Neural Processing Letters (2020) 51:41–66
https://doi.org/10.1007/s11063-019-10078-w

Entropy-Based Fuzzy Least Squares Twin Support Vector Machine for Pattern Classification

Sugen Chen[1,2] · Junfeng Cao[3] · Fenglin Chen[1] · Bingbing Liu[1]

Published online: 11 July 2019

Abstract

Least squares twin support vector machine (LSTSVM) is a new machine learning method, as opposed to solving two quadratic programming problems in twin support vector machine (TWSVM), which generates two nonparallel hyperplanes by solving a pair of linear system of equations. However, LSTSVM obtains the resultant classifier by giving same importance to all training samples which may be important for classification performance. In this paper, by considering the fuzzy membership value for each sample, we propose an entropy-based fuzzy least squares twin support vector machine where fuzzy membership values are assigned based on the entropy values of all training samples. The proposed method not only retains the superior characteristics of LSTSVM which is simple and fast algorithm, but also implements the structural risk minimization principle to overcome the possible over- fitting problem. Experiments are performed on several synthetic as well as benchmark datasets and the experimental results illustrate the effectiveness of our method.

Keywords Pattern classification · Information entropy · Least squares twin support vector machine · Fuzzy membership

✉ Sugen Chen
chensugen@126.com

Junfeng Cao
caojunfeng982@163.com

Fenglin Chen
chenfenglin@aqnu.edu.cn

Bingbing Liu
lbb122400@gmail.com

1 School of Mathematics and Computational Science, Anqing Normal University, Anqing 246133, Anhui, People's Republic of China

2 Key Laboratory of Modeling, Simulation and Control of Complex Ecosystem in Dabie Mountains of Anhui Higher Education Institutes, Anqing Normal University, Anqing 246133, China

3 School of Science, Jiangnan University, Wuxi 214122, Jiangsu, People's Republic of China

1 Introduction

Support vector machine (SVM) [1] is one of the most popular machine learning approach, which has already been successfully applied to a variety of real-world problems such as face recognition [2], bioinformatics [3], text categorization [4], intrusion detection [5] and various other classification problems [6, 7]. However, the training cost of SVM is very high, i.e. $O(m^3)$, where m is the number of training samples. Researchers have made many improvements on the basis of SVM, such as Fung and Mangasrian [8] proposed proximal support vector machine (PSVM) for binary classification. Recently, following PSVM, Mangasrian and Wild [9] proposed multi-surface proximal SVM via generalized eigenvalues (GEPSVM) for binary classification, which aims at seeking two nonparallel proximal hyperplanes such that each hyperplane is closer to one of two classes and as far as possible from the other. Inspired by GEPSVM, Jayadeva et al. [10] proposed another nonparallel hyperplane classifier for pattern classification, called twin support vector machine (TWSVM). The main idea of TWSVM is to solve two smaller quadratic programming problems (QPPs) rather than a single large QPP which makes training speed of TWSVM four times faster than SVM. From then on, TWSVM has been widely investigated [11–16]. Some improvements have been made to TWSVM by researchers to obtain higher classification accuracy with lower computational time, such as Least squares twin support vector machine (LSTSVM) [11], Twin bounded support vector machine (TBSVM) [12], Twin parametric-margin support vector machine (TPMSVM) [13], Robust twin support vector machine (RTSVM) [14], Nonparallel support vector machines (NPSVM) [15] and Angle-based twin support vector machine (ATSVM) [16], and so on [17–20].

It should be noted, in practical problems, some data are often polluted by noise or in low quality. The patterns, even belong to the same class, should play different roles in the model training. Since SVM treats all samples with the same importance, it ignores the differences between the positive and negative classes, which results in the learned decision surface biasing toward the majority class. To address this problem, based on fuzzy membership values, Lin et al. [21] propose the Fuzzy SVM (FSVM) such that different input samples have different contributions to the learning of decision surface. However, it assigns smaller fuzzy memberships to support vectors which might decrease the effects of support vectors on the construction of decision surface. In order to overcome this problem in FSVM, a new efficient approach fuzzy SVM for non-equilibrium data is proposed to reduce the misclassification accuracy of minority class in FSVM [22]. Adopting fuzzy membership, various Fuzzy SVMs are presented such as Bilateral-weighted FSVM (B-FSVM) [23], NFSVM [24], WCS-FSVM [25], FTSVM [26] and NFTSVM [27]. Moreover, the fuzzy set and fuzzy system theory is also widely used in various control problems [28–30].

Recently, Fan et al. [31] proposed an entropy-based fuzzy support vector machine (EFSVM) for class imbalance problem in which fuzzy membership is computed based on the class certainty of samples. Motivated by EFSVM, Gupta et al. [32] proposed a fuzzy twin support vector machine based on information entropy which is termed as EFTWSVM-CIL. Therefore, the choice of fuzzy membership is very important for classification problems. Each sample is given a fuzzy membership which indicates the importance of the corresponding sample toward one class and this change can make some contribution to the final decision surface. Based on the above discussion and inspired by EFTWSVM-CIL and LSTSVM, in this paper, we propose a new approach termed as entropy-based fuzzy least squares twin support vector machine (EFLSTSVM). The contributions of the proposed EFLSTSVM can be highlighted as follows. First, entropy-based fuzzy membership is given to evaluate the class certainty of training samples. Similar to EFSVM and EFTWSVM-CIL, our EFLSTSVM adopts the entropy to evaluate the class certainty of each sample and then determines the corresponding fuzzy membership based on the class certainty. Thus, it can pay more attention to the samples with higher class certainty to result in more robust decision surface, which enhances the classification accuracy and generalization ability. Second, we modify the QPP-based formulation of EFTWSVM-CIL in least squares sense which leads to solving the optimization problem with equality constraints. Different from EFTWSVM-CIL, our EFLSTSVM solves a pair of linear system of equations as opposed to solving two QPPs in EFTWSVM-CIL, which leads to simple algorithm and less computational time. Third, a regularization term in the objective function is introduced. Therefore, our EFLSTSVM implements the structural risk minimization principle instead of the empirical risk minimization principle, which can overcome the possible over-fitting problems in LSTSVM. And last but not least, the experimental results on several synthetic datasets and benchmark datasets show the effectiveness of our proposed EFLSTSVM.

The rest of this paper is organized as follows. In Sect. 2, we give a brief review of LSTSVM and FTSVM. Section 3 proposes the details of linear EFLSTSVM and its nonlinear version. Experimental results on both synthetic and real-world datasets to investigate the effectiveness of our method are described in Sect. 4. Finally, Sect. 5 gives the conclusion.

2 Brief Review of LSTSVM and FTSVM

In this section, we briefly explain the basics of LSTSVM and FTSVM. Let us consider a binary classification problem in the n-dimensional real space R^n and a set of training data samples is represented by $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, where $x_i \in R^n$ and $y_i \in \{-1, 1\}, i = 1, 2, \dots, m$. We organize the m_1 samples of positive class by a $m_1 \times n$ matrix $A \in R^{m_1 \times n}$ and m_2 samples of negative class by a $m_2 \times n$ matrix $B \in R^{m_2 \times n}$.

2.1 Least Squares Twin Support Vector Machine (LSTSVM)

Different from least squares support vector machine (LSSVM) [33], least squares twin support vector machine (LSTSVM) [11] aims to find a pair of nonparallel hyperplanes

$$w_1^T x + b_1 = 0 \quad \text{and} \quad w_2^T x + b_2 = 0 \tag{1}$$

such that each hyperplane is close to the training samples of one class and as far as possible from the samples of the other class. Then, the primal optimization problem of linear LSTSVM can be expressed as

$$\begin{aligned} &\min_{w_1, b_1, \xi_2} \ \frac{1}{2}||Aw_1 + e_1 b_1||_2^2 + \frac{c_1}{2}\xi_2^T \xi_2 \\ &s.t. \quad -(Bw_1 + e_2 b_1) + \xi_2 = e_2 \end{aligned} \tag{2}$$

$$\begin{aligned} &\min_{w_2, b_2, \xi_1} \ \frac{1}{2}||Bw_2 + e_2 b_2||_2^2 + \frac{c_2}{2}\xi_1^T \xi_1 \\ &s.t. \quad (Aw_2 + e_1 b_2) + \xi_1 = e_1 \end{aligned} \tag{3}$$

where c_1 and c_2 are positive penalty parameters, ξ_1 and ξ_2 are slack variables, e_1 and e_2 are vectors with each element of the value of 1.

By substituting the equality constraint into the objective function, we obtain the unconstrained optimization problem as follows.

$$[A\ e_1]^T[A\ e_1][w_1^T\ b_1]^T + c_1[B\ e_2]^T[B\ e_2][w_1^T\ b_1]^T + c_1[B\ e_2]^T e_2 = 0 \tag{4}$$

$$[B\ e_2]^T[B\ e_2][w_2^T\ b_2]^T + c_2[A\ e_1]^T[A\ e_1][w_2^T\ b_2]^T - c_2[A\ e_1]^T e_1 = 0 \tag{5}$$

Let E = $[A\ e_1]$, F= $[B\ e_2]$, form (4) and (5), we can get

$$[w_1^T\ b_1]^T = -\left(F^T F + \frac{1}{c_1} E^T E\right)^{-1} F^T e_2 \tag{6}$$

$$[w_2^T\ b_2]^T = \left(E^T E + \frac{1}{c_2} F^T F\right)^{-1} E^T e_1 \tag{7}$$

The solutions to the optimization problems (2) and (3) can be found directly by solving systems of linear Eqs. (4) and (5), more details can be seen in [11]. Once w_1, b_1 and w_2, b_2 are obtained from (6) and (7), the nonparallel hyperplanes (1) are known. A new data point $x \in R^n$ is then assigned to positive class W_1 or negative class W_2 by

$$x \in W_k,\ k = \arg\min_{k=1,2}\{|w_1^T x + b_1|, |w_2^T x + b_2|\} \tag{8}$$

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

2.2 Fuzzy Twin Support Vector Machine (FTSVM)

Different from TWSVM [10], in the case of linear FTSVM [26], a weighting parameter is used to construct the classifier based on fuzzy membership values. The formulation of linear FTSVM can be written as

$$\begin{aligned} \min_{w_1,b_1,\xi_2}\ & \frac{1}{2}||Aw_1 + e_1 b_1||_2^2 + c_1 s_2^T \xi_2 \\ s.t.\ & -(Bw_1 + e_2 b_1) + \xi_2 \ge e_2, \xi_2 \ge 0 \end{aligned} \tag{9}$$

$$\begin{aligned} \min_{w_2,b_2,\xi_1}\ & \frac{1}{2}||Bw_2 + e_2 b_2||_2^2 + c_2 s_1^T \xi_1 \\ s.t.\ & (Aw_2 + e_1 b_2) + \xi_1 \ge e_1, \xi_1 \ge 0 \end{aligned} \tag{10}$$

where c_1 and c_2 are positive penalty parameters, ξ_1 and ξ_2 are slack variables, e_1 and e_2 are vectors with each element of the value of 1, s_1 and s_2 represent fuzzy membership of each type of sample points.

