Int. J. Mach. Learn. & Cyber. (2018) 9:1553–1564
DOI 10.1007/s13042-017-0664-x

ORIGINAL ARTICLE

A new fuzzy twin support vector machine for pattern classification

Su-Gen Chen[1,2] · Xiao-Jun Wu[1]

Received: 28 August 2015 / Accepted: 13 March 2017 / Published online: 6 April 2017

Abstract Fuzzy SVM is often used to solve the problem that patterns belonging to one class often play more significant roles in classification. In order to improve the efficiency and performance of fuzzy SVM, this paper proposes a new fuzzy twin support vector machine (NFTSVM) for binary classification, in which fuzzy neural networks and twin support vector machine (TWSVM) are incorporated. By design, the influence of the samples with high uncertainty can be mitigated by employing fuzzy membership to weigh the margin of each training sample, which improves the generalization ability. In addition, we show that the existing TWSVM and twin bounded support vector machines (TBSVM) are special cases of the proposed NFTSVM when the parameters of NFTSVM are appropriately selected. Moreover, the successive overrelaxation (SOR) technique is adopted to solve the quadratic programming problems (QPPs) in the proposed NFTSVM algorithm to speed up the training procedure. Experimental results obtained on several artificial and real-world datasets validate the feasibility and effectiveness of the proposed method.

✉ Xiao-Jun Wu
xiaojun_wu_jnu@163.com

Su-Gen Chen
chensugen@126.com

1 School of IoT Engineering, Jiangnan University, Wuxi 214122, People's Republic of China

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

Keywords Pattern classification · Twin support vector machine · Fuzzy support vector machine · Successive overrelaxation technique

1 Introduction

Support vector machine (SVM) [1, 2], as a powerful tool for classification and regression, has been widely used in a variety of practical applications [3–6]. However, SVM is computational costly because its solution follows from solving constrained quadratic programming problems (QPPs), especially when handing large-scale data. Furthermore, SVM has only one separating plane hence it cannot cope with the complex XOR problem. To address these issues, the generalized eigenvalue proximal SVM (GEPSVM) [7] uses two nonparallel haperplanes such that each hyperplane is close to one of the two classes and is as far as possible from the other class. Motivated by GEPSVM, Jayadeva et al. [8] proposed a twin SVM (TWSVM) that developed a novel nonparallel hyperplane classifier for binary classification. In TWSVM, training samples in each class are proximal to one of the two nonparallel hyperplanes. The nonparallel hyperplanes are obtained by solving a pair of small-sized QPPs. Experimental results in [8] validated the superiority of TWSVM over the classical SVM. Moreover, TWSVM can effectively deal with the XOR problem due to the underlying model. Therefore, the method of constructing nonparallel hyperplane SVM has been extensively studied and a variety of methods have been proposed [9–16], such as LSTSVM [9], TBSVM [10], TPMSVM [11], RTSVM [12], PPSVC [13], NHSVM [14] and NPSVM [15]. Also, the method of finding two projection directions has been widely investigated, such as MVSVM [17], PTSVM [18] and LSPTSVM [19, 20]. An overview on twin support vector machines was given in [21].

However, in practice, patterns belonging to one class often play more significant roles in classification. Such a problem is usually solved using fuzzy SVMs [22–26]. In fuzzy SVMs, the patterns of more important classes are assigned higher membership values. Different from FSVM, Tao and Wang proposed a new framework for SVM to solve the fuzzy classification problem using fuzzy membership function [27], which termed new fuzzy SVM (NFSVM) [28]. However, FSVM and NFSVM have the same computational complexity because their solutions follow from solving QPP. Lastly, inspired by TWSVM, FTWSVM [29] and FTSVM [30] have also been proposed. Specifically, FTWSVM incorporates the information of fuzziness in the data and obtains nonparallel planes around which the data points of the corresponding class get clustered. However, based on v-twin support vector machine, FTSVM was proposed by introducing importance of training sample.

In order to enhance the efficiency and performance of fuzzy SVM and TWSVM for binary classification, in this paper, we propose a new fuzzy twin support vector machine (NFTSVM). The proposed NFTSVM relates to the ideas of both NFSVM and TWSVM. However, NFTSVM has two advantages compared to NFSVM and TWSVM. First, NFTSVM aims at generating two nonparallel hyperplanes such that each plane is close to one of the two classes and is as far as possible from the other. It employs the fuzzy membership to weigh the margin of each training sample, hence enhances the generalization ability. Second, NFTSVM has a more general formulation, compared to TWSVM and TBSVM. In fact, TWSVM and TBSVM are special cases of the proposed NFTSVM. Moreover, our proposed NFTSVM is different from FTWSVM and FTSVM. For FTWSVM and FTSVM, a fuzzy membership value is assigned to each pattern, and points are classified by assigning them to the nearest of two nonparallel planes that are close to their respective classes. However, inspired by NFSVM, the main idea of our proposed NFTSVM is to weigh the margin by using the fuzzy membership function. Thus, it is easy to think that the influence of the samples with high uncertainty can be decreased by weighing the margin of each training vector. In addition, one of the important privileges of NFTSVM by the idea from fuzzy neural networks is that we can employ some available fuzzy membership functions.

The rest of this paper is organized as follows. Section 2 gives a brief overview to related work, including NFSVM, TWSVM and TBSVM. Section 3 presents the proposed NFTSVM in detail. Experimental results and conclusion are given in Sects. 4 and 5, respectively.

2 Related work

Conventional SVMs solve the binary classification problem using the principle of structural risk minimization (SRM) [1, 2]. For a typical binary classification problem, we have a set of training samples:

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\} \tag{1}$$

where $x_i \in \mathfrak{R}^n$ is the i-th observation, $y_i \in \{-1, 1\}$ is the label of the i-th observation, and l is the number of training samples. The aim of linear SVM is to solve the following primal QPP:

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i(w^T x_i + b) \geqslant 1 - \xi_i, \\ & \xi_i \geqslant 0,\ i = 1, 2, \dots, l \end{aligned} \tag{2}$$

where C is penalty parameter and ξ_i are slack variables. The output of SVM is an optimal separating hyperplane:

$$w^T x + b = 0 \tag{3}$$

where $w \in \mathfrak{R}^n$ is the normal vector to the hyperplane, and $b \in \mathfrak{R}^1$ is the offset. Given a new observation x', we can use this hyperplane to determine the label, i.e. we assign the positive label to x' if $w^T x + b \geqslant 0$, otherwise, we assign the negative label to x'.

2.1 New fuzzy SVM (NFSVM)

However, one of the main drawbacks of the conventional SVM is that the linear model is very sensitive to outliers or noises. In order to deal with this problem, fuzzy support vector machines (FSVM) have been proposed [22–26]. In addition, Tao and Wang [28] proposed a new fuzzy support vector machine (NFSVM). In NFSVM, we consider the binary classification problem for a set of training samples:

$$T^* = \{(x_1, m_1), (x_2, m_2), \dots, (x_l, m_l)\} \tag{4}$$

where $x_i \in \mathfrak{R}^n$ is independently distributed, $m_i \in [0, 1$ is the fuzzy membership that measures the contribution of the i-th observation x_i to the positive class and l is the number of training samples. In order to get the similar formula to classical SVM, NFSVM uses a new label $y_i = 2m_i - 1$ to replace the fuzzy membership m_i. Hence, the primal problem of NFSVM can be expressed as

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}w^T w + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i(w^T x_i + b) \geqslant y_i^2 - y_i^2\xi_i, \\ & \xi_i \geqslant 0,\ i = 1, 2, \dots, l \end{aligned} \tag{5}$$

where C is penalty parameter, ξ_i are slack variables, $w \in \mathfrak{R}^n$ is the normal vector and $b \in \mathfrak{R}$ is the offset.

