A Fast and Robust TSVM for Pattern Classification

Bin-Bin Gao\textsuperscript{a}, Jian-Jun Wang\textsuperscript{b}

\textsuperscript{a}YouTu Lab, Tencent, Shenzhen 518000, China.
\textsuperscript{b}School of Mathematics and Statistics, Southwest University, Chongqing 400715, China.

Abstract

Twin support vector machine (TSVM) is a powerful learning algorithm by solving a pair of small-sized SVM-type problems. However, there are still some specific issues such as low efficiency and weak robustness when it is faced with some applications. In this paper, we propose a Fast and Robust TSVM (FR-TSVM) to deal with the above issues. In order to strengthen model robustness, we propose an effective fuzzy membership function and reformulate the TSVMs such that different input instances can make different contributions to the learning of the decision hyperplane. To further speed up the training procedure, we develop an efficient coordinate descent algorithm with shirking to solve the involved a pair of quadratic programming problems (QPPs). Moreover, the theoretical foundations of the proposed approach are analyzed in details. The experimental results on several artificial and benchmark datasets indicate that the FR-TSVM not only obtains a fast learning speed but also shows a robust classification performance. Code has been made available at: \url{https://github.com/gaobb/FR-TSVM}.

Keywords: Pattern classification, Support vector machine, Twin support vector machine, Fuzzy membership, Coordinate descent method

1. Introduction

Support vector machine (SVM), invented by Vapnik \cite{1}, is a very popular machine learning algorithm. Due to the adoption structural risk minimization principle, SVM is capable of handling excellently with many classification and regression problems, such as machine fault-diagnosis \cite{2}, image identification \cite{3}, text classification \cite{4}, biomedicine \cite{5} and financial forecast \cite{6}, etc. It has become a prominent highlight of machine learning research. However, the application of SVM is occasionally restricted by some practical issues, especially computational speed and model robustness. To overcome these limitations, some solutions have also been proposed.

Mangasarian et al. \cite{7} changed the proximal planes which are originally parallel to each other to generate a maximal spaced separating hyperplane into nonparallel ones, and proposed a generalized eigenvalue proximal SVM (GEPSVM). The GEPSVM speed-ups training with slighter accurate performance since it is solved by a pair of simple generalized eigenvalue problems. Following this concept, Jayadeva et al. \cite{8} proposed twin SVM (TSVM). Its objective is turned to be optimized by placing the non-parallel proximal planes as closely as possible to their corresponding instances’ cluster and as far as possible from their adversary instances’ cluster (as shown in Fig. 1a). Therefore, the TSVM algorithm converts GEPSVM into a pair of SVM-like convex programs and computation time is reduced satisfactorily to one-fourth of that of a conventional SVM. Recently, numerous nonparallel proximal-plane learning methods are proposed as variants of TSVM, such as TBSVM \cite{9}, Structural TSVM \cite{10}, Robust TSVM \cite{11}, Laplacian Smooth TSVM \cite{12}, Least-square TSVM(Ls-TSVM) \cite{13}, Ls-TSVM for multi-class \cite{14}, Twin Parametric-margin SVM \cite{15, 16}, Multi-label TSVM \cite{17}, RPTWSVM \cite{18} and Pin-TSVM \cite{19}. In additional, some regression model based TSVM are also proposed such as TSVR \cite{20}, TPSVR\cite{21}, \(\epsilon\)-TSVR \cite{22}, TWSVR \cite{23} and \(v\)-TWSVR \cite{24}. All of the proposed variants share the same merits of TSVM. However, one challenge is that training instances from real-world applications often carry information with significant noise. These TSVM methods are sensitive to outliers or noises.

