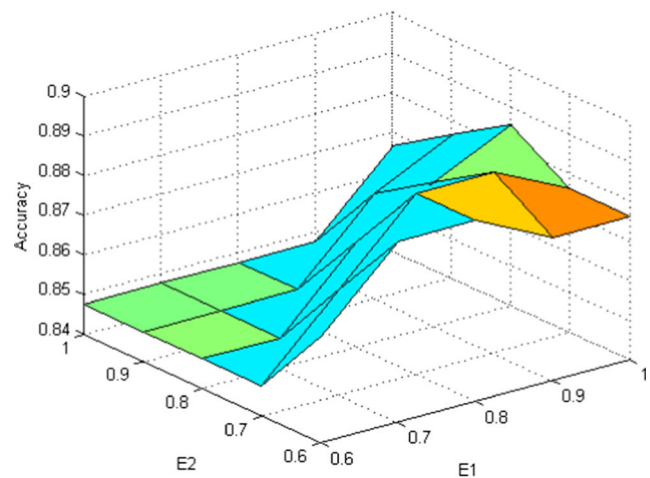
4.2.3 Applications

In this subsection, we check the integrity of RELS-TSVM on BCI competition II (Ia, Ib) [16] datasets. The goal of BCI competition II is to validate signal processing and classification methods for Brain Computer Interfaces (BCIs). The features in dataset Ia were taken from healthy person and in BCI-Ib from an artificially respirated ALS patient. BCI Ia training data consist of 268 trials recorded on two different days mixed randomly. 168 of the overall 268 trials origin from day 1, the remaining 100 trials from day 2. The training data matrix contains data of 135 trials belonging to class 0 and 133 trials belonging to class 1. The data matrix dimension is 268×5376 . Every line of a matrix contains the data of one trial, belonging to either of the two classes. The data values are the time samples of the 6 EEG channels. This starts with 896 samples from channel 1 and ends with 896 samples from channel 6. The test matrix contains 293 trials of test data with dimension 293×5376 . Every trial belongs to either class 0 or class 1. Similar to training data, every line contains 6 times 896 samples.

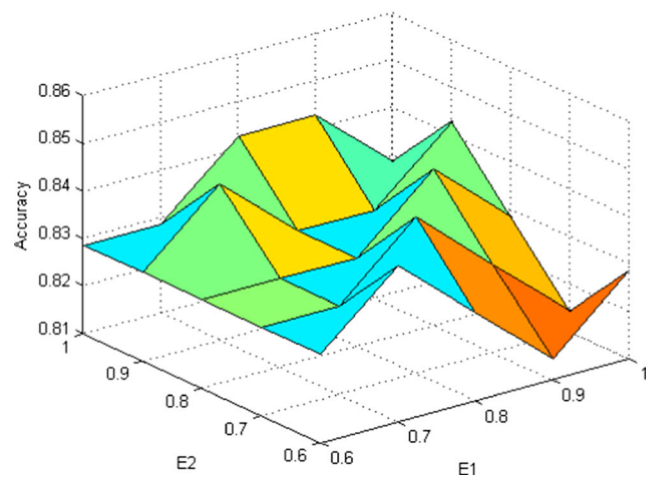
The BCI-1b training dataset contains 200 trials, where each class has 100 trials belonging to the corresponding class. The data matrix dimension is 200×8064 . Every line of a matrix contains the data of one trial, where each trial contains the time samples of the 7 EEG/EOG channels. This starts with 1152 samples from channel 1 and ends with 1152 samples from channel 7. The test data matrix contains 180 trials, where the dimension of matrix is 180×8064 . The 180 trials belong to either class 0 or class 1. Similar to training



(a) Ripley



(b) Ionosphere



(c) Heart-statlog

Fig. 1 Performance of RELS-TSVM on energy parameters (E_1 , E_2) for three classification datasets

data, every line contains 7 times 1152 samples. For further details, the interested reader is referred to [16]. The experiment is performed using 10-fold cross-validation to select the optimal parameters. One observes from Table 3, that our RELS-TSVM outperforms ELS-TSVM on both BCI-Ia and BCI-Ib datasets. The above results indicate that our RELS-TSVM performs better on large neural signal processing datasets.