By introducing the method of Lagrangian multipliers, the corresponding Wolfe dual of QPPs (9) and (10) can be represented as

$$\begin{aligned} \max_{\alpha}\ & e_2^T \alpha - \frac{1}{2}\alpha^T G(H^T H)^{-1} G^T \alpha \\ s.t.\ & 0 \le \alpha \le c_1 s_2 \end{aligned} \tag{11}$$

$$\begin{aligned} \max_{\gamma} \;& e_1^T\gamma - \frac{1}{2}\gamma^T H(G^T G)^{-1} H^T \gamma \\ s.t. \;& 0 \le \gamma \le c_2 s_1 \end{aligned} \tag{12}$$

where $G = [B\ e_2],\ H = [A\ e_1]$ and $\alpha \in R^{m_2},\ \gamma \in R^{m_1}$ are Lagrangian multipliers.

The nonparallel hyperplanes (1) can be obtained from the solutions α and γ of (11) and (12) by

$$[w_1^T\ b_1]^T = -(H^T H)^{-1} G^T \alpha \tag{13}$$

$$[w_2^T\ b_2]^T = (G^T G)^{-1} H^T \gamma \tag{14}$$

Once w_1, b_1 and w_2, b_2 are obtained from (13) and (14), the two nonparallel hyperplanes (1) are known. A new data point $x \in R^n$ is then assigned to positive class W_1 or negative class W_2 by

$$x \in W_k,\ k = \arg\min_{k=1,2}\{|w_1^T x + b_1|, |w_2^T x + b_2|\} \tag{15}$$

where $|\cdot|$ is the absolute value. More details about FTSVM can be seen in [26].

3 Entropy-Based Fuzzy Least Squares Twin Support Vector Machine

As the evaluation of fuzzy membership is the key issue of FSVM, in this section, we introduce the entropy-based fuzzy membership at first. Then, by adopting the entropy-based fuzzy membership, the entropy-based fuzzy least squares twin support vector machine (EFLSTSVM) for binary classification is presented.

3.1 Entropy-Based Fuzzy Membership

In information theory, entropy is measure of the information carried by a sample [34]. Thus, entropy characterizes the certainty about the source of information, that is, the smaller entropy indicates the information is more certain. By adopting entropy, we can evaluate the class certainty of training samples and assign the fuzzy membership of the training samples based on their class certainty. Specifically, the sample with higher class certainty will be assigned to larger fuzzy memberships to enhance their contribution to the decision surface, and vice versa. Supposing the probabilities of the training samples x_i belonging to the positive and negative class are p_{+i} and p_{-i}, respectively. The entropy of x_i is defined as

$$H_i = -p_{+i} \cdot \ln(p_{+i}) - p_{-i} \cdot \ln(p_{-i}) \tag{16}$$

where *ln* represents the natural logarithm operator. The key point of calculating H_i by (16) is to evaluate the probability of each sample belong to positive and negative class. We calculate the K nearest neighbours of sample x_i and assign the values to p_{+i} and p_{-i} based on count of total positive and negative class neighbours, i.e.

$$p_{+i} = \frac{num_{+i}}{k},\ p_{-i} = \frac{num_{-i}}{k} \tag{17}$$

where num_{+i} and num_{-i} represent the number of positive and negative samples in the selected K nearest neighbours, and $num_{+i}+num_{-i} = k$.

By adopting the above entropy evaluation, the entropy of the positive samples are $H_+ = \{H_{+1}, H_{+2}, \dots, H_{+m_1}\}$. Then, data points of positive class are divided into N_+ subsets based on increasing order of entropy and the fuzzy memberships of positive samples in each subset are calculated as

$$FM_{+j} = 1.0 - \beta \times (j-1),\ j = 1, 2, \dots, N_+ \tag{18}$$

where FM_{+j} is the fuzzy membership for positive samples distributed in jth subset with fuzzy membership parameter $\beta \in (0, \frac{1}{N_+-1}]$ which controls the scale of the fuzzy values of positive samples. Then, the fuzzy membership of positive samples are defined as

$$s_{+i} = 1 - \beta \times (j-1),\quad if\ x_i \in jth\ subset\ (i = 1, 2, \dots, m_1) \tag{19}$$

Similarly, the entropy of the negative samples are $H_- = \{H_{-1}, H_{-2}, \dots, H_{-m_2}\}$. Then, data points of negative class are divided into N_- subsets based on increasing order of entropy and the fuzzy memberships of negative samples in each subset are calculated as

$$FM_{-j} = 1.0 - \beta \times (j-1), j = 1, 2, \dots, N_- \tag{20}$$

where FM_{-j} is the fuzzy membership for negative samples distributed in jth subset with fuzzy membership parameter $\beta \in (0, \frac{1}{N_--1}]$ which controls the scale of the fuzzy values of negative samples. Then, the fuzzy membership of negative samples are defined as

$$s_{-i} = 1 - \beta \times (j-1),\quad if\ x_i \in jth\ subset\ (i = 1, 2, \dots, m_2) \tag{21}$$

3.2 Linear EFLSTSVM

For binary classification problem, inspired by LSTSVM [11] and FTSVM [26], the proposed EFLSTSVM seeks a pair of nonparallel hyperplanes and modifies the primal problems of FTSVM by replacing the inequality constrains with equality constraints and using the square of 2-norm slack variables. Different from LSTSVM, the structural risk is minimized by adding a regularization term in our EFLSTSVM. Thus, the primal problems of EFLSTSVM are expressed as follows.

$$\begin{aligned} \min_{w_1, b_1} \quad & \frac{c_2}{2}(w_1^2 + b_1^2) + \frac{1}{2}(Aw_1 + e_1 b_1)^T (Aw_1 + e_1 b_1) + \frac{c_1}{2}\xi_2^T S_- \xi_2 \\ s.t. \quad & -(Bw_1 + e_2 b_1) = e_2 - \xi_2 \end{aligned} \tag{22}$$

$$\begin{aligned} \min_{w_2, b_2} \quad & \frac{c_4}{2}(w_2^2 + b_2^2) + \frac{1}{2}(Bw_2 + e_2 b_2)^T (Bw_2 + e_2 b_2) + \frac{c_3}{2}\xi_1^T S_+ \xi_1 \\ s.t. \quad & Aw_2 + e_1 b_2 = e_1 - \xi_1 \end{aligned} \tag{23}$$

where c_i $(i = 1, 2, 3, 4)$ are positive penalty parameters, $S_+ = diag(s_{+1}, s_{+2}, \dots, s_{+m_1})$ and $S_- = diag(s_{-1}, s_{-2}, \dots, s_{-m_2})$ are the entropy-based fuzzy membership of positive class and negative class, ξ_1 and ξ_2 are slack variables, e_1 and e_2 are the vectors of ones with appropriate dimensions.

Consider the primal problem (22), by substituting the equality constraint into the objective function, we obtain the unconstrained problem. Therefore, we obtain

$$L(w_1,b_1) = \frac{c_2}{2}(w_1^2 + b_1^2) + \frac{1}{2}(Aw_1 + e_1b_1)^T(Aw_1 + e_1b_1) + \frac{c_1}{2}(e_2 + Bw_1 + e_2b_1)^T S\_(e_2 + Bw_1 + e_2b_1) \tag{24}$$

By taking partial derivative of (24) with respect to w_1 and b_1, we get

$$\frac{\partial L}{\partial w_1} = c_2w_1 + A^T(Aw_1 + e_1b_1) + c_1B^T S\_(e_2 + Bw_1 + e_2b_1) = 0 \tag{25}$$

$$\frac{\partial L}{\partial b_1} = c_2b_1 + e_1^T(Aw_1 + e_1b_1) + c_1e_2^T S\_(e_2 + Bw_1 + e_2b_1) = 0 \tag{26}$$

Then, combing (25) and (26) and solving for w_1 and b_1, it is easy to lead to a system of linear equations which is expressed as follows.

$$\begin{pmatrix} c_2I_1 + A^TA + c_1B^TS\_B & A^Te_1 + c_1B^TS\_e_2 \\ e_1^TA + c_1e_2^TS\_B & c_2 + e_1^Te_1 + c_1e_2^TS\_e_2 \end{pmatrix}\begin{pmatrix} w_1 \\ b_1 \end{pmatrix}=\begin{pmatrix} -c_1B^TS\_e_2 \\ -c_1e_2^TS\_e_2 \end{pmatrix} \tag{27}$$

where I_1 is an identity matrix.

Let $H = [A\ e_1] \in R^{m_1\times(n+1)}, G = [B\ e_2] \in R^{m_2\times(n+1)}$, by simplifying (27), we can get

$$(c_2I_+ + H^TH + c_1G^TS\_G)u_+ = -c_1G^TS\_e_2 \tag{28}$$

where I_+ is an identity matrix and $u_+ = [w_1;b_1] \in R^{n+1}$.

Thus, form (28), we can get

$$u_+ = -c_1(c_2I_+ + H^TH + c_1G^TS\_G)^{-1}G^TS\_e_2 \tag{29}$$

Consider the primal problem (23), and substitute the equality constrains into the objective function. Thus, we can obtain

$$L(w_2,b_2) = \frac{c_4}{2}(w_2^2 + b_2^2) + \frac{1}{2}(Bw_2 + e_2b_2)^T(Bw_2 + e_2b_2) + \frac{c_3}{2}(e_1 - Aw_2 - e_1b_2)^T S_+(e_1 - Aw_2 - e_1b_2) \tag{30}$$

By taking partial derivative of (30) with respect to w_2 and b_2, we obtain

$$\frac{\partial L}{\partial w_2} = c_4w_2 + B^T(Bw_2 + e_2b_2) - c_3A^TS_+(e_1 - Aw_2 - e_1b_2) = 0 \tag{31}$$

$$\frac{\partial L}{\partial b_2} = c_4b_2 + e_2^T(Bw_2 + e_2b_2) - c_3e_1^TS_+(e_1 - Aw_2 - e_1b_2) = 0 \tag{32}$$

Then, combing (31) and (32) and solving for w_2 and b_2, it is easy to lead to a system of linear equations which is expressed as follows.

$$\begin{pmatrix} c_4I_2 + B^TB + c_3A^TS_+A & B^Te_2 + c_3A^TS_+e_1 \\ e_2^TB + c_3e_1^TS_+A & c_4 + e_2^Te_2 + c_3e_1^TS_+e_1 \end{pmatrix}\begin{pmatrix} w_2 \\ b_2 \end{pmatrix}=\begin{pmatrix} c_3A^TS_+e_1 \\ c_3e_1^TS_+e_1 \end{pmatrix} \tag{33}$$

where I_2 is an identity matrix.