Noisy data often can deteriorate the generalization ability of SVM or TSVM. In term of fuzzy theory technique, no\textit{isy information} can be causally converted into the \textit{fuzzy information} to meet fuzzy inference theory. The training of SVM would be too sensitive to noisy inputs if all training instances are treated equivalently at the training stage. A conceptual way to alleviate this sensitive deterioration is to contract the influence of noisy inputs. This means that a conventional SVM, which intrinsically treats every input instance in equivalence, can be improved by introducing fuzzy membership functions to soften input information. A category of fuzzy SVMs are hence developed, such as Lin and Wang \cite{25, 26}. Wu

Email addresses: csgaobb@gmail.com (Bin-Bin Gao), wjj@swu.edu.cn (Jian-Jun Wang)
and Liu [27], Inoue and Abe [28], Yang et al. [29], and Tang [30] etc. The elementary concept of fuzzy SVM is to allocate a small confident membership for each noisy instance consistent with the information fuzziness which the instance has carried to reduce its influence on the optimization. The membership is generally assigned according to the instance’s confidence intrinsically related to its native class. The introduction of fuzzy membership reduces effectively the uncertainty caused by noisy instances and leads to a robust classifier. To utilize the fuzzy concept, some variants of fuzzy TSVM have been developed in [31] [32] [33]. However, these fuzzy SVM methods equivalently assigned the fuzzy membership to each instance and they cannot distinguish support vectors and outliers effectively.

In term of computational efficiency, plenty of input instances from real-world applications require abundant quadratic programming computations, which slow down training efficiency. This computational cost significantly limits the applications of SVM. There are various solutions for improvement, including chunking [34], decomposition [35], sequential minimal optimization (SMO) [36] etc. However, these fuzzy SVM methods equivalently assigned the fuzzy membership to each instance and they cannot distinguish support vectors and outliers effectively.

In this paper, we propose a fast and robust twin support vector machine (FR-TSVM) for a classification problem. Our contributions are summarized as follows.

- We construct a novel fuzzy membership function to measure the importance of each training instance. The assignment policy of fuzzy membership depends on the structural information of training instances in feature space.
- We embed the proposed fuzzy concept into TSVM and propose a fast and robust TSVM model. In FR-TSVM, different input instances can make different contributions to the learning of decision hyperplanes. The proposed model can effectively alleviate the effect of noisy instances and obtain a robust performance.
- We develop further coordinate descent algorithm with shrinking to speed-up the computations of linear and nonlinear FR-TSVM. In addition, this paper presents FR-TSVM in details including corresponding theoretical fundamentals and related properties.
- Experiments on artificial and benchmark datasets show that the proposal brings more satisfactory performances on both classification accuracy and learning efficiency, compared with traditional SVM, FSVM and TSVM.

The remaining parts of this paper are organized as follows. We first briefly review the basics of classical SVM and its related works, including FSVM and TSVM in Section 2. Then, Section 3 proposes FR-TSVM approach, including fuzzy membership function construction, linear and nonlinear FR-TSVM models and their optimization algorithms. After that, experimental results are reported in Section 4. Some preliminary results have been published in a conference presentation [10].

2. Background

In this paper, we mainly focus on a binary classification problem in the n-dimensional space $\mathbb{R}^n$. We denote training data as $D = \{(x_i, y_i)|i = 1, 2, \ldots, l\}$, where $x_i \in \mathbb{R}^n$ represents an input instance with the corresponding label $y_i \in \{+1, -1\}$. Without loss of generality, we assume that the matrix $X$ with size of $l \times n$ represents all training instances, and the matrices $X_{\pm}$ with sizes of $l_\pm \times n$ represent the training instances belonging to “+1” (positive) or “-1” (negative) class, respectively, where $l = l_+ + l_-$. 