To solve this primal problem, NFSVM solves its Lagrangian dual problem:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{l}\alpha_i y_i^2 \\ s.t. \quad & \sum_{i=1}^{l}\alpha_i y_i = 0 \\ & 0 \leqslant \alpha_i y_i^2 \leqslant C, \ i = 1, 2, \ldots, l \end{aligned} \tag{6}$$

where $\alpha \in \mathfrak{R}^n$ is Lagrangian multiplier [28].

2.2 TWSVM

Different from the single hyperplane SVM discussed above, linear twin support vector machine (TWSVM) [8] aims to find a pair of nonparallel hyperplanes:

$$w_1^T x + b_1 = 0 \quad \text{and} \quad w_2^T x + b_2 = 0 \tag{7}$$

such that each hyperplane is proximal to the training samples of one class and is as far as possible from the samples of the other class. More specifically, we consider the training of a binary classifier from p positive samples and q negative samples. The training samples in positive class are denoted by matrix $A \in \mathfrak{R}^{p\times n}$ and the training samples in negative class are denoted by matrix $B \in \mathfrak{R}^{q\times n}$. The primal problem of linear TWSVM can be expressed as:

$$\begin{aligned} \min_{w_1,b_1,\xi_2} \quad & \frac{1}{2}||Aw_1 + e_1 b_1||_2^2 + c_1 e_2^T \xi_2 \\ s.t. \quad & -(Bw_1 + e_2 b_1) + \xi_2 \geqslant e_2, \xi_2 \geqslant 0 \end{aligned} \tag{8}$$

$$\begin{aligned} \min_{w_2,b_2,\xi_1} \quad & \frac{1}{2}||Bw_2 + e_2 b_2||_2^2 + c_2 e_1^T \xi_1 \\ s.t. \quad & (Aw_2 + e_1 b_2) + \xi_1 \geqslant e_1, \xi_1 \geqslant 0 \end{aligned} \tag{9}$$

where $c_1 > 0$ and $c_2 > 0$ are 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 introducing the method of Lagrangian multipliers, the corresponding Wolfe dual of QPPs (8) and (9) can be represented as

$$\begin{aligned} \max_{\alpha} \quad & e_2^T\alpha - \frac{1}{2}\alpha^T G(H^T H)^{-1} G^T \alpha \\ s.t. \quad & 0 \leqslant \alpha \leqslant c_1 e_2 \end{aligned} \tag{10}$$

$$\begin{aligned} \max_{\gamma} \quad & e_1^T\gamma - \frac{1}{2}\gamma^T H(G^T G)^{-1} H^T \gamma \\ s.t. \quad & 0 \leqslant \gamma \leqslant c_2 e_1 \end{aligned} \tag{11}$$

where $G = [B\ e_2$ and $\alpha \in \mathfrak{R}^q$, $\gamma \in \mathfrak{R}^p$ are Lagrangian multipliers [8].

2.3 TBSVM

In order to embody the marrow of statistical learning theory and improve the performance of classification, the twin bounded support vector machines (TBSVM) was proposed [10]. TBSVM minimizes the structural risk by adding a regularization term with the idea of maximizing one side margin. For a linear binary classification problem, TBSVM solves the following two primal problems:

$$\begin{aligned} \min_{w_1,b_1,\xi_2} \quad & \frac{1}{2}c_3(w_1^2 + b_1^2) + \frac{1}{2}||Aw_1 + e_1 b_1||_2^2 + c_1 e_2^T \xi_2 \\ s.t. \quad & -(Bw_1 + e_2 b_1) + \xi_2 \geqslant e_2, \xi_2 \geqslant 0 \end{aligned} \tag{12}$$

$$\begin{aligned} \min_{w_2,b_2,\xi_1} \quad & \frac{1}{2}c_4(w_2^2 + b_2^2) + \frac{1}{2}||Bw_2 + e_2 b_2||_2^2 + c_2 e_1^T \xi_1 \\ s.t. \quad & (Aw_2 + e_1 b_2) + \xi_1 \geqslant e_1, \xi_1 \geqslant 0 \end{aligned} \tag{13}$$

where c_1, c_2, c_3 and c_4 are positive penalty parameters, ξ_1, ξ_2 are slack variables, and e_1, e_2 are vectors with each element of the value of 1. By introducing the method of Lagrangian multipliers, the Wolfe dual of QPPs (12) and (13) can be represented as

$$\begin{aligned} \max_{\alpha} \quad & e_2^T\alpha - \frac{1}{2}\alpha^T G(H^T H + c_3 I)^{-1} G^T \alpha \\ s.t. \quad & 0 \leqslant \alpha \leqslant c_1 e_2 \end{aligned} \tag{14}$$

$$\begin{aligned} \max_{\gamma} \quad & e_1^T\gamma - \frac{1}{2}\gamma^T H(G^T G + c_4 I)^{-1} H^T \gamma \\ s.t. \quad & 0 \leqslant \gamma \leqslant c_2 e_1 \end{aligned} \tag{15}$$

where $G = [B\ e_2]$, $H = [A\ e_1]$, $\alpha \in \mathfrak{R}^q$, $\gamma \in \mathfrak{R}^p$ are Lagrangian multipliers, and I is an identity matrix of appropriate dimensions [10].

3 New fuzzy twin support vector machine (NFTSVM)

3.1 Linear NFTSVM

In real-world applications, we may often face with the situations where patterns belonging to one class play a more significant role in classification. In such cases, the main concern is how to determine the final classes by assigning different importance degrees to training data. A well approach to deal with this challenge is to use the concept of fuzzy function. As we know, the existing nonparallel hyperplane support vector machines often obtain higher solving efficiency than conventional SVM, such as TWSVM is approximately four times faster than conventional SVM. In order to deal with real-world applications and get higher solving efficiency, we incorporate the concept of fuzzy theory into NFSVM and propose a new fuzzy twin support vector machine (NFTSVM) for binary classification. As discussed in Sect. 2.1, in fuzzy binary classification, m_i is the fuzzy membership that measures the membership of the corresponding observation x_i to the positive class. Given p positive samples $\{(\tilde{x}_1,\tilde{m}_1),(\tilde{x}_2,\tilde{m}_2),\dots,(\tilde{x}_p,\tilde{m}_p)\}$ and q negative samples$\{(\hat{x}_1,\hat{m}_1),(\hat{x}_q,\hat{m}_q),\dots,(\hat{x}_q,\hat{m}_q)\}$, then, we have the diagonal matrices $Y_1 = diag(\tilde{y}_1,\tilde{y}_2,\dots,\tilde{y}_p)$, $Y_2 = diag(\hat{y}_1,\hat{y}_2,\dots,\hat{y}_q)$, where $\tilde{y}_i = 2\tilde{m}_i - 1,\ (i=1,2,\dots,p)$, $\hat{y}_j = 2\hat{m}_j - 1,\ (j=1,\dots,q)$, respectively.

Similar to NFSVM [28], TWSVM [8] and TBSVM [10], the goal of the proposed linear NFTSVM is to find two nonparallel hyperplanes (7) using the following primal problems:

$$\begin{aligned}\min_{w_1,b_1,\xi_2}\ & \frac{1}{2}||Aw_1+e_1b_1||_2^2+\frac{1}{2}c_1(w_1^2+b_1^2)+c_2e_2^T\xi_2 \\ s.t.\quad & Y_2(Bw_1+e_2b_1)\geqslant Y_2^2e_2-Y_2^2\xi_2,\xi_2\geqslant 0\end{aligned} \tag{16}$$

$$\begin{aligned}\min_{w_2,b_2,\xi_1}\ & \frac{1}{2}||Bw_2+e_2b_2||_2^2+\frac{1}{2}c_3(w_2^2+b_2^2)+c_4e_1^T\xi_1 \\ s.t.\quad & Y_1(Aw_2+e_1b_2)\geqslant Y_1^2e_1-Y_1^2\xi_1,\xi_1\geqslant 0\end{aligned} \tag{17}$$

where c_1, c_2, c_3 and c_4 are positive penalty parameters, ξ_1 and ξ_2 are slack variables, e_1 and e_2 are vectors with each entry of the value of 1.