Similarly, by simplifying (33), we can obtain

$$(c_4 I_- + G^T G + c_3 H^T S_+ H)u_- = c_3 H^T S_+ e_1 \tag{34}$$

where I_- is an identity matrix and $u_- = [w_2; b_2] \in R^{n+1}$.

Thus, form (34), we can get

$$u_- = c_3 (c_4 I_- + G^T G + c_3 H^T S_+ H)^{-1} H^T S_+ e_1 \tag{35}$$

The solutions to the pair of QPPs (22) and (23) can be found directly by solving two systems of linear Eqs. (28) and (34), which involves two matrix inverses of size $(n+1) \times (n+1)$. When n is much smaller than the number samples of positive class and negative class, the training speed of linear EFLSTSVM is extremely fast. Once w_1, b_1 and w_2, b_2 are obtained from (29) and (35), the two nonparallel hyperplanes (1) are known. A new data point $x \in R^n$ is then assigned to positive class W_1 or negative class W_2 by

$$x \in W_k,\ k = \arg\min_{k=1,2} \{|w_1^T x + b_1|, |w_2^T x + b_2|\} \tag{36}$$

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

In summary, the steps for constructing linear EFLSTSVM classifier are shown in Algorithm 1.

Algorithm 1. Linear EFLSTSVM Classifier

Step 1. Given the samples of positive class $A \in R^{m_1 \times n}$ and negative class $B^{m_2 \times n}$.

Step 2. Select appropriate neighborhood size k and construct the entropy-based fuzzy membership $S_+ = diag\{s_{+1}, s_{+2}, \cdots, s_{+m_1}\}$ and $S_- = diag\{s_{-1}, s_{-2}, \cdots, s_{-m_2}\}$ by adopting (19) and (21).

Step 3. Choose the proper penalty parameters $c_i\ (i = 1,2,3,4)$ by cross-validation and obtain the solutions w_1, b_1 and w_2, b_2 from (29) and (35), respectively.

Step 4. Calculate perpendicular distances $|w_1^T x + b_1|$ and $|w_2^T x + b_2|$ for a new sample $x \in R^n$, and assign the sample to positive class or negative class using (36).

3.3 Nonlinear EFLSTSVM

For nonlinear case, firstly, we define $C = [A;B]$ and choose an appropriate kernel function K. Following the same idea, linear EFLSTSVM classifier can be extended to nonlinear version by considering the following kernel-generated surfaces

$$K(x^T, C^T)w_1 + b_1 = 0 \quad \text{and} \quad K(x^T, C^T)w_2 + b_2 = 0 \tag{37}$$

The primal problems of the nonlinear EFLSTSVM are expressed as follows.

$$\begin{aligned}&\min_{w_1,b_1} \ \frac{c_2}{2}(w_1^2+b_1^2)+\frac{1}{2}(K(A,C^T)w_1+e_1b_1)^T(K(A,C^T)w_1+e_1b_1)+\frac{c_1}{2}\xi_2^T S_-\xi_2\\ &s.t. \quad -(K(B,C^T)w_1+e_2b_1)=e_2-\xi_2\end{aligned} \tag{38}$$

$$\begin{aligned}&\min_{w_2,b_2} \ \frac{c_4}{2}(w_2^2+b_2^2)+\frac{1}{2}(K(B,C^T)w_2+e_2b_2)^T(K(B,C^T)w_2+e_2b_2)+\frac{c_3}{2}\xi_1^T S_+\xi_1\\ &s.t. \quad K(A,C^T)w_2+e_1b_2=e_1-\xi_1\end{aligned} \tag{39}$$

Similar to linear case, by substituting the constraints into objective function, then we can obtain the solutions to the problems (38) and (39) as follows.

$$\begin{aligned}\min_{w_1,b_1} \ &\frac{c_2}{2}(w_1^2+b_1^2)+\frac{1}{2}(K(A,C^T)w_1+e_1b_1)^T(K(A,C^T)w_1+e_1b_1)\\ &+\frac{c_1}{2}(e_2+K(B,C^T)w_1+e_2b_1)^T S_-(e_2+K(B,C^T)w_1+e_2b_1)\end{aligned} \tag{40}$$

$$\begin{aligned}\min_{w_2,b_2} \ &\frac{c_4}{2}(w_2^2+b_2^2)+\frac{1}{2}(K(B,C^T)w_2+e_2b_2)^T(K(B,C^T)w_2+e_2b_2)\\ &+\frac{c_3}{2}(e_1-K(A,C^T)w_2-e_1b_2)^T S_+(e_1-K(A,C^T)w_2-e_1b_2)\end{aligned} \tag{41}$$

Let $KerH=[K(A,C^T) \ \ e_1]\in R^{m_1\times(m+1)}, KerG=[K(B,C^T) \ \ e_2]^{m_2\times(m+1)}$, similar to linear case, the solutions of QPPs (40) and (41) can be obtained as follows.

$$(c_2I_+ + KerH^T\cdot KerH+c_1KerG^T\cdot S_-\cdot KerG)\cdot v_+=-c_1KerG^T\cdot S_-\cdot e_2 \tag{42}$$

$$(c_4I_- + KerG^T\cdot KerG+c_3KerH^T\cdot S_+\cdot KerH)\cdot v_-=c_3KerH^T\cdot S_+\cdot e_1 \tag{43}$$

where $v_+=[w_1;b_1]\in R^{m+1}$, $v_-=[w_2;b_2]\in R^{m+1}$.

By simplifying the expression (42) and (43), we obtain

$$v_+=-c_1(c_2I_+ + KerH^T\cdot KerH+c_1KerG^T\cdot S_-\cdot KerG)^{-1}\cdot KerG^T\cdot S_-\cdot e_2 \tag{44}$$

$$v_-=c_3(c_4I_- + KerG^T\cdot KerG+c_3KerH^T\cdot S_+\cdot KerH)^{-1}\cdot KerH^T\cdot S_+\cdot e_1 \tag{45}$$

In summary, the solutions to the pair of QPPs (38) and (39) can be found directly by solving two systems of linear Eqs. (42) and (43). Once w_1, b_1 and w_2, b_2 are obtained from (44) and (45), the two nonparallel hyperplanes (37) are known. A new data point $x\in R^n$ is then assigned to positive class W_1 or negative class W_2 by

$$x\in W_k, \ k=\arg\min_{k=1,2}\{|K(x,C^T)w_1+b_1|, |K(x,C^T)w_2+b_2|\} \tag{46}$$

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

Similar to linear EFLSTSVM, steps for constructing the nonlinear EFLSTSVM classifier are given in Algorithm 2 as follows.

Algorithm 2. Nonlinear EFLSTSVM Classifier

Step 1. Given the samples of positive class $A \in R^{m_1 \times n}$ and negative class $B^{m_2 \times n}$, and choose an appropriate kernel function K based on validation.

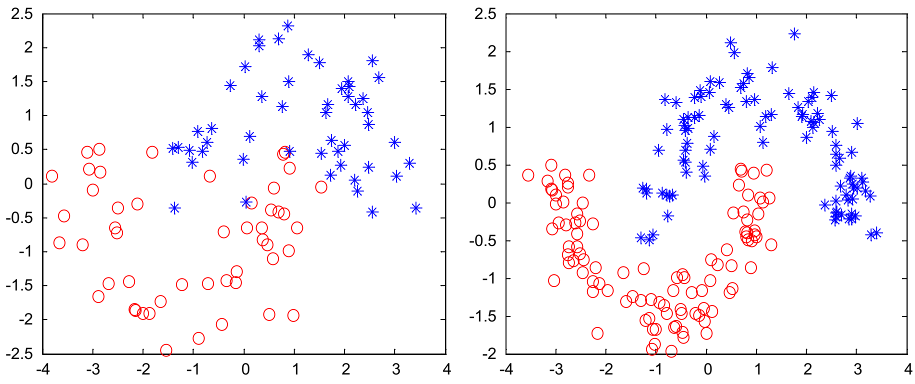
Step 2. Select appropriate neighborhood size k and construct the entropy-based fuzzy membership $S_+ = diag\{s_{+1}, s_{+2}, \cdots, s_{+m_1}\}$ and $S_- = diag\{s_{-1}, s_{-2}, \cdots, s_{-m_2}\}$ by adopting (19) and (21).

Step 3. Choose the proper penalty parameters c_i $(i = 1,2,3,4)$ by cross-validation and obtain the solutions w_1, b_1 and w_2, b_2 from (44) and (45), respectively.

Step 4. Calculate perpendicular distances $|K(x, C^T)w_1 + b_1|$ and $|K(x, C^T)w_2 + b_2|$ for a new sample $x \in R^n$, and then assign the sample to positive class or negative class using (46).

4 Experimental Results and Discussions

In order to validate the performance of the proposed EFLSTSVM, we investigate its classification accuracy and computational efficiency on several synthetic datasets, UCI benchmark datasets, NDC datasets and image recognition datasets, respectively. In our experiments, we focus on the comparison between our proposed EFLSTSVM and five state-of-the-art classifiers, including TWSVM [10], LSTSVM [11], FTSVM [26], EFSVM [31] and EFTWSVM-CIL [32]. And the classification accuracy of each method is evaluated by standard 5-fold cross-validation methodology. All methods are implemented in MATLAB R2018a on a personal computer (PC) with an Intel (R) Core (TM) i7-7700CPU (3.60 GHz$\times$8) and 32 GB random-access memory (RAM). The QPPs in TWSVM, FTSVM and EFTWSVM-CIL are solved by SOR algorithm, which is also used to solve QPPs in literatures [12, 18]. And the systems of linear equations in LSTSVM and our EFLSTSVM are solved by '\'. In addition, the grid search method is used to find the optimal parameters in all methods. Specifically, the penalty parameters c_i and kernel wide parameter σ of Gaussian kernel function $K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$ in all methods are selected form the set $\{2^i | i = -8, -7, \ldots, 7, 8\}$. The neighborhood size k in our EFLSTSVM, EFSVM and EFTWSVM-CIL are chosen from the set $\{1, 3, 5, \ldots, 17, 19\}$. The number of the separated subsets N_+ and N_- are set to 10 and the fuzzy membership parameter β is set to 0.05.



(**a**) Two-moons-1 dataset (**b**) Two-moons-2 dataset

Fig. 1 Two kinds of Two-moons datasets with different complexity

4.1 Synthetic Datasets

In this subsection, two artificial datasets, including Two-moons manifold dataset [35, 36] and Ripley's synthetic datasets [37] have been used to illustrate that the proposed EFLST-SVM can deal with linearly inseparable problems. In experiments, Two-moons-1 manifold dataset contains 100 samples (50 positive samples and 50 negative samples) and Two-moons-2 manifold dataset contains 200 samples (100 positive samples and 100 negative samples). Figure 1 shows two kinds of Two-moons datasets with different complexity.