2.1. Support vector machine

As a classical machine learning algorithm, the standard linear SVM [1] aims to construct a pair of parallel hyperplanes between two classes of instances:

$$w^T x + b = +1 \quad \text{and} \quad w^T x + b = -1,$$

where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are the normal vector and the bias term of hyperplanes, respectively. These separating hyperplanes are obtained by solving the following quadratic programming problem (QPP):

$$\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + c \sum_{i=1}^{l} \xi_i$$

s.t. $y_i(w^T x_i + b) + \xi_i \geq 1, \xi_i \geq 0, i = 1, 2, \ldots, l.$

where $\|\cdot\|$ stands for the $\ell_2$-norm, $\xi_i$ is called as slack variable which denotes the misclassification error associated with the $i$-th input instance and $c > 0$ is regularization factor that balances the importance between the maximization of margin width (i.e., the minimization of $\frac{1}{2}\|w\|^2$) and the minimization of the training error. The dual QPP of problem Eq.(2) is:

$$\max_{\alpha} \sum_{i=1}^{l} \alpha_i + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \langle x_i, x_j \rangle \alpha_i \alpha_j$$

s.t. $\sum_{i=1}^{l} \alpha_i y_i = 0, 0 \leq \alpha_i \leq 1, i = 1, 2, \ldots, l,$

where $\alpha \in \mathbb{R}^l$ is Lagrangian multiplier. After solving this dual QPP, a testing instance $x$ is classified as “+1” or “-1” following decision function

$$f(x) = \text{sign}(w^*^T x + b^*)$$

where $(w^*, b^*)$ and $\alpha^*_i$ are the solution of Eq.(2) and Eq.(3), respectively.
and multipliers, the dual QPPs of Eq.(6) and Eq.(7) can be tors with the vectors of slack variables for positive and negative instances, respectively. A pair of nonparallel decision hyperplanes are shown by red and blue dashed line and the best decision plane is represented by a black line. We can see that the proposed method is easy to get a better decision hyperplane because it can effectively detect noisy points and weaken their effects.

2.2. Twin support vector machine

Different from the conventional SVM, TSVM is in fact constructed by two non-parallel decision planes, i.e.,

$$w^T_+ x + b_+ = 0 \quad \text{and} \quad w^T_- x + b_- = 0. \quad (5)$$

To construct such two non-parallel decision planes, a pair of primal optimization problems are set up:

$$\min_{w_+, b_+} \frac{1}{2} \| X_+ w_+ + e_+ b_+ \|^2 + c_1 e^T_+ \xi_+$$

s.t. $$-(X_- w_+ + e_+ b_+) + \xi_+ \geq e_-; \xi_- \geq 0,$$

and

$$\min_{w_-, b_-} \frac{1}{2} \| X_- w_- + e_- b_- \|^2 + c_2 e^T_- \xi_-$$

s.t. $$(X_+ w_- + e_- b_-) + \xi_+ \geq e_+; \xi_+ \geq 0,$$

where $c_1 > 0$ and $c_2 > 0$ are parameters, $\xi_+$ and $\xi_-$ denote the vectors of slack variables for positive and negative classes, respectively, and $e_+, e_-$ correspond to unit vectors with $l_\pm$ dimensions. By introducing the Lagrangian multipliers, the dual QPPs of Eq.(5) and Eq.(7) can be represented as followings

$$\max_{\alpha} \quad e^T_- \alpha - \frac{1}{2} \alpha^T H_- (H^T_- H_-)^{-1} H^T_- \alpha$$

s.t. $$0 \leq \alpha \leq c_1 e_-,$$

$$\max_{\beta} \quad e^T_+ \beta - \frac{1}{2} \beta^T H_+ (H^T_+ H_+)^{-1} H^T_+ \beta$$

s.t. $$0 \leq \beta \leq c_2 e_+,$$

where $H_+ = [X_+, e_+], H_- = [X_-, e_-]$. The non-parallel hyperplanes Eq.(6) can be obtained from the solutions $\alpha^\ast$ and $\beta^\ast$ of Eq.(8) and Eq.(9) by