Let us compare the objective functions and constraints in (12), (13) and (16), (17). Obviously, their objective functions are the same and their constraints are different. In fact, for a classical binary classification problem, the fuzzy membership $\tilde{m}_i = 1$ and $\hat{m}_j = 0$, and the label $\tilde{y}_i = 1,\ (i=1,2,\dots,p)$ and $\hat{y}_j = -1,\ (j=1,\dots,q)$. In this case, the constraints in (16), (17) are the same as in (12), (13). That is to say, TBSVM is a special case of our NFTSVM when the parameters of NFTSVM are appropriately selected. However, in practical problem, the fuzzy memberships $\tilde{m}_i\ (i=1,2,\dots,p)$ are not all 1 and $\hat{m}_j\ (j=1,2,\dots,q)$ are not all 0. This means that the punishment for different samples with different coefficients, which maybe improve the classification accuracy in practical problems.

To obtain the solutions of the objective functions (16) and (17), we have to derive their dual problems. For (16), by introducing the method of Lagrangian multipliers, we can obtain the Lagrangian function:

$$\begin{aligned}L(w_1,b_1,\xi_2,\alpha,\beta)=&\frac{1}{2}||Aw_1+e_1b_1||_2^2+\frac{1}{2}c_1(w_1^2+b_1^2)+c_2e_2^T\xi_2\\&-\alpha^T[Y_2(Bw_1+e_2b_1)-Y_2^2e_2+Y_2^2\xi_2\end{aligned} \tag{18}$$

where $\alpha\in\Re^q$ and $\beta\in\Re^q$ are Lagrangian multipliers. With the Karush-Kuhn- Tucker (KKT) conditions [31], we have:

$$\frac{\partial L}{\partial w_1}=A^T(Aw_1+e_1b_1)+c_1w_1-B^TY_2^T\alpha=0, \tag{19}$$

$$\frac{\partial L}{\partial b_1}=e_1^T(Aw_1+e_1b_1)+c_1b_1-e_2^TY_2^T\alpha=0, \tag{20}$$

$$\frac{\partial L}{\partial \xi_2}=c_2e_2-Y_2^T\alpha-\beta=0, \tag{21}$$

$$Y_2(Bw_1+e_2b_1)\geqslant Y_2^Te_2-Y_2^T\xi_2,\xi_2\geqslant 0, \tag{22}$$

$$\alpha^T[Y_2(Bw_1+e_2b_1)-Y_2^Te_2+Y_2^T\xi_2 \tag{23}$$

$$\alpha\geqslant 0,\beta\geqslant 0. \tag{24}$$

Since $\beta\geqslant 0$, from (21) and (24), we have:

$$0\leqslant\alpha\leqslant c_2(Y_2^2)^{-1}e_2 \tag{25}$$

Let $G=[B\ e_2]$, $G=[H\ A\ e_1]$, $v_1=[w_1^Tb_1]^T$, (19) and (20) imply:

$$(H^TH+c_1I)v_1-G^TY_2^T\alpha=0 \tag{26}$$

Thus, we can get the augmented vector

$$v_1=(H^TH+c_1I)^{-1}G^TY_2^T\alpha \tag{27}$$

where I is an identity matrix. Then, we obtain the Wolfe dual problem of (16) by combing (27) with (18) using (19), (21):

$$\begin{aligned}\max_{\alpha}\ & e_2^TY_2^T\alpha-\frac{1}{2}\alpha^TY_2G(H^TH+c_1I)^{-1}G^TY_2^T\alpha\\ s.t.\quad & 0\leqslant\alpha\leqslant c_2(Y_2^2)^{-1}e_2\end{aligned} \tag{28}$$

In the same fashion, we can obtain the Wolfe dual problem of (19):

$$\begin{aligned}\max_{\gamma}\ & e_1^TY_1^T\gamma-\frac{1}{2}\gamma^TY_1H(G^TG+c_3I)^{-1}H^TY_1^T\gamma\\ s.t.\quad & 0\leqslant\gamma\leqslant c_4(Y_1^2)^{-1}e_1\end{aligned} \tag{29}$$

where $\alpha\in\Re^q,\gamma\in\Re^p$ are Lagrangian multipliers.

The nonparallel hyperplanes (7) can be obtained from the solutions of α and γ in (28) and (29) by:

$$v_1=(H^TH+c_1I)^{-1}G^TY_2^T\alpha,\quad \text{where } v_1=[w_1^T\ b_1]^T \tag{30}$$

$$v_2=(G^TG+c_3I)^{-1}H^TY_1^T\gamma,\quad \text{where } v_2=[w_2^T\ b_2]^T \tag{31}$$

Once v_1 and v_2 are obtained, these two nonparallel hyperplanes (7) are known. A new sample $x\in\Re^n$ is assigned to either positive or negative label, depending on the hyperplanes (7) it lies closest to, i.e.

$$x\in W_k,\ k=\arg\min_{i=1,2}\left\{\frac{|w_1^Tx+b_1|}{||w_1||},\frac{|w_2^Tx+b_2|}{||w_2||}\right\} \tag{32}$$

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

3.2 Nonlinear NFTSVM

In this subsection, we show that our linear NFTSVM can be easily extended to nonlinear case. Here, we consider the following kernel-generated hyperplanes

$$K(x^T, C^T)w_1 + b_1 = 0 \quad \text{and} \quad K(x^T, C^T)w_2 + b_2 = 0, \tag{33}$$

where $C = [A\ B] \in \Re^{(p+q)\times n}$ and K is an appropriately kernel. Similar to the linear case, the nonlinear optimization problems can be expressed as:

$$\begin{aligned} \min_{w_1,b_1,\eta_2}\ & \frac{1}{2}||K(A,C^T)w_1 + e_1b_1||_2^2 + \frac{1}{2}c_1(w_1^2 + b_1^2) + c_2e_2^T\eta_2 \\ s.t.\ \ & Y_2(K(B,C^T)w_1 + e_2b_1) \geqslant Y_2^2e_2 - Y_2^2\eta_2, \eta_2 \geqslant 0, \end{aligned} \tag{34}$$

$$\begin{aligned} \min_{w_2,b_2,\eta_1}\ & \frac{1}{2}||K(B,C^T)w_2 + e_2b_2||_2^2 + \frac{1}{2}c_3(w_2^2 + b_2^2) + c_4e_1^T\eta_1 \\ s.t.\ \ & Y_1(K(A,C^T)w_2 + e_1b_2) \geqslant Y_1^2e_1 - Y_1^2\eta_1, \eta_1 \geqslant 0 \end{aligned} \tag{35}$$

With the Lagrangian method and KKT conditions, we can obtain the corresponding Wolfe dual problems

$$\begin{aligned} \max_{\alpha}\ & e_2^TY_2^T\alpha - \frac{1}{2}\alpha^TY_2\tilde{G}(\tilde{H}^T\tilde{H} + c_1I)^{-1}\tilde{G}^TY_2^T\alpha \\ s.t.\ \ & 0 \leqslant \alpha \leqslant c_2(Y_2^2)^{-1}e_2, \end{aligned} \tag{36}$$

$$\begin{aligned} \max_{\gamma}\ & e_1^TY_1^T\gamma - \frac{1}{2}\gamma^TY_1\tilde{H}(\tilde{G}^T\tilde{G} + c_3I)^{-1}\tilde{H}^TY_1^T\gamma \\ s.t.\ \ & 0 \leqslant \gamma \leqslant c_4(Y_1^2)^{-1}e_1, \end{aligned} \tag{37}$$

where $\tilde{G} = [K(B, C^T)\ e_2]$, $\tilde{H} = [K(A, C^T)e_1]$

According to (34)–(37), the augmented vectors $v_1 = [w_1^T\ b_1]^T$ and $v_2 = [w_2^T\ b_2]^T$ can be obtained by

$$v_1 = (\tilde{H}^T\tilde{H} + c_1I)^{-1}\tilde{G}^TY_2^T\alpha, \tag{38}$$

$$v_2 = (\tilde{G}^T\tilde{G} + c_3I)^{-1}\tilde{H}^TY_1^T\gamma, \tag{39}$$

Once the vector v_1 and v_2 are obtained, the two nonparallel hyperplanes (33) are known. A new sample $x \in \Re^n$ is assigned to either positive or negative label, depending on which of the hyperplanes (33) it lies closest to, i.e.