4.2.4 Influence of energy parameters E_1 and E_2

We conduct experiments on Ripley, Ionosphere and Heart-statlog datasets to investigate the influence of the energy parameters E_1 and E_2 on the performance of our RELS-TSVM. The constraints of LSTSVM require hyperplane to be at a distance of unity from points of the other class. This makes it more sensitive to outliers. However, our RELS-TSVM introduces an energy term for each hyperplane E_1 and E_2 , appropriate value of these terms reduce the sensitivity of the classifier towards noise, thus making it more effective and robust. The energy parameters E_1 and E_2 are selected using grid search from the range [0.6, 0.7, 0.8, 0.9, 1.0]. If the certainty of a sample to be classified in any of the two classes is equal then the proportion E_1/E_2 will be equal to one. If the proportion E_1/E_2 is large then the sample will have higher certainty to be of class 1 than of class 2 and vice versa. We set the parameters c_1, c_2, c_3 and c_4 as ($c_1 = c_2; c_3 = c_4$) in Table 1 to capture the effect of energy parameters. Figure 1 shows the variation of performance with energy parameters E_1 and E_2 on Ripley, Ionosphere and Heart-statlog datasets. It can be observed from Fig. 1a that the

performance of Ripley dataset is better for higher value of E_1 and lower value of E_2 which indicate that our classifier adjusts accordingly to reduce sensitivity of samples to be misclassified. Figure 1b shows that RELS-TSVM improves the performance on comparable values of E_1 and E_2 . Similarly, Fig. 1c shows better performance on Heart-statlog dataset for higher value of E_1 and lower value of E_2 . One observes that the performance of the proposed RELS-TSVM fluctuates when E_1 and E_2 varies. This fluctuation shows how hyperplane adjusts and finally at an optimum energy value how it tends to be more effective towards noise.

4.2.5 Statistical Analysis

To verify the statistical significance of our RELS-TSVM in comparison to TSVM, TBSVM, LSTSVM, LSPTSVM and ELS-TSVM, we use the Friedman test. This test with the corresponding post hoc tests is pointed out to be a simple, safe, and robust non parametric test for comparison of more classifiers over multiple datasets [6]. We use it to compare the performance of six algorithms. The average ranks of all the algorithms on accuracies with linear kernel were computed and listed in Table 4. We employ the Friedman test to check whether the measured average ranks are significantly different from the mean rank $R_j = 3.5$. Under the null hypothesis, the Friedman statistic

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_{j=1}^4 R_j^2 - \frac{k(k+1)^2}{4} \right]$$

Table 4 Average ranks of TSVM, TBSVM, LSTSVM, LSPTSVM, ELS-TSVM and our RELS-TSVM with linear kernel on accuracies

Datasets	TSVM	TBSVM	LSTSVM	LSPTSVM	ELS-TSVM	RELS-TSVM
Cross Planes	3.5	1.5	6	1.5	5	3.5
Ripley	3	4	2	5	6	1
Heart-c	3	1	4	6	5	2
Ionosphere	6	4	4	2	4	1
Heart-statlog	3	6	3	3	5	1
Bupa Liver	4.5	1	4.5	2	6	3
WPBC	3	1.5	6	5	4	1.5
Pima-Indians	4	4	6	1.5	4	1.5
Cleve	4	5.5	3	2	5.5	1
German	5	3.5	1.5	6	3.5	1.5
Australian	5	5	2	5	2	2
Transfusion	6	2	3.5	5	3.5	1
Sonar	5	2	6	4	3	1
Average rank	4.23	3.15	3.96	3.69	4.35	1.62

Table 5 Average ranks of TSVM, TBSVM, LSTSVM, ELS-TSVM and our RELS-TSVM with Gaussian kernel on accuracies