$$u^\ast_+ = -(H^T_+ H_-)^{-1} H^T_- \alpha^\ast, \quad u^\ast_- = (H^T_+ H_-)^{-1} H^T_+ \beta^\ast, \quad (10)$$

where $u^\ast_\pm = [w^T_\pm, b^T_\pm]^T$, $w^T_\pm$ and $b^\ast_\pm$ are the solutions of Eq.(8) and Eq.(9). TSVM then can easily assign a label “+1” or “−1” to a testing instance $x$ by

$$f(x) = \arg\min_{\pm} \frac{\| w^T_{\pm} x + b^\ast_{\pm} \|}{\| w^T_{\pm} \|}, \quad (11)$$

where $\| \cdot \|$ is a function taking its absolute value. If the matrix $H^T_+ H_+$ or $H^T_- H_-$ is ill-conditioned, TSVM artificially introduces a term $\lambda I (\lambda > 0)$, where $I$ is an identity matrix of appropriate dimension. In the experiments, we fix the value of $\lambda$ as 0.01.

2.3. Fuzzy support vector machine

To reduce the effects of outliers, Lin et al. introduced fuzzy membership to each input instance of SVM and proposed fuzzy SVM (FSVM) [25]. In FSVM, training instance $x_i$ is assigned a fuzzy membership $0 \leq \mu_i \leq 1$ besides a label $y_i \in \{+1, -1\}$. The input dataset $T$ is thus modified as $T' = \{(x_i, y_i, \mu_i) | i = 1, 2, \ldots, l\}$. These fuzzy memberships $\{\mu_i | i = 1, 2, \ldots, l\}$ are used to reduce the influence of noisy instances for generating final decision function, which induces fuzzy SVM as followings:

$$\min_{w, b, \xi} \frac{1}{2} \| w \|^2 + c \sum_{i=1}^l \mu_i \xi_i$$

s.t. $$y_i (w^T x_i + b) + \xi_i \geq 1, \xi_i \geq 0, i = 1, 2, \ldots, l.$$
It is noted that a smaller \( s_i \) can reduce the effect of the parameter \( \xi_i \) in problem Eq.(12) so that the corresponding instance \( x_i \) can be treated as less important. The classification of any testing instance \( x \) can be obtained by determining the sign of \( w^* x + b^* \) where \( w^* \) and \( b^* \) are the solution of Eq.(12).

3. The proposed FR-TSVM approach

In this section, we will propose the approach of FR-TSVM. To this end, we first introduce fuzzy membership construction. Then, we propose linear and nonlinear FR-TSVM models by embedding fuzzy membership into TSVM. Finally, a fast optimization method is developed for solving the dual problems of the proposed FR-TSVM.

3.1. Fuzzy membership construction

Fuzzy membership plays a key role in robust classification learning. However, there is no unified standard to construct such fuzzy membership so far. As we know, support vectors geometrically locate near the boundary area of two adjacent classes. The noisy points also reside in this area typically and unfortunately. This means that support vectors and noisy points are frequently mixed together. Inspired by Tang [30], we propose a fuzzy membership assignment for training instances. The proposed method considers not only reducing the noise carried by the outliers but also keeping the importance of support vectors.

3.1.1. Linear case

The construction of fuzzy membership considers firstly linear kernel case. We define directly positive and negative class centers, \( x_{c+} \) and \( x_{c-} \), as the mean points in the input space for positive and negative instances,

\[
x_{c+} = \frac{1}{l_+} \sum_{y_i = +1} x_i, \quad x_{c-} = \frac{1}{l_-} \sum_{y_i = -1} x_i.
\]

Measuring the distance between instances and their class center, the hypersphere radius of positive and negative are defined and given as

\[
r_+ = \max \{ \| x_i - x_{c+} \| | y_i = +1 \}, \quad r_- = \max \{ \| x_i - x_{c-} \| | y_i = -1 \}.
\]