For Two-moons manifold datasets, we investigate the performance of nonlinear TWSVM, LSTSVM, EFSVM, FTSVM, EFTWSVM- CIL and our EFLSTSVM with Gaussian kernel function. We randomly select 40% for training sets and 60% for testing sets, each experiment repeat 10 times and the average results are listed in Table 1. For Ripley's synthetic dataset, which contains 250 training points and 1000 test points, we investigate the classification performance of nonlinear TWSVM, LSTSVM, EFSVM, FTSVM, EFTWSVM-CIL and our proposed EFLSTSVM with Gaussian kernel function. The test accuracy is listed in Table 2. From Tables 1 and 2, we can observe that our EFLSTSVM obtains the best performance on Two-moons and Ripley datasets. Moreover, the hyperplanes of TWSVM, LSTSVM, EFSVM, FTSVM, EFTWSVM-CIL and our EFLSTSVM on Ripley's dataset are shown in Fig. 2.

Table 1 Classification accuracy on Two-moons datasets

Dataset	TWSVM Acc ± Std	LSTSVM Acc ± Std	EFSVM Acc ± Std	FTSVM Acc ± Std	EFTWSVM-CIL Acc ± Std	EFLSTSVM Acc ± Std
Two-moons-1	93.83 ± 2.94	94.00 ± 2.11	93.67 ± 1.53	92.17 ± 2.16	93.50 ± 2.54	**94.17 ± 2.39**
Two-moons-2	99.42 ± 0.40	99.17 ± 0.96	99.75 ± 0.40	99.68 ± 0.32	99.83 ± 0.35	**99.92 ± 0.26**

Table 2 Classification accuracy on Ripley's datasets

Dataset	TWSVM Acc	LSTSVM Acc	EFSVM Acc	FTSVM Acc	EFTWSVM-CIL Acc	EFLSTSVM Acc
Ripley	88.80	88.40	88.00	87.60	89.20	**89.40**

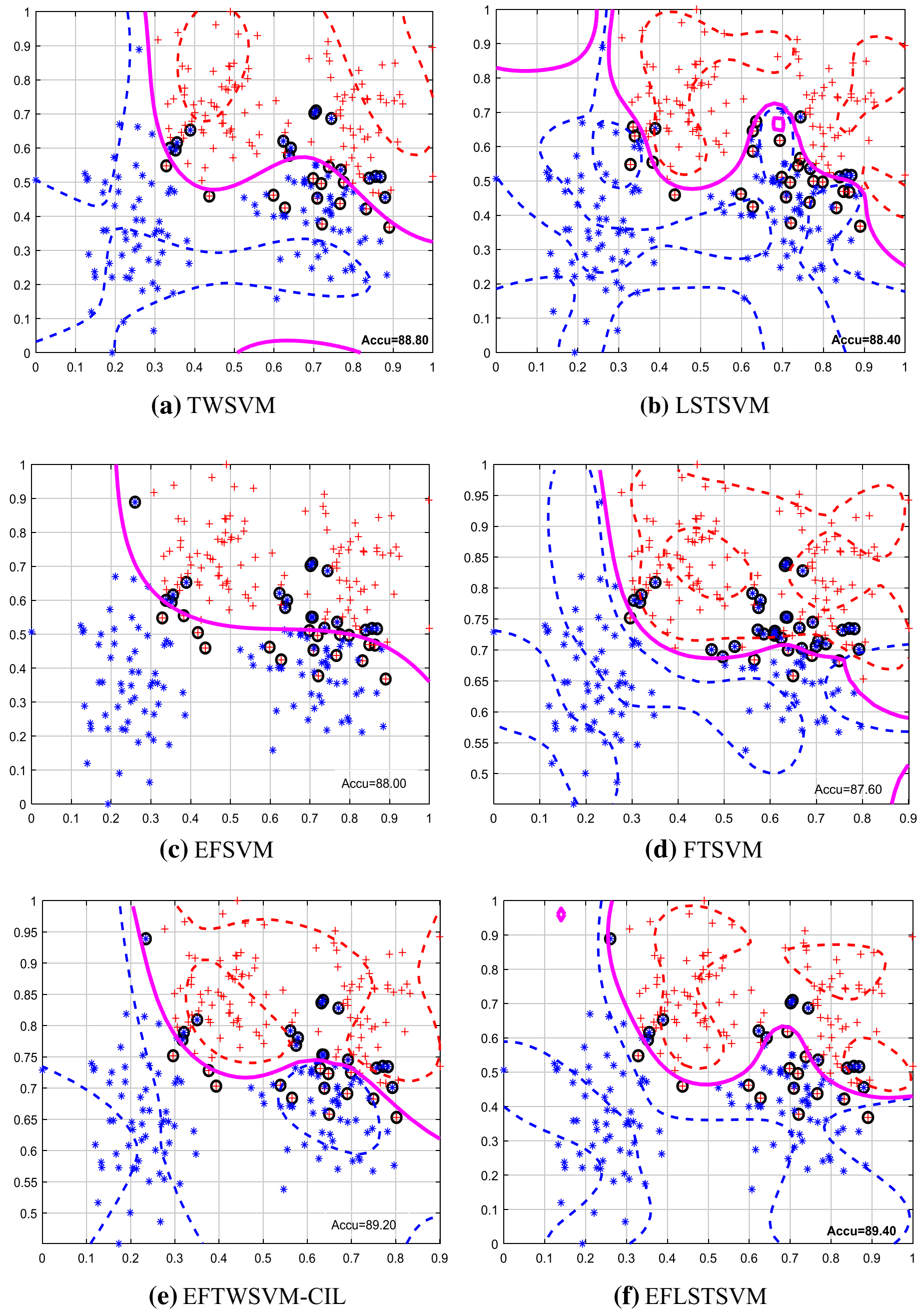


Fig. 2 Classification hyperplanes of nonlinear methods on Ripley's datasets. **a** TWSVM, **b** LSTSVM, **c** EFSVM, **d** FTSVM, **e** EFTWSVM-CIL and **f** EFLSTSVM

Table 3 Test results of linear TWSVM, LSTSVM, EFSVM, FTSVM, EFTWSVM-CIL and EFLSTSVM

Datasets	TWSVM	LSTSVM	EFSVM	FTSVM	EFTWSVM-CIL	EFLSTSVM
	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)
Australian 690*14	87.54 ± 2.87 0.0155	86.96 ± 4.21 <u>0.0005</u>	86.52 ± 3.64 0.0328	**87.83 ± 2.53** 0.0137	87.68 ± 1.98 0.0153	87.59 ± 4.92 0.0011
Bupa-Liver 345*6	71.01 ± 6.56 0.0042	70.43 ± 4.55 <u>0.0003</u>	69.28 ± 4.49 0.0080	71.29 ± 5.08 0.0044	**73.33 ± 5.48** 0.0037	72.51 ± 5.24 0.0008
House-Votes 435*16	96.09 ± 1.31 0.0074	95.86 ± 2.97 <u>0.0004</u>	95.40 ± 2.70 0.0121	96.32 ± 1.49 0.0078	**96.78 ± 0.51** 0.0071	96.32 ± 1.36 0.0007
Heart-c 303*13	85.81 ± 2.48 0.0043	85.15 ± 3.30 <u>0.0004</u>	84.18 ± 2.33 0.0065	82.89 ± 3.25 0.0038	85.80 ± 3.83 0.0039	**86.13 ± 1.55** 0.0007
Heart-Statlog 270*13	85.56 ± 3.04 0.0039	85.19 ± 6.80 <u>0.0004</u>	85.56 ± 4.01 0.0057	85.60 ± 2.75 0.0038	85.93 ± 2.81 0.0037	**86.30 ± 2.81** 0.0005
Ionosphere 351*34	93.74 ± 5.09 0.0048	94.02 ± 3.94 <u>0.0005</u>	89.18 ± 2.16 0.0104	94.02 ± 3.69 0.0049	**94.87 ± 2.97** 0.0046	94.01 ± 3.11 0.0011
Musk 476*166	84.66 ± 4.76 0.0099	82.14 ± 3.87 <u>0.0014</u>	84.98 ± 4.49 0.0169	84.26 ± 4.16 0.0107	85.08 ± 1.76 0.0096	**85.91 ± 4.01** 0.0024
PimaIndian 768*8	77.86 ± 2.92 0.0164	77.74 ± 3.01 <u>0.0005</u>	77.21 ± 3.13 0.0345	78.13 ± 3.19 0.0187	**78.52 ± 1.54** 0.0198	78.13 ± 3.08 0.0011
Sonar 208*60	78.85 ± 1.93 0.0034	79.37 ± 4.84 <u>0.0005</u>	81.74 ± 4.45 0.0047	80.75 ± 4.85 0.0034	81.27 ± 4.19 0.0031	**82.29 ± 3.87** 0.0007
Spect 267*44	81.24 ± 4.73 0.0050	80.89 ± 5.76 <u>0.0005</u>	83.15 ± 4.35 0.0059	81.67 ± 4.15 0.0051	81.68 ± 4.86 0.0046	**83.89 ± 3.42** 0.0006
Wpbc 198*34	83.36 ± 4.48 0.0033	83.32 ± 4.83 <u>0.0004</u>	82.37 ± 4.12 0.0052	83.83 ± 3.16 0.0093	84.20 ± 4.61 0.0034	**84.35 ± 4.66** 0.0005
Average	84.16 ± 3.65	83.73 ± 4.38	83.60 ± 3.62	84.24 ± 3.48	85.01 ± 3.14	**85.22 ± 3.46**
Average rank	4.2273	4.9545	4.7727	3.3182	1.9091	**1.8182**

4.2 UCI Datasets

To further compare EFLSTSVM with TWSVM, LSTSVM, EFSVM, FTSVM and EFTWSVM-CIL, we choose 11 datasets from UCI machine learning repository [38]. Specifically, they are Australian, Bupa-Liver, House-Votes, Heart-c, Heart-Statlog, Ionosphere, Musk, PimaIndian, Sonar, Spect and Wpbc, respectively. Experimental results of their linear and nonlinear versions are given in Tables 3 and 4. The best accuracy is shown in boldface and the shortest time is shown by underline for each dataset. In Table 3, we can find that the accuracy of our proposed linear EFLSTSVM is better than that of TWSVM, LSTSVM, EFSVM, FTSVM and EFTWSVM-CIL on most of the datasets. For example, for the Spect dataset, the accuracy of our linear EFLSTSVM is 83.89%, while TWSVM is 81.24%, LSTSVM is 80.89%, EFSVM is 83.15%, FTSVM is 81.67% and EFTWSVM-CIL is 81.68%, respectively.