Datasets	TSVM	TBSVM	LSTSVM	ELS-TSVM	RELS-TSVM
Ripley	2	4.5	3	4.5	1
Heart-c	5	1	3.5	3.5	2
Heart-stat	4	3	1.5	5	1.5
Ionosphere	2.5	5	4	2.5	1
Bupa Liver	4	1	5	3	2
Votes	2	2	4	5	2
WPBC	1.5	3	4	5	1.5
Pima-Indian	3	4	5	2	1
German	5	3.5	3.5	2	1
Australian	2.5	2.5	5	4	1
Haberman	5	4	1.5	3	1.5
Transfusion	3	5	4	2	1
WDBC	5	3	4	2	1
Splice	4	2	3	5	1
CMC	5	4	2	3	1
Average rank	3.57	3.53	3.43	3.16	1.30

is distributed according to χ_F^2 with $k-1$ degrees of freedom, where k is the number of methods and N is the number of datasets.

$$\chi_F^2 = \frac{12 \times 13}{6(6+1)} \left[4.23^2 + 3.15^2 + 3.96^2 + 3.69^2 + 4.35^2 + 1.62^2 - \frac{6(7)^2}{4} \right] = 19.16.$$

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1) - \chi_F^2} = \frac{(13-1) \times 19.16}{13(6-1) - 19.16} = 5.015.$$

With six algorithms and thirteen datasets, F_F is distributed according to the F -distribution with $(k-1)$ and $(k-1)(N-1) = (5, 60)$ degrees of freedom. The critical value of $F(5, 60)$ for $\alpha = 0.05$ is 2.37. Since the value of F_F is larger than the critical value, we reject the null hypothesis. For further pairwise comparison, we use the Nemenyi test. At $p = 0.10$, the critical difference (CD) = $2.589\sqrt{\frac{6 \times 7}{6 \times 13}} = 1.89$. Since the difference between ELS-TSVM and our RELS-TSVM is larger than the critical difference ($4.35 - 1.62 = 2.73 > 1.89$), we conclude that the generalization performance of RELS-TSVM is superior to ELS-TSVM. In the same way, we conclude that our RELS-TSVM is significantly better than LSPTSVM, LSTSVM and TSVM. Next, we see that the difference between TBSVM and our RELS-TSVM is slightly smaller than the critical difference ($3.15 - 1.62 = 1.53 < 1.89$), we conclude that the posthoc test is not powerful enough to detect any significant difference between TBSVM and our RELS-TSVM.

For further comparisons, we check the performance of five algorithms statistically on accuracies with Gaussian

kernel. The average ranks of all the algorithms on accuracies were computed and listed in Table 5. Under the null hypothesis, Friedman statistic will be

$$\chi_F^2 = \frac{12 \times 15}{5(5+1)} \left[3.57^2 + 3.53^2 + 3.43^2 + 3.16^2 + 1.30^2 - \frac{5(6)^2}{4} \right] = 21.88.$$

$$F_F = \frac{(15-1) \times 21.88}{15(5-1) - 21.88} = 8.035.$$

With five algorithms and fifteen datasets, F_F is distributed according to the F -distribution with $(k-1)$ and $(k-1)(N-1) = (4, 56)$ degrees of freedom. The critical value of $F(4, 56)$ for $\alpha = 0.05$ is 2.53. Since the value of F_F is larger than the critical value, so we reject the null hypothesis. For further pairwise comparison, we use the Nemenyi test. At $p = 0.10$, the critical difference (CD) = $2.459\sqrt{\frac{5 \times 6}{6 \times 15}} = 1.42$. Since the difference between TSVM, TBSVM, LSTSVM, ELS-TSVM and our RELS-TSVM is larger than the critical difference, we conclude that the generalization performance of RELS-TSVM is superior to ELS-TSVM. Since the value of F_F is larger than the critical value, we reject the null hypothesis. By the posthoc test, one concludes that the performance of RELS-TSVM is significantly better than ELS-TSVM, LSTSVM, TBSVM and TSVM.