With known \( x_{c+}, x_{c-}, r_+, \) and \( r_- \), membership \( s_i \) of an instance can be assigned according to the relationship between \( \| x_i - x_{c+} \| \) and \( \| x_i - x_{c-} \| \). In this paper, \( s_i \) of a positive instance be given as:

\[
s_i = \begin{cases} \mu (1 - \| x_i - x_{c+} \|/(r_+ + \delta)), & \text{if } \| x_i - x_{c+} \| \\
(1 - \mu) (1 - \| x_i - x_{c-} \|/(r_- + \delta)), & \text{if } \| x_i - x_{c+} \| < \| x_i - x_{c-} \| \end{cases} \quad \text{for } y_i = +1, \quad \text{where } \mu \in [0, 1]
\]

where \( \mu \in [0, 1] \) is used to balance the effect of normal and noisy instances, and \( \delta > 0 \) to avoid \( s_i = 0 \). The relationship reveals that an instance is generally assigned by a proportionally decreasing \( s_i \) value when it drifts farther from its native class center to increase uncertainty. Moreover, some instances are highly suspected as outliers which dwell with a sufficient far distance from their native class center (i.e. \( \| x_i - x_{c+} \| \geq \| x_i - x_{c-} \| \)). In order to decrease outliers’ effect toward the hyperplane, we assign a small positive real number for \( \mu \). In practice, we set \( \mu = 0.1 \) for FR-TSVM. A similar rule can be applied to the fuzzy membership of those negative instances. As shown in Fig. 2, we intuitively show linear fuzzy membership values for training instances.

3.1.2. Nonlinear case

For nonlinear case, fuzzy membership function must be consistently reconstructed in the feature space \( H \). Similar to linear case, positive and negative class centers \( \varphi_{c+} \) and \( \varphi_{c-} \) in the feature space are defined as:

\[
\varphi_{c+} = \frac{1}{l_+} \sum_{y_i = +1} \varphi(x_i), \quad \varphi_{c-} = \frac{1}{l_-} \sum_{y_i = -1} \varphi(x_i).
\]

where \( \varphi(x_i) \in H \) denotes the transformation of an arbitrary input instance \( x_i \). The squared distance from \( \varphi(x_i) \) to \( \varphi_{c+} \) or \( \varphi_{c-} \) can be rearranged and expressed in terms of...
of the kernel function $\kappa(\cdot, \cdot)$:

\[
\| \varphi(x_i) - \varphi_{c+} \|^2 = \| \varphi(x_i) \|^2 - 2\langle \varphi(x_i), \varphi_{c+} \rangle + \| \varphi_{c+} \|^2
\]

\[= \langle \varphi(x_i), \varphi(x_i) \rangle - \frac{2}{l} \sum_{y_j=+1} \langle \varphi(x_i), \varphi(x_j) \rangle
\]

\[+ \frac{1}{l} \sum_{y_i=+1} \sum_{y_k=+1} \kappa(x_i, x_k)
\]

\[= \kappa(x_i, x_i) - \frac{2}{l} \sum_{y_j=+1} \kappa(x_i, x_j)
\]

\[+ \frac{1}{l} \sum_{y_i=+1} \sum_{y_k=+1} \kappa(x_i, x_k),
\]

where $\kappa(x_i, x_j)$ is the kernel function which implicitly calculates the high-dimensional dot product of $\varphi(x_i)$ and $\varphi(x_j)$. The scattering hypersphere radii in the feature space are

\[r_{\varphi+} = \max \{ \| \varphi(x_i) - \varphi_{c+} \| \mid y_i = +1 \}, \quad (20)\]

\[r_{\varphi-} = \max \{ \| \varphi(x_i) - \varphi_{c-} \| \mid y_i = -1 \}.
\]