$$x \in W_k,\ k = \arg\min_{i=1,2}\left\{\frac{|w_1^TK(x,C^T) + b_1|}{\sqrt{w_1^TK(C,C^T)w_1}}, \frac{|w_2^TK(x,C^T) + b_2|}{\sqrt{w_2^TK(C,C^T)w_2}}\right\} \tag{40}$$

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

3.3 Implementation

In this subsection, we discuss the implementation of our proposed NFTSVM. In our NFTSVM, the dual problem can be rewritten as the following unified form

$$\begin{aligned} \min_{\alpha}\ & \tfrac{1}{2}\alpha^TQ\alpha - d^T\alpha \\ s.t.\ \ & 0 \leqslant \alpha \leqslant ce\ , \end{aligned} \tag{41}$$

where Q is a positive definite matrix and d is a vector. For example, if we set $Q = Y_2G(H^TH + c_1I)^{-1}G^TY_2^T$, $d = Y_2e_2$ and $c = c_2(Y_2^2)^{-1}$, problem (41) becomes the problem in (28) and if we set $Q = Y_2\tilde{G}(\tilde{H}^T\tilde{H} + c_1I)^{-1}\tilde{G}^TY_2^T$, $d = Y_2e_2$ and $c = c_2(Y_2^2)^{-1}$, problem (41) becomes the problem in (36).

Algorithm 1. Successive overrelaxation (SOR) Algorithm

Step1. Select the parameter $t \in (0, 2)$ and the initial vector α^0, set $k = 0$;

Step2. For k=1,2,3,…, Compute α^{k+1} by

$$\alpha^{k+1} = (\alpha^k - t\cdot D^{-1}(Q\alpha^k - d + L(\alpha^{k+1} - \alpha^k)))_\#$$

where Q is the matrix in (41). $(u)_\#$ denotes a piecewise function

$$(u_i)_\# = \begin{cases} 0,\ if\ u_i \le 0 \\ u_i,\ if\ 0 < u_i < c \\ c,\ if\ u_i \ge c \end{cases}$$

Define $L + D + L^T = Q$, where L is a strictly lower triangular matrix, Q is a diagonal matrix in (41);

Step3. Repeat Step 2, until $\| \alpha^{k+1} - \alpha^k \| < \varepsilon$, where ε is a pre-defined tolerance.

In our proposed NFTSVM, the most expensive part in terms of computational cost is solving the dual QPPs (41). To solve the QPPs more efficiently, we use a state- of-the-art optimization technique, i.e. successive overrelaxation (SOR) algorithm [10, 14, 32]. SOR is an excellent QPPs solver because it is able to deal with large scale problems, without storing all the data in memory [32]. The experimental results in the following section show that the SOR accelerates our proposed NFTSVM.

To have a further analysis and understanding, we give some remarks to our NFTSVM. Firstly, if we define the membership $m_i = 1$ when x_i is a positive sample, the corresponding label $y_i = 2m_i - 1 = 1$. If we define the membership $m_i = 0$ when x_i is a negative sample, the corresponding label $y_i = 2m_i - 1 = -1$. Therefore, $Y_1 = diag\{1, 1, \dots, 1\}$, $Y_2 = diag\{-1, -1, \dots, -1\}$ and the primal problems of NFTSVM can be described as

$$\begin{aligned} \min_{w_1,b_1,\xi_2}\ & \frac{1}{2}||Aw_1 + e_1b_1||_2^2 + \frac{1}{2}c_1(w_1^2 + b_1^2) + c_2e_2^T\xi_2 \\ s.t.\ \ & -(Bw_1 + e_2b_1) \geqslant e_2 - \xi_2, \xi_2 \geqslant 0, \end{aligned} \tag{42}$$

$$\begin{aligned} \min_{w_2,b_2,\xi_1}\ & \frac{1}{2}||Bw_2 + e_2b_2||_2^2 + \frac{1}{2}c_3(w_2^2 + b_2^2) + c_4e_1^T\xi_1 \\ s.t.\ \ & (Aw_2 + e_1b_2) \geqslant e_1 - \xi_1, \xi_1 \geqslant 0, \end{aligned} \tag{43}$$

It is obvious that (42) and (43) are the same as (12) and (13). Moreover, if the parameters $c_1 = c_3 = 0$, (42) and (43) degenerate to (8) and (9). Thus, the classical TWSVM [8] and TBSVM [10] are special cases of our NFTSVM when the parameters of NFTSVM are appropriately selected. Secondly, for nonlinear NFTSVM, if the number of training points is large, the rectangular kernel technique [33] can be applied to reduce the dimensionality of (36) and (37). Lastly, in practical classification applications, we often only have the information of labels and have to define the fuzzy membership based on prior knowledge. Fortunately, we can employ some available fuzzy membership functions from fuzzy neural networks. In this paper, we directly use a fuzzy membership function that is also used in [27, 28]. Keller and Hunt [27] suggested the following way to assign fuzzy membership values such that a fuzzy 2-partition is formed. Given a positive sample x_i with positive label +1, the fuzzy membership is defined as

$$m_1(x_i) = 0.5 + \frac{\exp(C_0(d_{-1}(x_i) - d_1(x_i))/d) - \exp(-C_0)}{2(\exp C_0 - \exp(-C_0))},$$
$$m_{-1}(x_i) = 1 - m_1(x_i)$$

Given a negative sample x_i with negative label −1, the fuzzy membership is defined as

$$m_{-1}(x_i) = 0.5 + \frac{\exp(C_0(d_1(x_i) - d_{-1}(x_i))/d) - \exp(-C_0)}{2(\exp C_0 - \exp(-C_0))},$$
$$m_1(x_i) = 1 - m_{-1}(x_i)$$

where $d_1(x_i)$ is the distance between x_i and the mean of the positive class, $d_{-1}(x_i)$ is the distance between x_i and the mean of the negative class, d is the distance between the means of positive and negative classes, and C_0 is a constant controlling the membership function.

4 Experimental results

In order to evaluate the performance of our proposed NFTSVM, we investigated its classification accuracy and computational efficiency both on artificial datasets and real-world benchmark datasets. In our experiments, we focused on the comparison between our proposed NFTSVM and some classical classifiers, e.g. NFSVM [28], TWSVM [8], TBSVM [10], FTWSVM [29] and FTSVM [30]. These methods were implemented in MATLAB R2013a on a PC with Intel (R) Core (TM) processor (3.40 GHz) CPU and 4 GB RAM. TWSVM, TBSVM, and our proposed NFTSVM were solved using SOR algorithm. The QPPs in NFSVM, FTWSVM and FTSVM were solved by the optimization toolbox QP in MATLAB. The "Accuracy" used in our experiments was defined as Accuracy = (TP + TN)/(TP + FP + TN + FN), where TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively. The parameters were selected by employing the standard tenfold cross-validation methodology [31]. The parameters c_i and Gaussian kernel width σ are selected from the set $\{2^i | i = -8, \ldots, 8\}$ and C_0 is selected from the set $\{0.5, 1, 1.5, 2, 2.5\}$.