Furthermore, Table 4 shows the experimental results for the nonlinear TWSVM, LSTSVM, EFSVM, FTSVM, EFTWSVM-CIL and our proposed EFLSTSVM on the above 11 UCI datasets. The results in Table 4 are similar to those in Table 3, and it also confirms the observation above. Especially for Ionosphere dataset, our nonlinear EFLSTSVM obtains the classification accuracy 96.58%, which is 2.26% higher than TWSVM, 2.28% higher than LSTSVM, 0.85% higher than EFSVM and 1.98% higher than FTSVM and EFTWSCM-CIL,

Table 4 Test results of nonlinear TWSVM, LSTSVM, EFSVM, FTSVM, EFTWSVM-CIL and EFLSTSVM

Datasets	TWSVM	LSTSVM	EFSVM	FTSVM	EFTWSVM-CIL	EFLSTSVM
	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)	Acc + Std (%) Time (s)
Australian 690*14	87.83 ± 1.57 0.0279	87.54 ± 3.38 <u>0.0128</u>	87.10 ± 1.80 0.0971	87.39 ± 3.46 0.0294	87.68 ± 1.98 0.0291	**87.97 ± 0.97** 0.0152
Bupa-Liver 345*6	75.07 ± 4.39 0.0081	**75.94 ± 4.29** <u>0.0033</u>	73.91 ± 4.39 0.0209	74.78 ± 3.92 0.0097	74.49 ± 4.18 0.0083	75.65 ± 4.27 <u>0.0033</u>
House-Votes 435*16	96.32 ± 2.21 0.0129	**96.55 ± 2.11** <u>0.0049</u>	95.86 ± 3.11 0.0312	96.32 ± 2.74 0.0130	96.09 ± 1.92 0.0129	**96.55 ± 1.41** 0.0059
Heart-c 303*13	85.14 ± 3.52 0.0071	84.50 ± 3.12 <u>0.0026</u>	84.84 ± 4.64 0.0157	83.83 ± 2.17 0.0080	**86.13 ± 1.57** 0.0068	85.82 ± 3.39 0.0029
Heart-Statlog 270*13	85.56 ± 3.46 0.0063	85.19 ± 3.70 <u>0.0021</u>	85.19 ± 4.54 0.0131	85.86 ± 2.81 0.0060	**86.30 ± 2.81** 0.0058	85.93 ± 2.11 0.0023
Ionosphere 351*34	94.32 ± 3.32 0.0091	94.30 ± 2.03 <u>0.0037</u>	95.73 ± 2.26 0.0209	94.60 ± 3.67 0.0121	94.60 ± 3.51 0.0086	**96.58 ± 1.62** 0.0046
Musk 476*166	95.79 ± 2.11 0.0153	94.96 ± 2.40 <u>0.0076</u>	94.54 ± 0.88 0.0767	96.43 ± 1.89 0.0163	95.58 ± 2.41 0.0157	**96.63 ± 2.02** 0.0087
PimaIndian 768*8	78.52 ± 4.27 0.0356	78.30 ± 3.17 <u>0.0166</u>	77.74 ± 4.42 0.0919	78.13 ± 0.54 0.0402	78.64 ± 2.64 0.0384	**79.16 ± 1.83** 0.0232
Sonar 208*60	88.95 ± 2.68 0.0046	91.39 ± 3.69 0.0018	90.88 ± 2.57 0.0116	89.94 ± 4.52 0.0047	**91.86 ± 3.58** 0.0041	91.81 ± 4.74 <u>0.0017</u>
Spect 267*44	82.42 ± 4.15 0.0072	83.89 ± 4.54 <u>0.0022</u>	82.78 ± 5.42 0.0061	83.15 ± 3.49 0.0077	83.52 ± 4.22 0.0074	**84.65 ± 0.74** 0.0026
Wpbc 198*34	83.86 ± 6.18 0.0045	83.36 ± 5.56 <u>0.0015</u>	82.83 ± 3.23 0.0043	83.33 ± 5.18 0.0047	84.36 ± 3.74 0.0042	**84.88 ± 4.02** <u>0.0015</u>
Average	86.71 ± 3.44	86.90 ± 3.45	86.49 ± 3.39	86.71 ± 3.13	87.20 ± 2.96	**87.78 ± 2.47**
Average rank	3.7727	3.8182	4.9545	4.2727	2.6818	**1.4091**

respectively. In addition, from Tables 3 and 4, we can find that our proposed EFLSTSVM is not the fastest method. Although it is a bit slower than LSTSVM in most case, it is faster than the other four methods on all chosen datasets. The main reason might be that TWSVM, EFSVM, FTSVM and EFTWSVM-CIL are required to solve QPPs, while LSTSVM and our proposed EFLSTSVM are only required to solve systems of linear equations. Figure 3 shows the training times of nonlinear methods on selected UCI datasets.

In addition, Friedman test [39] is conducted to give a statistic comparison on the effectiveness with the compared algorithms. For this test, the average ranks of the compared algorithms on the used datasets are listed in the last row of Tables 3 and 4. In the experiments, we consider k (=6) number of compared algorithms and n (=11) number of datasets. Let r_i^j be the rank of the jth algorithms on the ith datasets. The average rank of the jth algorithms is calculated as $R_j = \frac{1}{n}\sum_{i=1}^{n} r_i^j$. We assume all the methods are equivalent under null hypothesis, and the Friedman statistic

$$\chi_F^2 = \frac{12n}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{47}$$

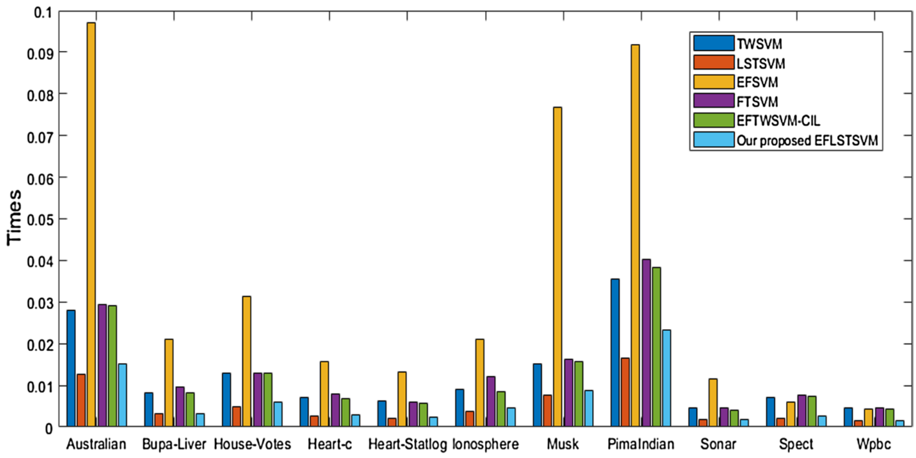


Fig. 3 Training times of nonlinear methods on selected UCI datasets

is distributed according to χ_F^2 with $(k-1)$ degrees of freedom, when n and k are reasonable large.

Moreover, Iman and Davenport [40] showed that Friedman's χ_F^2 presents a pessimistic behavior, and they derived a better statistic

$$F_F = \frac{(n-1)\chi_F^2}{n(k-1)-\chi_F^2} \tag{48}$$

which is distributed according to the F-distribution with $(k-1)$ and $(k-1)(n-1)$ degrees of freedom.

For linear case, in Table 3, it is noticed that our proposed EFLSTSVM ranks the first with an average score of 1.8182. To validate that the measured average ranks are significantly different from the mean rank by the null hypothesis, according to (47) and (48), we obtain

$$\chi_F^2 = \frac{12\times 11}{6\times 7}\left[(4.2273^2+4.9545^2+4.7727^2+3.3182^2+1.9091^2+1.8182^2)-\frac{6\times 7^2}{4}\right] = 30.3498$$

$$F_F = \frac{10\times 30.3498}{11\times 5-30.3498}=12.3122$$

Specifically, for 6 algorithms and 11 datasets, F_F is distributed according to the F distribution with $(6-1)$=5 and $(6-1)\times(11-1)$=50 degrees of freedom. We can find that the critical value of $F(5,50)$ is 2.400 for the level of significant α=0.05 and it is less than the value of F_F, which indicates the null hypothesis is rejected. Thus, the compared algorithms are significantly different on the adopted datasets.

For nonlinear case, in Table 4, it is noticed that EFLSTSVM ranks the first with an average score of 1.4091. According to (47) and (48), we obtain

$$\chi_F^2 = \frac{12\times 11}{6\times 7}\left[(3.7727^2+3.8182^2+4.9545^2+4.2727^2+2.6818^2+1.4091^2)-\frac{6\times 7^2}{4}\right] = 22.9195$$