5 Conclusions

In this paper, we propose an improved version of ELS-TSVM based on LSTSVM. Different from LSTSVM and

ELS-TSVM, we add an extra regularization term to maximize the margin, ensuring the optimization problems in our RELS-TSVM are positive definite and implements the structural risk minimization principle which embodies the marrow of statistical learning theory. Two parameters c_3 and c_4 introduced in our RELS-TSVM are the weights between the regularization term and the empirical risk, so that they can be chosen flexibly, improving the ELS-TSVM and LSTSVM. Unlike LSTSVM, our RELS-TSVM introduce energy parameters to reduce the effect of noise and outliers. The superiority of our RELS-TSVM is demonstrated on several synthetic and real-world benchmark datasets showing better classification ability with less training time in comparison to ELS-TSVM, LSPTSVM, LSTSVM, TBSVM and TSVM. There are seven parameters in our RELS-TSVM, so the parameter selection is a practical problem and will need to address in future.

Acknowledgments The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

- Balasundaram S, Tanveer M (2013) On Lagrangian twin support vector regression. *Neural Comput & Applic* 22(1):257–267
- Burges CJC (1998) A tutorial on support vector machines for pattern recognition. *Data Min Knowl Disc* 2:1–43
- Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol (TIST)* 2(3):27
- Cortes C, Vapnik VN (1995) Support vector networks. *Mach Learn* 20:273–297
- Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel based learning method. Cambridge University Press, Cambridge
- Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
- Duda RO, Hart PR, Stork DG (2001) *Pattern Classification*, 2nd. John Wiley and Sons
- Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. In: *Proceedings of 7th international conference on knowledge and data discovery, San Fransisco*, pp 77–86
- Golub GH (2012) *C.F.V. Loan, Matrix Computations*, vol 3. JHU Press
- Hsu CW, Lin CJ (2002) A comparison of methods for multi-class support vector machines. *IEEE Trans Neural Networks* 13:415–425
- Hua X, Ding S (2015) Weighted least squares projection twin support vector machines with local information. *Neurocomputing* 160:228–237. doi:10.1016/j.neucom.2015.02.021
- Jayadeva, Khemchandani R, Chandra S (2007) Twin support vector machines for pattern classification. *IEEE Trans Pattern Anal Mach Intell* 29(5):905–910
- Joachims T (1999) Making large-scale support vector machine learning practical, *Advances in Kernel Methods*. MIT Press, Cambridge
- Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. *Expert Systems with Applications* 36:7535–7543
- Kumar MA, Khemchandani R, Gopal M, Chandra S (2010) Knowledge based least squares twin support vector machines. *Inf Sci* 180(23):4606–4618
- Lal TN, Hinterberger T, Widman G, Schrder M, Hill J, Rosenstiel W, Elger C, Schlkopf B, Birbaumer N (2004) Methods towards invasive human brain computer interfaces. *Advances in Neural Information Processing Systems (NIPS)*
- Lee YJ, Mangasarian OL (2001a) RSVM: Reduced support vector machines. In: *Proceedings of the 1st SIAM international conference on data mining*, pp 5–7
- Lee YJ, Mangasarian OL (2001b) SSVN: A smooth support vector machine for classification. *Comput Optim Appl* 20(1): 5–22
- Mangasarian OL, Musicant DR (2001) Lagrangian support vector machines. *J Mach Learn Res* 1:161–177
- Mangasarian OL, Wild EW (2006) Multisurface proximal support vector classification via generalized eigenvalues. *IEEE Trans Pattern Anal Mach Intell* 28(1):69–74
- Mehrkanoon S, Huang X, Suykens JAK (2014) Non-parallel support vector classifiers with different loss functions. *Neurocomputing* 143(2):294–301
- Murphy PM, Aha DW (1992) UCI repository of machine learning databases. University of California, Irvine. <http://www.ics.uci.edu/~mllearn>
- Nasiri JA, Charkari NM, Mozafari K (2014) Energy-based model of least squares twin support vector machines for human action recognition. *Signal Process* 104:248–257
- Peng X (2010) TSVR: An efficient twin support vector machine for regression. *Neural Netw* 23(3):365–372
- Platt J (1999) Fast training of support vector machines using sequential minimal optimization. In: Scholkopf B, Burges CJC, Smola AJ (eds) *Advances in Kernel Methods-Support Vector Learning*. MIT Press, Cambridge, MA, pp 185–208
- Ripley BD (2007) *Pattern recognition and neural networks*, Cambridge University Press
- Tanveer M (2015) Robust and sparse linear programming twin support vector machines. *Cogn Comput* 7:137–149
- Shao YH, Zhang CH, Wang XB, Deng NY (2011) Improvements on twin support vector machines. *IEEE Trans Neural Networks* 22(6):962–968
- Shao YH, Deng NY, Yang ZM (2012) Least squares recursive projection twin support vector machine for classification. *Pattern Recognit* 45(6):2299–2307
- Shao YH, Chen WJ, Wang Z, Li CN, Deng NY (2014) Weighted linear loss twin support vector machine for large scale classification. *Knowl-Based Syst* 73:276–288
- Tanveer M, Mangal M, Ahmad I, Shao YH (2016) One norm linear programming support vector regression. *Neurocomputing* 173:1508–1518. doi:10.1016/j.neucom.2015.09.024
- Tanveer M (2015) Application of smoothing techniques for linear programming twin support vector machines. *Knowl Inf Syst* 45(1):191–214. doi:10.1007/s10115-014-0786-3
- Tian Y, Ping Y (2014) Large-scale linear nonparallel support vector machine solver. *Neural Netw* 50:166–174
- Vapnik VN (1998) *Statistical Learning Theory*. Wiley, New York
- Vapnik VN (2000) *The nature of statistical learning theory* 2nd Edition. Springer, New York
- Ye Q, Zhao C, Ye N (2012) Least squares twin support vector machine classification via maximum one-class within class variance. *Optimization methods and software* 27(1): 53–69
- Zhang Z, Zhen L, Deng NY (2014) Sparse least square twin support vector machine with adaptive norm. *Appl Intell* 41(4):1097–1107