4.1 Artificial datasets

In this subsection, three artificial datasets, including crossplane (XOR), complex XOR and Ripley's synthetic datasets [34] have been used to validate our proposed NFTSVM in dealing with linearly inseparable problems. Ripley's synthetic dataset contains 250 training samples and 1000 test samples. The average results of linear NFSVM, TWSVM, TBSVM and our proposed NFTSVM on crossplane (XOR) dataset and complex XOR dataset are demonstrated in Table 1. The average results of nonlinear NFSVM, TWSVM, TBSVM, and our NFTSVM with Gaussian kernels on Ripley's synthetic dataset are presented in Table 2. In all tables, "Acc" and "Std" denote "Accuracy" and "Standard deviation" respectively.

From Table 1, we can conclude that NFSVM cannot well deal with XOR and complex XOR problems. Although TWSVM and TBSVM have better

Table 2 Classification accuracy (training and testing) on Ripley datasets

Datasets	NFSVM Acc ± Std(%) Acc (%)	TWSVM Acc ± Std(%) Acc (%)	TBSVM Acc ± Std(%) Acc (%)	NFTSVM Acc ± Std(%) Acc (%)
Ripley 1000 × 2	86.80 ± 7.07 87.20	**88.40 ± 4.79** 85.40	**88.40 ± 4.27** 85.80	86.40 ± 8.68 **87.60**

Best accuracy values are in bold

Table 1 Classification accuracy (training and testing) on crossplane (XOR) datasets

Datasets	NFSVM Acc ± Std(%) Acc (%)	TWSVM Acc ± Std(%) Acc (%)	TBSVM Acc ± Std(%) Acc (%)	NFTSVM Acc ± Std(%) Acc (%)
XOR 200 × 2	65.00 ± 6.91 70.50	98.50 ± 2.42 **98.00**	**99.00 ± 2.11** **98.00**	**99.00 ± 3.16** 97.50
Complex XOR 260 × 2	64.23 ± 6.22 70.00	91.54 ± 5.68 90.77	91.15 ± 5.14 90.38	**95.77 ± 2.84** **95.00**

Best accuracy values are in bold

performance on XOR dataset, the performance decreases on the complex XOR dataset. In contrast, our proposed NFTSVM performs best both on the XOR and complex XOR datasets. According to Table 2, the accuracy of our proposed NFTSVM is 87.60% that beats all the others, which demonstrates that our method has better

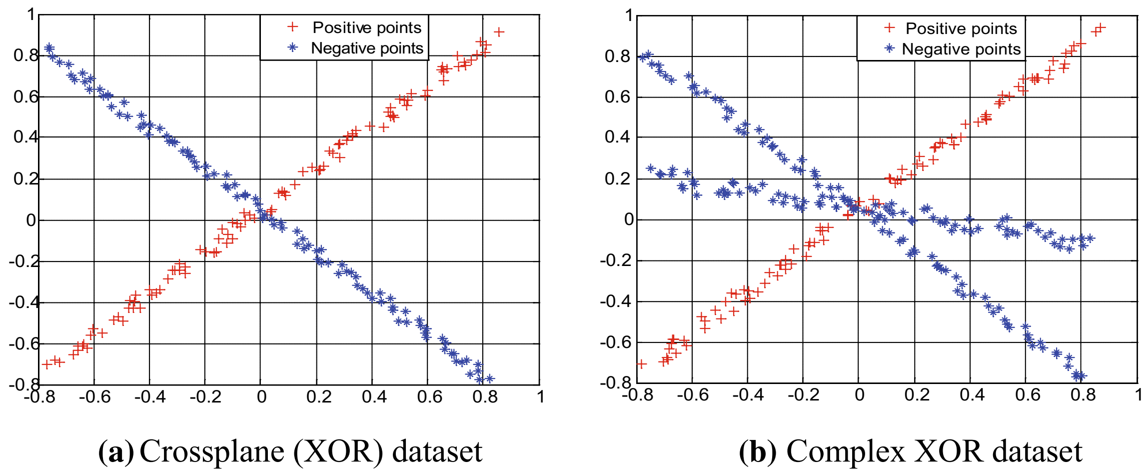

Fig. 1 Crossplane (**a**) and complex XOR (**b**) datasets

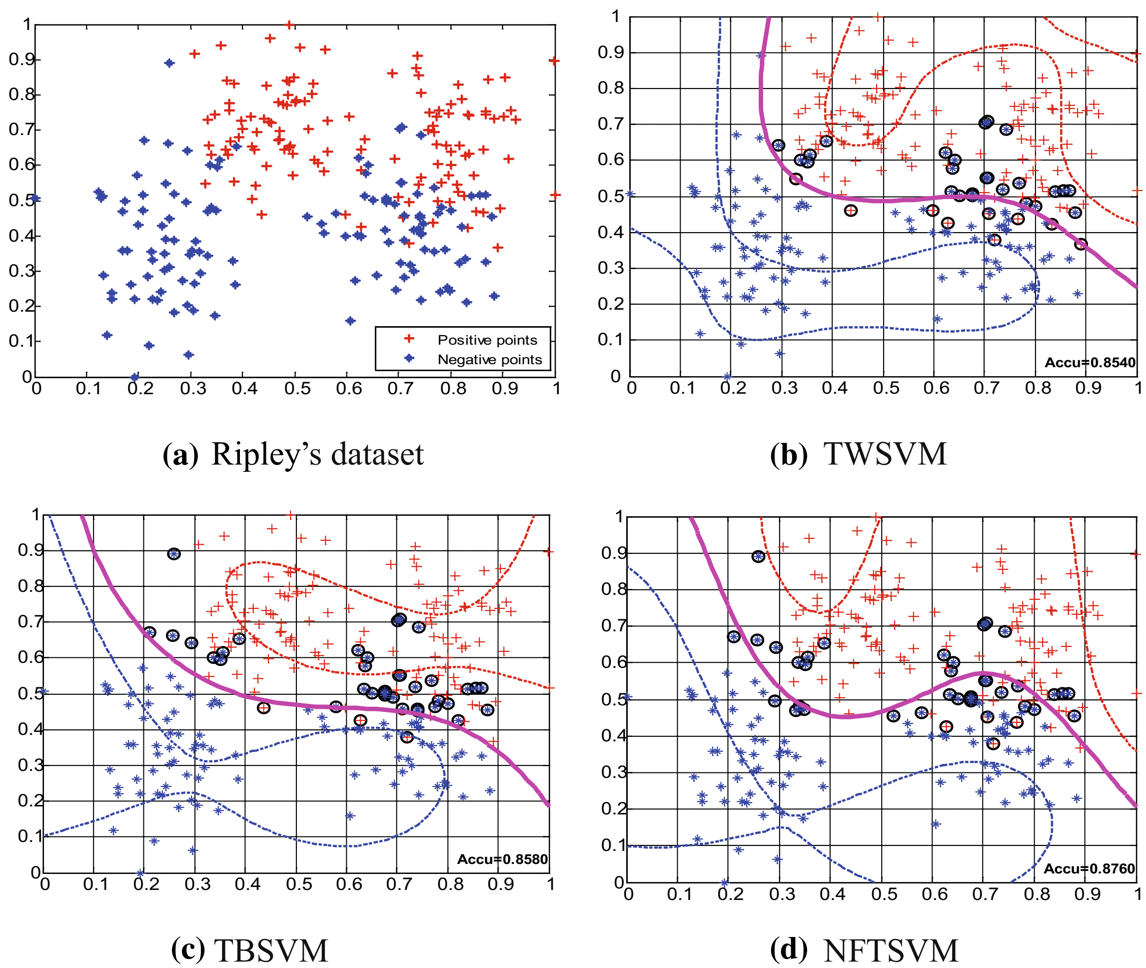

Fig. 2 Ripley's dataset: (**a**) the distribution of the data; (**b**) the results of TWSVM; (**c**) the results of TBSVM; and (**d**) the results of NFTSVM

generalization ability. Figure 1 illustrates the crossplane (XOR) and complex XOR datasets. Ripley's dataset and the hyperplanes of TWSVM, TBSVM and our proposed NFTSVM are shown in Fig. 2.