$$F_F = \frac{10\times 22.9195}{11\times 5-22.9195}= 7.1444$$

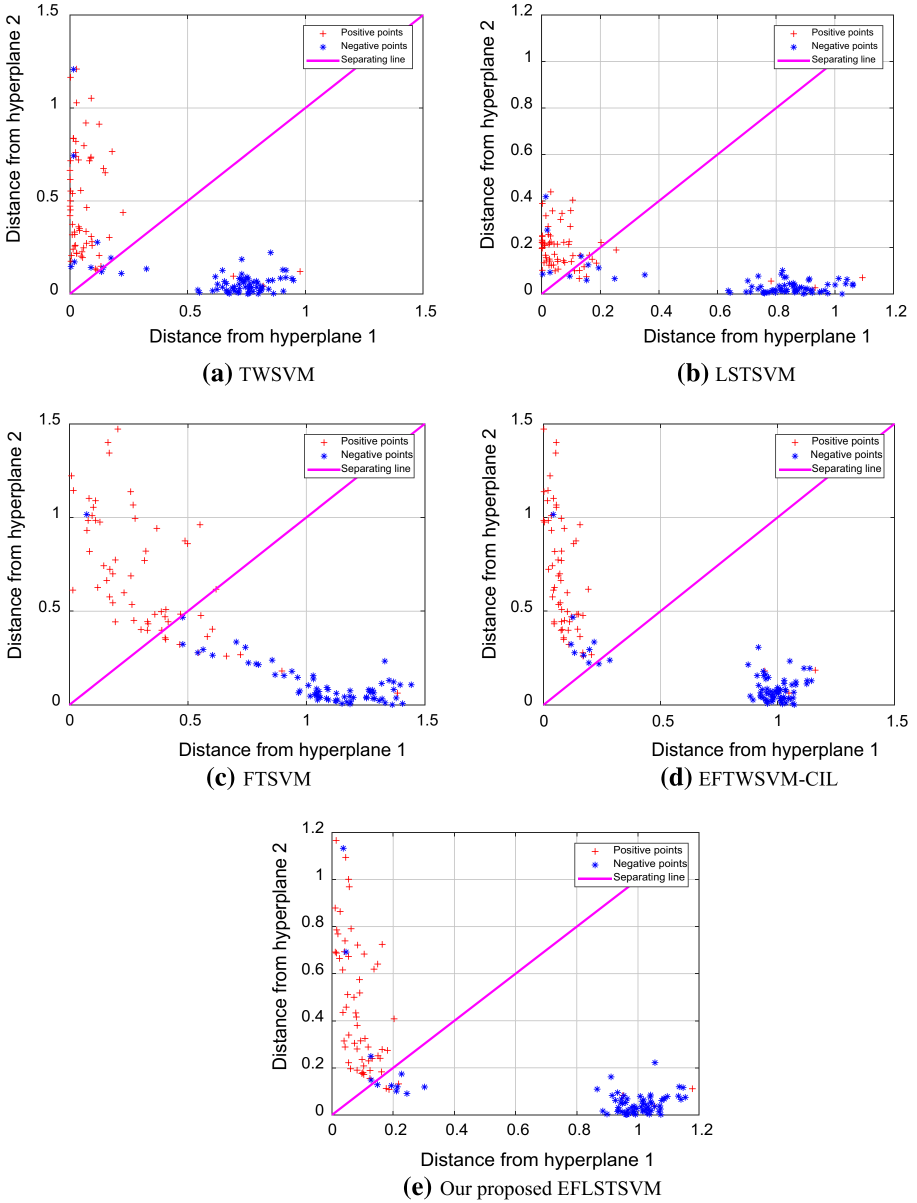


Fig. 4 Two-dimensional projection for test points from Australian dataset. **a** TWSVM, **b** LSTSVM, **c** FTSVM, **d** EFTWSVM-CIL and **e** Our proposed EFLSTSVM

Similar to linear case, for 6 algorithms and 11 datasets, the critical value of $F(5, 50)$ is 2.400 for the level of significant α=0.05 and it is also less than the value of F_F. Therefore, the null hypothesis is rejected and the compared algorithms are significantly different.

On the other hand, in order to compare the performance of different nonparallel hyperplanes support vector machines, i.e. TWSVM, LSTSVM, FTSVM, EFTWSVM- CIL and our proposed EFLSTSVM, we give the two-dimensional scatter plots [14, 18] of part test points for above classifiers. Here, Australian UCI data set is selected for a practical example and the corresponding scatter plots are shown in Fig. 4 for Australian data set with about 20% of data points, where the coordinates (d_i^+, d_i^-) are the respective distances of a test point x_i to the two hyperplanes. Specifically, in Fig. 4, the figures (a–e) are the results obtained by TWSVM, LSTSVM, FTSVM, EFTWSVM-CIL and our proposed EFLSTSVM, respectively. From Fig. 4, it can be seen that most test samples are clustered around the corresponding hyperplanes for all methods. And it is clearly noticeable that our proposed EFLSTSVM obtains better clustered points and separated classes than other methods.

4.3 Image Recognition

In this subsection, we apply our method to image recognition. Three well-known and publicly available databases corresponding to typical image classifications, i.e., handwritten digits (USPS), objects (COIL-20) and recognition of faces (AR) are used to evaluate our proposed EFLSLSVM with TWSVM, LSTSVM, EFSVM, FTSVM and EFTWSVM-CIL.

Fig. 5 An illustration of 10 subjects in the USPS database

Fig. 6 An illustration of 20 subjects in the COIL-20 database

Fig. 7 An illustration of 14 images of one person from the AR database

The USPS database [41] consists of gray-scale handwritten digit images from 0 to 9, as shown in Fig. 5. Each digit contains 1100 images, and the size of each image is 16×16 pixels with 256 gray levels. Here we select five pairwise digits of varying difficulty for odd versus even digit classification.

COIL-20 [42] is a database of gray scale images of 20 objects, which are illustrated in Fig. 6. The objects were placed on a motorized turntable against a black background. Images of the objects were taken at pose intervals 5°, which corresponds to 72 images per object. In our experiments, we have resized each of the original 1440 images into 32×32 pixels.

The AR database [43] contains 100 subjects and each subject has 26 face images taken in two sessions. For each session, there are 13 face images. Here 14 unoccluded images from the two sessions of each person are chosen for experiments, which are shown in Fig. 7. In our experiments, the 1400 images are all cropped into the same size of 40×30 pixels.

For these datasets, we randomly partition the images of each project into two parts with same sizes such that one part is selected for training and the remaining part is used for testing. This process is repeated ten times. We only consider the Gaussian kernel for these methods and Table 5 lists the experimental results of these methods in USPS, COIL-20

Table 5 The classification performance comparison on the USPS, COIL-20 and AR datasets

Datasets	TWSVM	LSTSVM	EFSVM	FTSVM	EFTWSVM-CIL	EFLSTSVM
	Acc + Std	Acc + Std	Acc + Std	Acc + Std	Acc + Std	Acc + Std
USPS 1 versus 7	99.79 ± 0.12	99.83 ± 0.12	99.85 ± 0.09	99.87 ± 0.08	99.82 ± 0.13	**99.90 ± 0.07**
USPS 2 versus 3	98.43 ± 0.19	98.35 ± 0.50	99.18 ± 0.23	99.19 ± 0.21	99.15 ± 0.19	**99.33 ± 0.17**
USPS 2 versus 7	99.68 ± 0.12	99.54 ± 0.20	99.62 ± 0.17	99.63 ± 0.14	99.59 ± 0.18	**99.71 ± 0.11**
USPS 3 versus 8	98.49 ± 0.40	98.47 ± 0.39	98.90 ± 0.32	99.01 ± 0.30	98.99 ± 0.34	**99.04 ± 0.37**
USPS 4 versus 7	99.79 ± 0.14	99.75 ± 0.15	99.81 ± 0.12	99.82 ± 0.11	99.81 ± 0.12	**99.85 ± 0.07**
COIL-20	99.35 ± 0.35	98.19 ± 1.06	99.58 ± 0.44	99.59 ± 0.28	**99.71 ± 0.28**	99.61 ± 0.52
AR	96.84 ± 0.93	95.93 ± 0.87	98.29 ± 0.53	98.33 ± 0.37	**98.93 ± 0.49**	98.36 ± 0.77

Table 6 The characteristics of benchmark datasets

Datasets	Training data	Testing data	Features
NDC-500	500	50	32
NDC-1000	1000	100	32
NDC-2000	2000	200	32
NDC-3000	3000	300	32
NDC-5000	5000	500	32
NDC-8000	8000	800	32
NDC-10000	10,000	1000	32

Table 7 Comparison on NDC datasets for linear classifiers

Datasets	TWSVM	LSTSVM	EFSVM	FTSVM	EFTWSVM-CIL	EFLSTSVM
	Train/test (%) Time (s)	Train/test (%) Time (s)	Train/test (%) Time (s)	Train/test (%) Time (s)	Train/test (%) Time (s)	Train/test (%) Time (s)
NDC-500	96.00/88.00 0.0229	95.40/88.00 0.0012	94.60/82.00 0.0841	95.40/84.00 0.0274	95.80/88.00 0.0366	95.80/90.00 0.0016
NDC-1000	95.30/92.00 0.0519	94.90/93.00 0.0016	94.90/94.00 0.1545	95.40/94.00 0.0895	95.30/92.00 0.0568	94.90/94.00 0.0028
NDC-2000	94.90/95.00 0.2345	94.85/95.50 0.0018	94.85/94.50 0.7506	95.00/95.50 0.3872	94.85/95.00 0.3206	95.00/96.00 0.0035
NDC-3000	95.17/94.33 0.7046	94.70/94.00 0.0030	94.77/94.67 1.9055	94.87/95.00 0.7688	95.17/94.67 0.7209	95.50/94.33 0.0059
NDC-5000	94.74/94.40 2.7820	94.78/94.20 0.0090	94.58/94.00 8.0141	94.74/94.40 2.9094	94.74/94.40 2.8640	94.90/94.40 0.0141
NDC-8000	94.94/92.37 9.3884	94.67/91.87 0.0135	94.91/92.00 32.1632	94.86/91.87 10.0274	94.95/92.37 9.5797	94.91/92.00 0.0454
NDC-10000	95.22/94.40 17.5482	95.01/94.60 0.0347	95.12/94.80 64.2797	95.26/94.30 17.2046	95.22/94.40 18.6533	95.22/94.40 0.0702

and AR datasets. It can be seen that the proposed EFLSTSVM obtains the best classification performance than the other five methods on USPS datasets. Although our proposed EFLSTSVM obtains lower classification accuracies than EFTWSVM-CIL on COIL-20 and AR datasets, it also obtains higher accuracies than the other four methods.

4.4 NDC Datasets

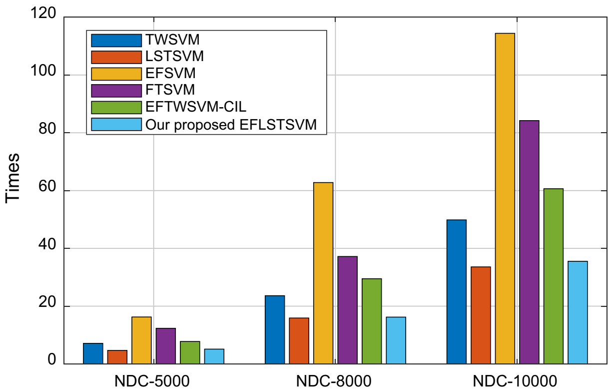
In this subsection, we conduct some experiments on large scale classification datasets. So, the David Musicants NDC Data Generator [44] is used to evaluate the computation time for all methods with respect to number of data points. Table 6 lists a description of NDC datasets, each dataset is divided into a training set and testing set.