Mohammad Tanveer is currently working as Postdoctoral Research Fellow at School of Computer Engineering, NTU Singapore. He received his B.Sc. (Hons), M.Sc. and M.Phil. degrees in Mathematics from Aligarh Muslim University, Aligarh, INDIA, in 2003, 2005 and 2007 respectively, and Ph.D. degree in Computer Science from Jawaharlal Nehru University, New Delhi, INDIA, in 2013. He has been an Assistant Professor of Computer Science

and Engineering at The LNM Institute of Information Technology, Jaipur, INDIA. His research interests include optimization methods, machine learning and fixed point theory and applications. He has published over 20 refereed journal papers of international repute.



Mohammad Asif Khan is a final year student pursuing his B.Tech. in Electronics and Communication engineering from The LNM Institute of Information Technology, Jaipur, India. His current research interests spans over signal processing, machine learning, deep learning and artificial intelligence.



Shen-Shyang Ho received the B.S. degree in mathematics and computational science from the National University of Singapore, Singapore, in 1999, and the M.S. and Ph.D. degrees in computer science from George Mason University, Fairfax, VA, USA, in 2003 and 2007, respectively.

He was a NASA Post-Doctoral Program Fellow and then a Post-Doctoral Scholar with the California Institute of Technology, Pasadena, CA, USA, affiliated to the Jet

Propulsion Laboratory, Pasadena, from 2007 to 2010. From 2010 to 2012, he was a Researcher involved in projects funded by NASA at the University of Maryland Institute for Advanced Computer Studies, College Park, MD, USA. He is currently a Tenure-Track Assistant Professor with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. His current research interests include data mining, machine learning, pattern recognition in spatiotemporal/data streaming settings, array-based databases, and privacy issues in data mining.

Dr. Ho has given the tutorial titled “Conformal Predictions for Reliable Machine Learning” at AAAI Conference on Artificial Intelligence, the International Joint Conference on Neural Networks (IJCNN), and the European Conference on Machine Learning (ECML). His current research projects are funded by BMW, Rolls-Royce, NTU, and the Ministry of Education in Singapore.