4.2 UCI datasets

To further compare our proposed NFTSVM with NFSVM, TWSVM, TBSVM, FTWSVM and FTSVM, we selected 15 datasets from the UCI machine learning repository [35]. The numerical results of their linear versions are given in Table 3. In Table 3, the best results are highlighted in bold font. We can find that the accuracy of linear NFTSVM is better than that of NFSVM, TWSVM, TBSVM, FTWSVM and FTSVM on most of the datasets. A win-tie-loss (W-T-L) summarization based on mean accuracy is also reported at the bottom of Table 3. For example, for the Heart-Statlog dataset, the accuracy of our NFTSVM is 86.30%, while that of NFSVM is 84.07%, TWSVM is 84.81%, TBSVM is 85.19%, FTWSVM is 84.85% and FTSVM is 84.81%.

Table 4 shows the experimental results of nonlinear NFSVM, TWSVM, TBSVM, FTWSVM, FTSVM and our NFTSVM on the selected UCI datasets. Note that the Gaussian kernel $K(x,y)=e^{-||x-y||^2/2\sigma^2}$ was used in this experiment. The accuracy and the time of training of these methods are also reported. The results in Table 4 are similar to those in Table 3, and therefore confirm our conclusion again. In addition, from Tables 3 and 4, we

Table 3 Tenfold testing percentage accuracy of linear classifiers

Datasets	NFSVM Acc ± Std(%) Time(s)	TWSVM Acc ± Std(%) Time(s)	TBSVM Acc ± Std(%) Time(s)	FTWSVM Acc ± Std(%) Time(s)	FTSVM Acc ± Std(%) Time(s)	NFTSVM Acc ± Std(%) Time(s)
Australian (690 × 14)	85.51 ± 4.16 2.5451	86.81 ± 4.07 0.3891	86.96 ± 2.65 0.4269	86.96 ± 2.39 0.5079	86.81 ± 3.71 0.6748	**87.25 ± 3.19** 0.3143
BUPA liver (345 × 6)	67.28 ± 9.33 1.1243	69.53 ± 6.70 0.0630	69.53 ± 9.45 0.0338	70.04 ± 7.25 0.2683	69.67 ± 6.48 0.3623	**71.81 ± 7.54** 0.1584
House votes (435 × 16)	94.04 ± 3.23 0.8110	95.18 ± 5.11 0.0754	95.64 ± 2.96 0.0569	95.82 ± 2.90 0.3019	95.62 ± 2.32 0.4107	**95.87 ± 2.37** 0.1411
Heart-c (303 × 14)	84.78 ± 5.04 0.9173	85.17 ± 7.45 0.0210	**85.86 ± 6.45** 0.0303	85.16 ± 7.19 0.1216	84.84 ± 3.26 0.2623	85.48 ± 3.17 0.0152
Heart-Statlog (270 × 13)	84.07 ± 9.25 0.4829	84.81 ± 4.77 0.0332	85.19 ± 4.28 0.0456	85.25 ± 3.24 0.0729	85.04 ± 4.36 0.1181	**86.30 ± 6.06** 0.0263
Ionosphere (351 × 34)	86.36 ± 5.99 0.7604	91.18 ± 5.42 0.0385	91.16 ± 4.15 0.0183	**91.74 ± 4.75** 0.2476	90.60 ± 4.87 0.3834	91.47 ± 3.74 0.1237
Monk3 (432 × 7)	80.11 ± 6.14 0.4887	**87.49 ± 7.42** 0.0460	81.71 ± 6.25 0.0697	82.14 ± 4.21 0.3021	82.08 ± 3.27 0.4082	82.42 ± 4.61 0.0690
Musk (476 × 166)	78.98 ± 4.49 2.5894	85.91 ± 3.89 0.2313	86.55 ± 3.48 0.1701	86.45 ± 5.42 0.6587	86.02 ± 6.35 0.7085	**86.56 ± 5.37** 0.5029
Sonar (208 × 60)	79.31 ± 5.59 0.2546	78.76 ± 7.22 0.0315	80.26 ± 9.50 0.0125	79.60 ± 7.25 0.0824	79.48 ± 6.82 0.1502	**81.26 ± 7.96** 0.0667
Spect (267 × 44)	**83.18 ± 7.22** 0.2361	80.19 ± 6.97 0.0632	81.28 ± 7.24 0.0293	80.92 ± 3.23 0.2338	80.63 ± 3.75 0.2932	81.70 ± 7.40 0.0680
Wpbc (198 × 34)	72.18 ± 6.56 0.9140	82.89 ± 5.23 0.1867	**82.95 ± 8.48** 0.0668	82.12 ± 6.25 0.1608	82.84 ± 4.52 0.2418	82.29 ± 6.91 0.0831
German (1000 × 20)	73.50 ± 5.38 16.4945	74.90 ± 2.32 1.0132	75.90 ± 2.18 0.9183	75.48 ± 4.26 5.6823	75.67 ± 3.72 5.4905	**76.20 ± 4.16** 0.8182
CMC (1473 × 9)	76.16 ± 3.39 23.3055	77.24 ± 2.19 1.6267	**77.40 ± 2.00** 1.4657	77.28 ± 3.54 7.3454	77.26 ± 3.46 7.9850	**77.40 ± 3.72** 1.2031
Hypothyroid (3163 × 25)	96.38 ± 1.35 51.2482	97.33 ± 1.19 5.5438	**97.47 ± 1.72** 5.1408	97.33 ± 1.27 23.4976	97.35 ± 1.19 23.6237	97.44 ± 0.67 4.3745
Spambase (4601 × 57)	92.45 ± 1.42 61.6878	92.12 ± 1.19 8.6856	92.32 ± 1.35 8.3635	**92.85 ± 1.45** 31.6878	92.48 ± 0.89 32.2624	92.65 ± 1.05 7.2252
W-T-L	14-0-1	13-0-2	11-1-3	13-0-2	14-0-1	