Based on the same principle in linear case, the fuzzy membership function for non-linear kernel can equivalently given as:

\[
s_i = \left\{ \begin{array}{ll}
\mu (1 - \sqrt{\| \varphi(x_i) - \varphi_{c+} \|^2 / (r_{\varphi+}^2 + \delta)}) & \text{if } \| \varphi(x_i) - \varphi_{c+} \| \geq \| \varphi(x_i) - \varphi_{c-} \| \text{ and } y_i = +1 \\
(1 - \mu) (1 - \sqrt{\| \varphi(x_i) - \varphi_{c+} \|^2 / (r_{\varphi+}^2 + \delta)}) & \text{if } \| \varphi(x_i) - \varphi_{c+} \| < \| \varphi(x_i) - \varphi_{c-} \| \text{ and } y_i = +1
\end{array} \right.
\]

\[= s_i = \frac{\mu (1 - \sqrt{\| \varphi(x_i) - \varphi_{c+} \|^2 / (r_{\varphi+}^2 + \delta)})}{(1 - \mu) (1 - \sqrt{\| \varphi(x_i) - \varphi_{c+} \|^2 / (r_{\varphi+}^2 + \delta)})},
\]

where $\delta$ is similarly defined as a small positive constant to avoid the vanishing of $s_i$. Of course, a fuzzy membership function $s_i$ for negative class instances can be similarly defined.

### 3.2. Fast and Robust twin support vector machine

In this section, we propose an efficient learning approach termed as fast and robust twin support vector machine (FR-TSVM). As mentioned earlier, the FR-MSVM is similar to TSVM, as it also derives a pair of nonparallel decision hyperplanes through two QPPs. What is more, FR-TSVM is more robust and fast than TSVM.

#### 3.2.1. Linear FR-TSVM

For linear case, the FR-TSVM finds two nonparallel hyperplanes in $\mathbb{R}^n$ space

\[w_+^T x + b_+ = 0, \quad \text{and} \quad w_-^T x + b_- = 0. \quad (23)\]

Considering the crucial trade-off balance between margin maximization and error minimization, a margin term similar to that in the standard SVM [1], should be added firstly in the model. Since TSVM has two proximal decision functions, $w_+^T x + b_+ = 0$, two margin terms $1/\|w_+\|$ and $1/\|w_-\|$ are accordingly defined for proximal decision functions, respectively. Together with the introduced fuzzy membership function and the margin terms, a weight regularized model of FR-TSVM with the linear kernel is hence proposed:

\[
\min_{w_+, b_+, \xi_+} \frac{1}{2} c_1 \| w_+ \|^2 + \frac{1}{2} \| X_+ w_+ + e_+ b_+ \|^2 + c_3 s^T \xi_-
\]

\[\text{s.t.} \quad -(X_+ w_+ + e_- b_+) + \xi_+ \geq e_+, \quad \xi_- \geq 0, \quad (24)\]

\[
\min_{w_-, b_-, \xi_-} \frac{1}{2} c_2 \| w_- \|^2 + \frac{1}{2} \| X_- w_- + e_- b_- \|^2 + c_4 s^T \xi_+
\]

\[\text{s.t.} \quad (X_- w_- + e_+ b_+) + \xi_+ \geq e_+, \quad \xi_+ \geq 0, \quad (25)\]

where $c_i > 0 (i = 1, 2, 3, 4)$ are trade-off parameters for weighting the regularization, $\xi_+$ and $\xi_-$ denote the subsets of misclassification error for positive and negative classes respectively, both $s_+ \in R^{l+}$ and $s_- \in R^{l-}$ are the fuzzy-number vectors sequentially associated with positive and negative instances, and $e_+, e_-$ correspond to unit vectors with their dimensions exact to sample sizes in positive and negative classes. The parameters $c_i (i = 1, 2, 3, 4)$ are used to balance the effect of maximizing the margin and minimizing the adapting error which aggregates all the individual error measured from instances to its corresponding hyperplane. An intuitive geometric interpretation for the linear FR-TSVM is shown in Fig. [11].