For the experiments on NDC datasets, we fixed parameters of all methods to be the same (i.e. c_i= 1,σ = 1,k = 9). The training accuracy, testing accuracy and training time of linear and nonlinear classifiers are reported in Tables 7 and 8, respectively. In particular, Table 7 shows the comparison results for linear TWSVM, LSTSVM, EFSVM, FTSVM, EFTWSVM-CIL and our proposed EFLSTSVM on NDC datasets. In Table 7, we can see that our EFLSTSVM obtains the comparable accuracies and performs faster than other

Table 8 Comparison on NDC datasets for nonlinear classifiers

Datasets	TWSVM	LSTSVM	EFSVM	FTSVM	EFTWSVM-CIL	EFLSTSVM
	Train/test (%) Time (s)	Train/test (%) Time (s)	Train/test (%) Time (s)	Train/test (%) Time (s)	Train/test (%) Time (s)	Train/test (%) Time (s)
NDC-500	100/98.00 0.0314	99.00/98.00 0.0241	98.00/98.00 0.1326	99.00/98.00 0.0370	100/98.00 0.0328	100/98.00 0.0218
NDC-1000	100/99.00 0.1375	100/99.00 0.1168	99.10/98.00 0.7052	99.00/98.00 0.1411	100/99.00 0.1406	100/99.00 0.1061
NDC-2000	100/100 0.7116	99.10/99.50 0.6102	99.05/99.50 2.1702	99.50/98.00 1.5637	100/99.00 0.8274	100/99.50 0.6699
NDC-3000	100/100 1.8373	99.50/98.00 1.4127	99.50/98.00 5.0040	100/99.00 3.1360	99.00/98.00 2.1400	99.50/99.00 1.4233
NDC-5000	100/100 7.1206	100/99.60 4.6743	99.68/99.20 16.2511	100/99.60 12.3371	100/99.20 7.7756	100/99.60 5.1280
NDC-8000	100/100 23.6220	99.80/98.50 15.8974	99.70/99.25 62.7824	99.50/99.00 37.1907	99.50/99.25 29.4945	100/99.80 16.2110
NDC-10000	100/100 49.8520	100/99.90 33.5872	99.70/99.80 114.4295	100/99.80 84.2280	100/99.90 60.6392	100/99.90 35.5281

Fig. 8 Training times of all nonlinear classifiers on three large scale NDC datasets



methods on most datasets. In addition, for nonlinear case, Table 8 shows the comparison results of all methods conducted on NDC datasets with Gaussian kernel. The results on these datasets also illustrate that LSTSVM and our EFLSTSVM are much faster than TWSVM, EFSVM, FTSVM and EFTWSVM-CIL. The reason might be that LSTSVM and our proposed EFLSTSVM are only solving systems of linear equations rather than quadratic programming in other methods.

For linear case, taking NDC-10000 dataset for example, the training time of TWSVM is 17.5482 s, EFSVM is 64.2797 s and FTSVM is 17.2046 s, EFTWSVM-CIL is 18.6533 s, while LSTSVM is 0.0347 s and our EFLSTSVM is 0.0702 s, respectively. Moreover, for nonlinear case, the training times of all methods on three large scale NDC datasets, e.g. NDC-5000, NDC-8000 and NDC-10000, are shown in Fig. 8. Thus, the results of Tables 7, 8 and Fig. 8 can indicate the efficiency of our proposed EFLSTSVM when dealing with large scale problems.

4.5 Further Discussion

In the proposed EFLSTSVM, there are so many parameters, i.e. the number of nearest neighbor k, the positive penalty parameters c_1, c_2, c_3, c_4 and the kernel wide parameter σ for nonlinear case. However, these parameters significantly impact the classification performance of the proposed EFLSTSVM. In order to investigate the influence of these parameters to EFLSTSVM, we select 5 datasets from Table 4 and discuss their effects to the classification performance to EFLSTSVM. For simplicity, we choose the penalty parameters $c_1 = c_3$ and $c_2 = c_4$. In addition, we should be declared that when discussing on k, other parameters are set to the best parameters which are selected by fivefold cross-validation. Take Sonar

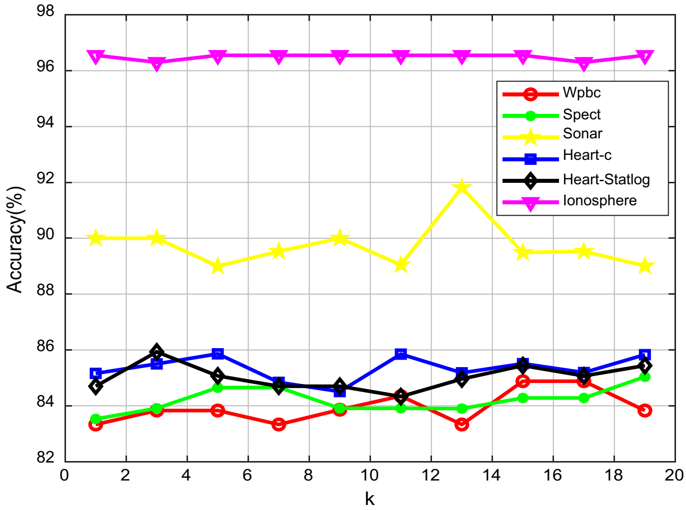


Fig. 9 Accuracy of EFLSTSVM with respect to k on the selected datasets

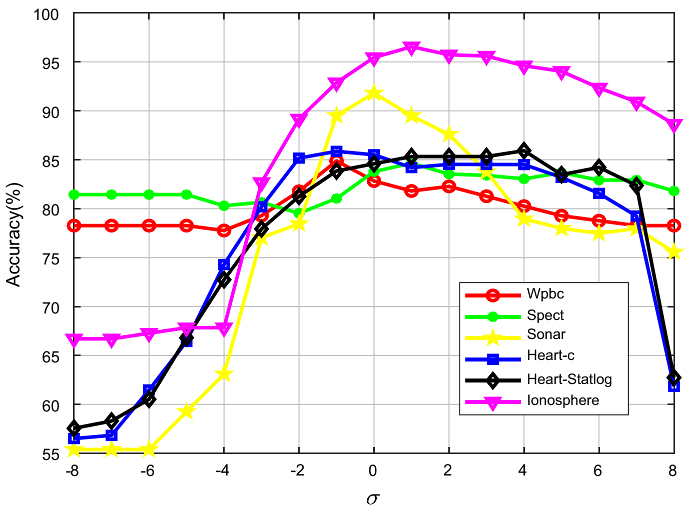
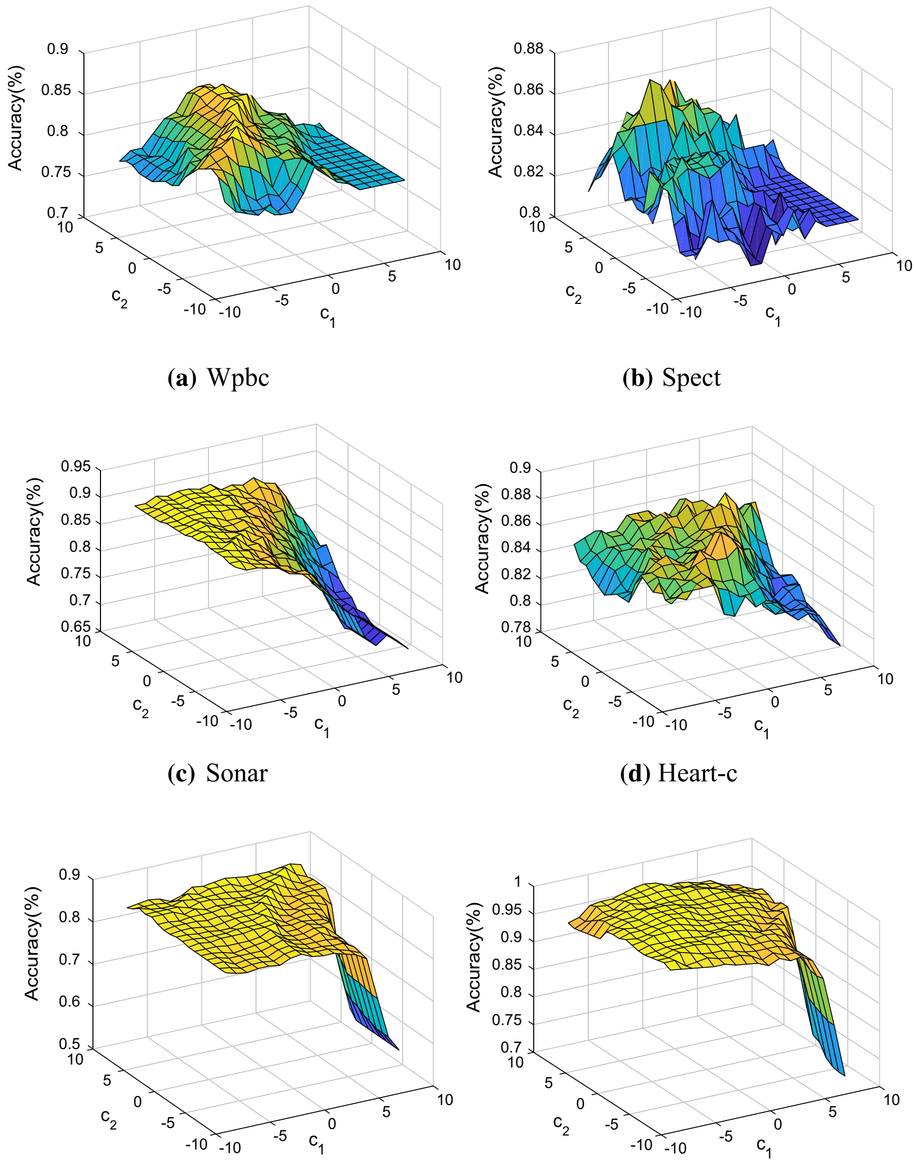


Fig. 10 Accuracy of EFLSTSVM with respect to σ on the selected datasets



(a) Wpbc **(b)** Spect

(c) Sonar **(d)** Heart-c

(e) Heart-Statlog **(f)** Ionosphere

Fig. 11 Accuracy of EFLSTSVM with respect to $c_1 = c_3, c_2 = c_4$ on the selected datasets. **a** Wpbc, **b** Spect, **c** sonar, **d** Heart-c, **e** Heart-Statlog and **f** Ionosphere

dataset for example, the best parameters of Sonar are $k = 13, c_1 = c_3 = 1, c_2 = c_4 = 2^{-6}$ and $\sigma = 1$. Specifically, when discussing on k, k is selected from the set of $\{1, 3, \ldots, 17, 19\}$ and the other parameters are set to $c_1 = c_3 = 1, c_2 = c_4 = 2^{-6}, \sigma = 1$, respectively. When discussing on σ, σ is selected from the set of $\{2^{-8}, 2^{-7}, \ldots, 2^7, 2^8\}$ and the other parameters are set to $c_1 = c_3 = 1, c_2 = c_4 = 2^{-6}, k = 13$. When discussing on c_i, the parameter k is set to 13, σ is set to 1 and $c_1 = c_3$,$c_2 = c_4$ are selected from the set of $\{2^{-8}, 2^{-7}, \dots, 2^7, 2^8\}$. At last, classification accuracies of the proposed EFLSTSVM with respect to k, σ and c_i on adopted datasets are shown in Figs. 9, 10 and 11, respectively.