Best accuracy values are in bold

Table 4 Tenfold testing percentage accuracy of nonlinear classifiers

Datasets	NFSVM Acc ± Std(%) Time(s)	TWSVM Acc ± Std(%) Time(s)	TBSVM Acc ± Std(%) Time(s)	FTWSVM Acc ± Std(%) Time(s)	FTSVM Acc ± Std(%) Time(s)	NFTSVM Acc ± Std(%) Time(s)
Australian (690 × 14)	85.86 ± 3.25	86.96 ± 4.27	87.10 ± 3.83	86.96 ± 4.04	86.96 ± 3.47	**87.25 ± 4.26**
	3.2673	0.6897	1.6723	1.3443	2.0890	1.0293
BUPA liver (345 × 6)	65.23 ± 8.11	**74.82 ± 5.75**	73.55 ± 8.10	74.38 ± 6.28	73.86 ± 7.21	74.77 ± 6.88
	0.9612	0.2307	0.2391	1.0743	1.2702	0.8086
House votes (435 × 16)	92.86 ± 5.05	95.87 ± 3.17	95.86 ± 1.80	96.08 ± 2.63	95.63 ± 2.27	**96.31 ± 1.64**
	0.6754	0.2210	0.4401	0.6454	0.6854	0.4024
Heart-c (303 × 14)	83.17 ± 4.54	83.49 ± 4.15	**85.51 ± 3.76**	83.82 ± 4.79	84.86 ± 5.70	84.81 ± 6.65
	0.7326	0.1442	0.1916	0.3010	0.3292	0.1994
Heart-Statlog (270 × 13)	82.59 ± 8.91	82.59 ± 8.74	84.07 ± 6.54	84.96 ± 6.81	84.59 ± 4.95	**85.56 ± 6.16**
	0.6054	0.1130	0.1916	0.2707	0.3430	0.1607
Ionosphere (351 × 34)	93.45 ± 3.01	94.59 ± 2.84	95.72 ± 3.87	95.48 ± 3.25	95.14 ± 4.13	**96.29 ± 3.32**
	0.7326	0.1352	0.3562	0.4110	0.6332	0.2552
Monk3 (432 × 7)	90.51 ± 5.31	97.45 ± 1.72	97.47 ± 2.52	**98.16 ± 1.82**	97.91 ± 1.72	97.45 ± 1.72
	0.6492	0.1832	0.4620	0.5238	0.7669	0.4606
Musk (476 × 166)	81.95 ± 7.00	94.77 ± 3.83	95.17 ± 1.74	95.38 ± 3.69	95.21 ± 3.64	**95.81 ± 2.39**
	2.2624	0.6172	0.9081	1.2113	2.2113	0.8326
Sonar (208 × 60)	87.40 ± 8.79	90.88 ± 4.72	90.95 ± 9.10	90.85 ± 2.32	90.93 ± 2.56	**91.40 ± 4.90**
	0.6996	0.0706	0.1194	0.2579	0.2824	0.1033
Spect (267 × 44)	79.49 ± 7.52	82.75 ± 7.50	82.78 ± 6.08	**83.52 ± 7.65**	82.91 ± 6.32	82.09 ± 7.89
	0.9495	0.0823	0.1644	0.2395	0.2858	0.1691
Wpbc (198 × 34)	81.87 ± 7.52	82.87 ± 6.24	**82.34 ± 5.83**	81.86 ± 4.57	81.84 ± 6.07	81.87 ± 7.44
	0.7974	0.0995	0.1260	0.2252	0.5011	0.1261
German (1000 × 20)	76.16 ± 4.57	75.60 ± 1.36	76.30 ± 2.67	76.18 ± 2.16	76.26 ± 3.61	**76.40 ± 4.18**
	19.9993	1.1272	1.2073	6.9048	6.4612	1.0423
CMC (1473 × 9)	77.46 ± 2.32	77.82 ± 2.42	**78.32 ± 3.49**	77.94 ± 2.56	77.89 ± 3.85	78.05 ± 2.29
	25.2132	2.4304	2.4127	8.2132	8.0437	2.2367
Hypothyroid (3163 × 25)	97.15 ± 1.42	97.58 ± 1.18	97.72 ± 1.02	97.67 ± 1.18	97.74 ± 1.06	**97.86 ± 0.92**
	58.5326	7.1536	6.8230	27.5221	26.2528	6.2550
Spambase (4601 × 57)	93.22 ± 1.25	93.25 ± 1.25	93.36 ± 1.97	**93.52 ± 1.25**	93.27 ± 1.42	93.38 ± 1.29
	65.4455	10.3106	10.1102	35.4455	34.7402	8.2252
W-T-L	14-1-0	11-1-3	10-0-5	12-0-3	12-0-3	

Best accuracy values are in bold

can find that the training times of NFSVM, FTWSVM and FTSVM are more than TWSVM, TBSVM and our NFTSVM, which indicates that the SOR can speed up the training procedure.

4.3 Image recognition

In this subsection, we test our method using the task of image recognition. Three well-known and publicly available databases for image classifications benchmarking, handwritten digits (USPS), objects (COIL-20), and recognition of faces (AR), are used to evaluate our proposed NFTSVM compared with TWSVM, TBSVM, FTWSVM and FTSVM. The USPS database [36] consists of

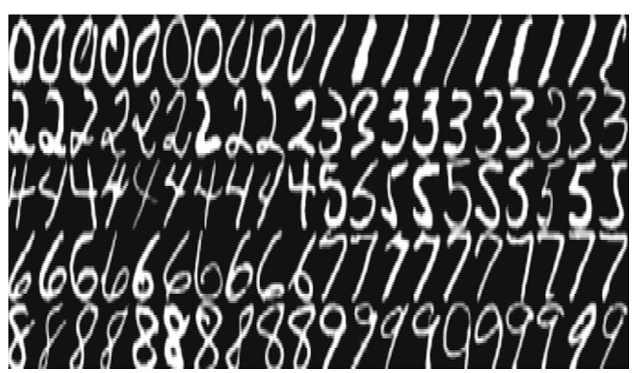

Fig. 3 Subjects in the USPS database

Fig. 4 The subjects in the COIL-20

gray-scale handwritten digit images from 0 to 9, as shown in Fig. 3. Each digit has 1100 images, and the resolution of each image is 16×16. In this paper, we selected five pair-wise digits of varying difficulty for odd vs. even digit classification. COIL-20 [37] is a database of gray scale images of 20 objects, which are illustrated in Fig. 4. 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 resized each image to 32×32. The AR database [38] contains 100 subjects and each subject has 26 face images taken in two sessions. For each session, there are 13 face images. We selected 14 unoccluded images

Fig. 5 An illustration of the selected 14 images of one person from the AR database

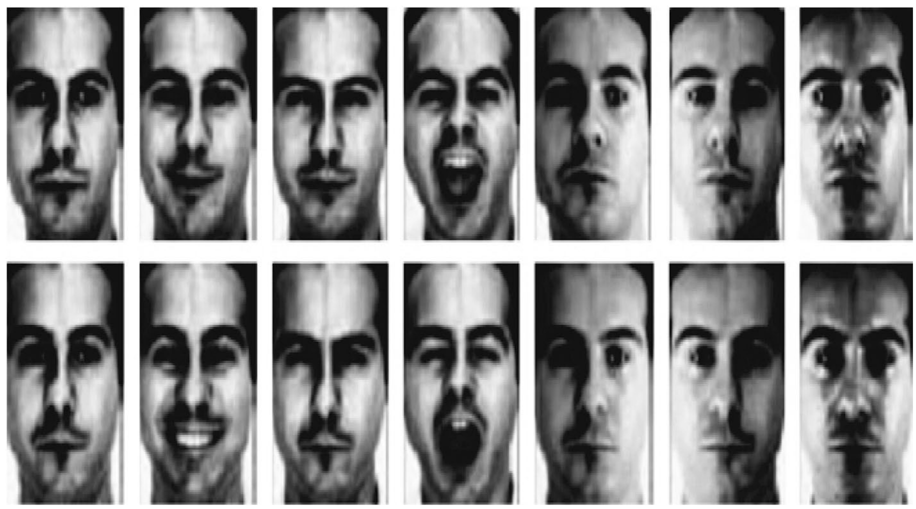

Table 5 The classification performance comparison on the COIL-20, AR and USPS datasets

Datasets	TWSVM Acc + std (%) Time(s)	TBSVM Acc + std (%) Time(s)	FTWSVM Acc + std (%) Time(s)	FTSVM Acc + std (%) Time(s)	NFTSVM Acc + std (%) Time(s)
USPS 1 vs. 7	99.62 ± 0.15 2.1386	**99.87 ± 0.12** 2.3537	99.67 ± 0.13 3.5136	99.76 ± 0.10 2.6658	**99.87 ± 0.13** 2.1647
USPS 2 vs. 3	97.99 ± 0.55 2.1227	98.20 ± 0.18 2.1836	98.12 ± 0.46 3.4369	98.16 ± 0.25 2.6338	**98.29 ± 0.45** 2.1312
USPS 2 vs. 7	99.55 ± 0.17 2.1846	**99.65 ± 0.12** 2.3048	98.76 ± 0.25 3.5391	99.64 ± 0.18 2.5954	**99.65 ± 0.15** **2.1339**
USPS 3 vs. 8	97.29 ± 0.47 2.2499	98.34 ± 0.32 2.2471	98.12 ± 0.69 3.4630	97.73 ± 0.41 2.5324	**98.48 ± 0.27** **2.2319**
USPS 4 vs. 7	99.57 ± 0.17 2.2155	99.73 ± 0.10 2.3963	99.66 ± 0.23 3.5182	99.68 ± 0.16 2.6872	**99.77 ± 0.12** **2.1596**
COIL-20	98.32 ± 0.59 9.3016	98.81 ± 0.70 9.7372	98.85 ± 0.45 14.3223	98.87 ± 0.37 10.5956	**99.19 ± 0.66** **9.2855**
AR	94.33 ± 1.39 10.5869	95.66 ± 0.71 10.7321	96.38 ± 0.59 15.0674	96.43 ± 0.34 11.1707	**96.94 ± 0.71** 10.4379
W-T-L	7-0-0	5-2-0	7-0-0	7-0-0	

Best accuracy values are in bold

from these two sessions in our experiments, as shown in Fig. 5. The 1400 images are all cropped into the same size of 40×30. For these datasets, we randomly split the images of each object into two parts with the same sizes such that one part is selected for training and the remaining part is used for testing. This process is repeated ten times and the average result is reported. Moreover, we only consider the Gaussian kernel for these methods. Table 5 lists the experimental results of these four methods in USPS, COIL-20 and AR datasets. The proposed NFTSVM obtains the best results both on USPS, COIL-20 and AR than the others in most cases according to the W-T-L summarization.