Further, according to (44) and (45), it can be noted that the solutions of nonlinear EFLSTSVM require calculating the matrix inversion of size $(m+1) \times (m+1)$ twice. Therefore, the computational complexity of the matrix inversion will be high when m is large enough. However, using Sherman–Morrison–Woodbury (SMW) formula [45], the solution of (42) and (43) can be solved using for inverses of smaller dimension than $(m+1) \times (m+1)$. Thus, the expression (44) and (45) can be rewritten as follows.

$$v_+ = -c_1 \big(Y - Y \cdot KerH^T \cdot (\bar{I}_+ + KerH \cdot Y \cdot KerH^T)^{-1} \cdot KerH \cdot Y \big) \cdot KerG^T \cdot S_- \cdot e_2 \tag{49}$$

$$v_- = c_3 \big(Z - Z \cdot KerG^T \cdot (\bar{I}_- + KerG \cdot Z \cdot KerG^T)^{-1} \cdot KerG \cdot Z \big) \cdot KerH^T \cdot S_+ \cdot e_1 \tag{50}$$

$$\begin{aligned} \mathrm{Y} &= (c_2 I_+ + c_1 KerH^T \cdot S_- \cdot KerG)^{-1} \\ &= \frac{1}{c_2} \left(I_+ - KerG^T \cdot S_- \cdot (\frac{c_2}{c_1} \bar{I}_- + KerG \cdot KerG^T \cdot S_-)^{-1} \cdot KerG \right) \end{aligned} \tag{51}$$

$$\begin{aligned} \mathrm{Z} &= (c_4 I_- + c_3 KerH^T \cdot S_+ \cdot KerH)^{-1} \\ &= \frac{1}{c_4} \left(I_- - KerH^T \cdot S_+ \cdot (\frac{c_4}{c_3} \bar{I}_+ + KerH \cdot KerH^T \cdot S_+)^{-1} \cdot KerH \right) \end{aligned} \tag{52}$$

where $\bar{I}_+ \in R^{m_1 \times m_1}, \bar{I}_- \in R^{m_2 \times m_2}$ are identity matrices.

After using SMW formula, according to (49), (50), (51) and (52), the solutions of (42) and (43) require inversion of matrix of size $m_1 \times m_1$ and $m_2 \times m_2$ twice. Thus, when m is more than m_1 and m_2, we can utilize SMW formula to reduce computational complexity in our experiment.

5 Conclusions

In this paper, an entropy-based fuzzy least squares twin support vector machine (EFLSTSVM) is proposed by adopting the entropy to evaluate the class certainty of each sample and calculating the corresponding fuzzy membership based on the class certainty. This method pays more attention to the samples with higher class certainty to result in more robust decision surface, which enhances the classification accuracy and generalization ability. Experimental results obtained on synthetic and real-world datasets illustrate the effectiveness of our proposed EFLSTSVM. It should be pointed out that there are many parameters in our EFLSTSVM, so the parameter selection is a practical problem. Therefore, in the future work, we will propose some technique to select the optimal parameters and further improve the performance of the proposed EFLSTSVM. In addition, the extension of EFLSTSVM to multi-class classification [46, 47], multi-view classification [48, 49], semi-supervised classification [50, 51] and expand its practical application areas are also interesting.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (Grant No. 61702012), the University Outstanding Young Talent Support Project of Anhui Province of China (Grant No. gxyq2017026), the University Natural Science Research Project of Anhui Province of China (Grant Nos. KJ2016A431, KJ2017A361 and KJ2017A368) and the Program for Innovative Research Team in Anqing Normal University.

Compliance with Ethical Standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Cortes C, Vapnik VN (1995) Support-vector networks. Mach Learn 20(3):273–297
2. Osuna E, Freund R, Girosi F (1997) Training support vector machines: an application to face detection. In: Proceedings of computer vision and pattern recognition, pp 130–136
3. Noble WS (2004) Kernel methods in computational biology. In: Schölkopf B, Tsuda K, Vert JP (eds) Support vector machine applications in computational biology. MIT, Cambridge, pp 71–92
4. Isa D, Lee LH, Kallimani VP, Rajkumar R (2008) Text document preprocessing with the Bayes formula for classification using the support vector machine. IEEE Trans Knowl Data Eng 20(9):1264–1272
5. Khan L, Awad M, Thuraisingham B (2007) A new intrusion detection system using support vector machines and hierarchical clustering. VLDB J 16(4):507–521
6. Schmidt M, Gish H (1996) Speaker identification via support vector classifiers and hierarchical clustering. In: Proceedings of IEEE international conference on acoustics, speech, and signal processing, pp 105–108
7. Zhang J, Liu Y (2004) Cervical cancer detection using SVM-based feature screening. In: Proceedings of international conference on medical image computing and computer-assisted intervention, pp 873–880
8. Fung G, Mangasarian OL (2001) Proximal support vector machine classifiers. In: Proceedings KDD-2001, knowledge discovery and data mining, pp 77–86
9. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
10. Jayadeva Khemchandai R, Chandra S (2007) Twin support vector machine classification for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
11. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36(4):7535–7543
12. Shao YH, Zhang CH, Wang XB, Deng NY (2011) Improvements on twin support vector machines. IEEE Trans Neural Netw 22(6):962–968
13. Peng XJ (2011) TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. Pattern Recognit 44(10):2678–2692
14. Qi ZQ, Tian YJ, Shi Y (2013) Robust twin support vector machine for pattern classification. Pattern Recognit 46(1):305–316
15. Tian YJ, Qi ZQ, Ju XC, Shi Y, Liu XH (2014) Nonparallel support vector machines for pattern classification. IEEE Trans Cybern 44(7):1067–1079
16. Rastogi R, Saigal P, Chandra S (2018) Angle-based twin parametric-margin support vector machine for pattern classification. Knowl Based Syst 139:64–77
17. Mehrkanoon S, Huang XL, Suykens JAK (2014) Non-parallel support vector classifiers with different loss functions. Neurocomputing 143:294–301

18. Chen SG, Wu XJ, Zhang RF (2016) A novel twin support vector machine for binary classification problems. Neural Process Lett 44(3):795–811
19. Ding SF, An YX, Zhang XK, Wu FL, Xue Y (2017) Wavelet twin support vector machine based on glowworm swarm optimization. Neurocomputing 225:157–163
20. Xu YT, Yang ZJ, Pan XL (2017) A novel twin support vector machine with pinball loss. IEEE Trans Neural Netw Learn Syst 28(2):359–370
21. Lin CF, Wang SD (2002) Fuzzy support vector machines. IEEE Trans Neural Netw 13(2):464–471
22. Batuwita R, Palade V (2010) FSVM-CIL: fuzzy support vector machines for class imbalance learning. IEEE Trans Fuzzy Syst 18(3):558–571
23. Wang Y, Wang S, Lai KK (2005) A new fuzzy support vector machine to evaluate credit risk. IEEE Trans Fuzzy Syst 13(6):820–831
24. Tao Q, Wang J (2004) A new fuzzy support vector machine based on the weighted margin. Neural Process Lett 20(3):139–150
25. An WJ, Liang MG (2013) Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises. Neurocomputing 110(7):101–110
26. Li K, Ma HY (2013) A fuzzy twin support vector machine algorithm. Int J Appl Innov Eng Manag 2(3):459–465
27. Chen SG, Wu XJ (2017) A new fuzzy twin support vector machine for pattern classification. Int J Mach Learn Cybern 9(9):1553–1564
28. Shen H, Li F, Wu ZG, Park JH, Sreeram V (2018) Fuzzy-model-based nonfragile control for nonlinear singularly perturbed systems with semi-Markov jump parameters. IEEE Trans Fuzzy Syst 26(6):3428–3439
29. Shen H, Men YZ, Wu ZG, Cao JD, Lu GP (2019) Network-based quantized control for fuzzy singularly perturbed semi-Markov jump systems and its application. IEEE Trans Circuits Syst I Regul Pap 66(3):1130–1140
30. Shen H, Li F, Yan HC, Karimi HR, Lam HK (2018) Finite-time event-triggered H_∞ control for T-S fuzzy Markov jump systems. IEEE Trans Fuzzy Syst 26(5):3122–3135
31. Fan Q, Wang Z, Li DD, Gao DQ, Zha HY (2017) Entropy-based fuzzy support vector machine for imbalance datasets. Knowl Based Syst 115:87–99
32. Gupta D, Richhariya B, Borah P (2018) A fuzzy twin support vector machine based on information entropy for class imbalance learning. Neural Comput Appl. https://doi.org/10.1007/s00521-018-3551-9
33. Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9(3):293–300
34. Shannon C (2001) A mathematical theory of communication. ACM SIGMOBILE Mobile Comput Commun 5(1):3–55
35. Benkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7(11):2399–2434
36. Chen SG, Wu XJ, Xu J (2016) Locality preserving projection twin support vector machine and its application in classification. J Algorithms Comput Technol 10(2):65–72
37. Ripley BD (2008) Pattern recognition and neural networks. Cambridge University Press, Cambridge
38. Muphy PM, Aha DW (1992) UCI repository of machine learning databases. University of California, Irvine. http://www.ics.uci.edu/~mlearn. Accessed 10 Jan 2018
39. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
40. Iman RL, Davenport JM (1980) Approximations of the critical region of the Friedman statistic. Commun Stat Theory Methods 9(6):571–595
41. The USPS Database. http://www.cs.nyu.edu/roweis/data.html. Accessed 15 Nov 2018
42. Nene SA, Nayar SK, Murase H (1996) Columbia object image library (COIL-20). Technical report CUCS-005096, February
43. Martinez AM, Benavente R (1998) The AR face database. CVC technical report #24, June
44. Musicant DR (1998) NDC: Normally distributed clustered datasets. Computer Science Department, University of Wisconsin, Madison, USA. http://research.cs.wisc.edu/dmi/svm/ndc/. Accessed 16 Feb 2018
45. Golub GH, Van Loan CF (2012) Matrix computations, vol 3. JHU Press, Baltimore
46. Nasiri JA, Charkari NM, Jalili S (2015) Least squares twin multi-class classification support vector machine. Pattern Recognit 48(3):984–992
47. Chen SG, Wu XJ (2017) Multiple birth least squares support vector machine for multi-class classification. Int J Mach Learn Cybern 8(6):1731–1742
48. Tang JJ, Li DW, Tian YJ, Liu DL (2018) Multi-view learning based on nonparallel support vector machine. Knowl Based Syst 158:94–108

49. Houthuys L, Langone R, Suykens JAK (2018) Multi-view least squares support vector machines classification. Neurocomputing 282:78–88
50. Qi ZQ, Tian YJ, Shi Y (2012) Laplacian twin support vector machine for semi-supervised classification. Neural Netw 35:46–53
51. Chen WJ, Shao YH, Deng NY, Feng ZL (2014) Laplacian least squares twin support vector machine for semi-supervised classification. Neurocomputing 145:465–476