5 Conclusions

Inspired by NFSVM and TWSVM, a new fuzzy twin support vector machine (NFTSVM) was presented in this paper. NFTSVM employs fuzzy membership to weigh each training sample to improve the generalization ability of the system. As discussed in the paper, TWSVM and TBSVM are special cases of our proposed NFTSVM when the parameters of NFTSVM are appropriately selected. Experimental results obtained on a number of benchmarks illustrate the superiority of the proposed NFTSVM. In our future works, we will address the following three issues. First, it is worth noting that there are five parameters in the proposed NFTSVM, hence parameter selection is a practical problem that should be carefully investigated in the future. Second, we will construct fast algorithm to solve QPPs, such as genetic algorithm (GA) [39]. Third, extensions of NFTSVM to multi-class classification [40] and uncertainty mining [41] are also interesting.

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (Grant No. 61373055 and 61672265), the University Natural Science Research Project of Anhui Province of China (Grant No. KJ2015A266, KJ2016A431 and KJ2017A361) and the University Outstanding Young Talent Support Project of Anhui Province of China (Grant No. gxyq2017026).

References

1. Cortes C, Vapnik VN (1995) Support vector machine. Mach Learn 20(3):273–297
2. Vapnik VN (2000) The nature of statistical learning theory. Springer-Verlag, New York (**Incorporated**)
3. Burges C (1998) A tutorial support vector machines for pattern recognition. Data Min Knowl Disc 2:1–43
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. Yen SJ, Wu YC, Yang JC, Lee YS, Liu LL (2013) A support vector machine-based context-ranking model for question answering. Inf Sci 224(1):77–87
6. You ZH, Yu JZ, Zhu L, Li S, Wen ZK (2014) A mapreduce based parallel SVM for large-scale predicting protein–protein interactions. Neurocomputing 145: 37–43
7. Mangasarian OL, Wild EW (2006) Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Trans Pattern Anal Mach Intell 28(1):69–74
8. Jayadeva, R Khemchandani, S Chandra (2007) Twin support vector machines for pattern classification. IEEE Trans Pattern Anal Mach Intell 29(5):905–910
9. Kumar MA, Gopal M (2009) Least squares twin support vector machines for pattern classification. Expert Syst Appl 36(4):7535–7543
10. Shao YH, Zhang CH, Wang XB, Deng NY (2011) Improvements on twin support vector machines. IEEE Trans Neural Networks 22(6):962–968
11. Peng XJ (2011) TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. Pattern Recognit 44(10):2678–2692
12. Qi ZQ, Tian YJ, Shi Y (2013) Robust twin support vector machine for pattern classification. Pattern Recognit 46(1):305–316
13. Wang Z, Shao YH, Wu TR (2014) Proximal parametric-margin support vector classifier and its applications. Neural Comput Applic 24:755–764
14. Shao YH, Chen WJ, Deng NY (2014) Nonparallel hyperplane support vector machine for binary classification problems. Inf Sci 263:22–35
15. Tian YJ, Qi ZQ, Ju XC, Shi Y, Liu XH (2014) Nonparallel support vector machines for pattern classification. IEEE Trans on Cybern 44(7):1067–1079
16. Chen SG, Wu XJ, Zhang RF (2016) A novel twin support vector machine for binary classification problems. Neural Process Lett 44(3):795–811
17. Ye QL, Zhao CX, N. Y, Chen YN (2010) Multi-weight vector projection support vector machines. Pattern Recognit Lett 31:2006–2011
18. Chen XB, Yang J, Ye QL, Liang J (2011) Recursive projection twin support vector machine via within-class variance minimization. Pattern Recognit 44(10):2643–2655
19. Shao YH, Deng NY, Yang ZM (2012) Least squares recursive projection twin support vector machine for classification. Pattern Recognit 45(6):2299–2307
20. Ding SF, Hua XP (2014) Recursive least squares projection twin support vector machines for nonlinear classification. Neurocomputing 130: 3–9
21. Ding SF, JZ Yu, BJ Qi (2014) An overview on twin support vector machines. Artif Intell Rev 42(2):245–252
22. Lin CF, Wang SD (2002) Fuzzy support vector machines. IEEE Trans Neural Networks 13(2):464–471
23. Jiang XF, Yi Z, Lv JC (2006) Fuzzy SVM with a new fuzzy membership function. Neural Comput Appl 15(3–4):268–276
24. 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
25. Ding SF, Han YZ, JZ Yu, YX Gu (2013) A fast fuzzy support vector machine based on information granulation. Neural Comput Appl 23:139–144
26. Wang XZ, RAR Ashfaq, Fu AM (2015) Fuzziness based sample categorization for classifier performance improvement. J Intell Fuzzy Syst 29(3):1185–1196
27. Keller JM, Hunt DJ (1985) Incorporating fuzzy membership functions into the perceptron algorithm. IEEE Trans Pattern Anal Mach Intell 6:693–699

28. Tao Q, Wang J (2004) A new fuzzy support vector machine based on the weighted margin. Neural Process Lett 20(3):139–150
29. Khemchandani R, Jayadeva, Chandra S (2008) Fuzzy twin support vector machines for pattern classification. Mathematical programming and game theory for decision making. World Scientific, Singapore, pp 131–142
30. Li K, Ma HY (2013) A fuzzy twin support vector machine algorithm. Int J Appl Innov Eng Manag 2(3):459–465
31. R.O. Duda, P.E. Hart, D.G. Stork (2001) Pattern classification. Second edition, Wiley, Hoboken
32. Mangasarian OL, Musicant DR (1999) Successive overrelaxation for support vector machines. IEEE Trans Neural Networks 10(5):1032–1037
33. Lee YJ, Huang SY (2007) Reduced support vector machines: a statistical theory. IEEE Trans Neural Networks 13(1):1–13
34. Ripley BD (2008) Pattern recognition and neural networks. Cambridge University, Cambridge
35. Muphy PM, Aha DW (1992) UCI repository of machine learning databases. University of California, Irvine. http://www.ics.uci.edu/~mlearn
36. The USPS Database, http://www.cs.nyu.edu/~roweis/data.html
37. Nene SA, Nayar SK, Murase H (1996) Columbia object image library (COIL-20). Technical report CUCS-005-96, Department of Computer Science, Columbia University
38. Martinez AM, Benavente R (1998) The AR face database. CVC Technical Report #24
39. Wang XZ, He Q, Chen DG, Yeung D (2005) A genetic algorithm for solving the inverse problem of support vector machines. Neurocomputing 68: 225–238
40. Wang XZ, Lu SX, Zhai JH (2008) Fast fuzzy multi-category SVM based on support vector domain description. Int J Pattern Recognit Artif Intell 22(1):109–120
41. Wang XZ (2015) Uncertainty in learning from big data-Editorial. J Intell Fuzzy Syst 28(5):2329–2330