**Theorem 1.** The dual forms of the primal problems Eq. (24)-(25) are

\[
\max_{\alpha} \quad e^T \alpha - \frac{1}{2} \alpha^T H_+ (H^T_+ H_+ + c_1 E_1)^{-1} H^T_+ \alpha
\]

\[\text{s.t.} \quad 0 \leq \alpha \leq c_3 s_-, \quad (26)\]

\[
\max_{\beta} \quad e^T \beta - \frac{1}{2} \beta^T H_- (H^T_- H_- + c_2 E_2)^{-1} H^T_- \beta
\]

\[\text{s.t.} \quad 0 \leq \beta \leq c_4 s_+, \quad (27)\]

where $H_+ = [X_+, e_+], H_- = [X_-, e_-], \text{and } E_i = \begin{pmatrix} 1 & 0 \end{pmatrix}$ ($i = 1, 2$). Relationships of the optimal solutions between the primal problems Eq. (24)-(25) and their dual problems Eq. (26)-(27) are

\[
u^*_+ = -(H^T_+ H_+ + c_1 E_1)^{-1} H^T_+ \alpha^*;
\]

\[
u^*_- = (H^T_- H_- + c_2 E_2)^{-1} H^T_- \beta^*, \quad (28)\]
where \( \mathbf{u}^*_+ = [\mathbf{w}^*_+^T, b^*_+]^T \), \( \alpha^* \) and \( \beta^* \) denote the optimal values of \( \alpha \) and \( \beta \), respectively.

**Proof.** Taking Lagrangian of the primal problem Eq.(42), the problem becomes:

\[
\begin{align*}
L(w_+, b_+, \xi) &= \frac{1}{2} c_1 \| w_+ \|^2 + \frac{1}{2} \| X_+ w_+ + e_+ b_+ \|^2 \\
&+ \alpha^T (X_+ w_+ + e_+ b_+ - \xi - e_-) \\
&+ c_3 s^T \xi - \eta^T \xi, \\
\end{align*}
\]

where non-negative \( \alpha \) and \( \eta \) are Lagrange multipliers. According to the KKT conditions, we have

\[
\begin{align*}
\nabla_{w_+} L &= c_1 w_+ + X_+^T (X_+ w_+ + e_+ b_+) + X_+^T \alpha = 0, \\
\nabla_{b_+} L &= e_+^T (X_+ w_+ + e_+ b_+) + e_+^T \alpha = 0, \\
\nabla_{\xi} L &= c_3 s - \alpha - \eta = 0, \\
-(X_+ w_+ + e_+ b_+) + \xi &\geq e_-, \xi \geq 0, \\
\alpha^T (X_- w_+ + e_- b_+ - \xi - e_-) &= 0, \eta^T \xi = 0.
\end{align*}
\]

Summarized from presumed conditions \( \alpha \geq 0, \eta \geq 0 \), and Eq.(32), \( \alpha \) is bounded by:

\[
0 \leq \alpha \leq c_3 s_-
\]

Combining Eq.(30) and Eq.(31) yields:

\[
([X_+, e_+]^T [X_+, e_-] + c_1 E_1) [w^*_+^T, b^*_+]^T + [X_-, e_-]^T \alpha = 0
\]

**Theorem 2.** The dual forms of the primal problems Eq.(41) and Eq.(42) are:

\[
\begin{align*}
\max_{\mathbf{\alpha}} \quad & e^T \mathbf{\alpha} - \frac{1}{2} \alpha^T S_+ (S_+^T S_+ + c_1 E_1)^{-1} S_+^T \mathbf{\alpha} \\
\text{s.t.} \quad & 0 \leq \alpha \leq c_3 s_-, \\
\max_{\mathbf{\beta}} \quad & e^T \mathbf{\beta} - \frac{1}{2} \beta^T S_- (S_-^T S_- + c_2 E_2)^{-1} S_-^T \mathbf{\beta} \\
\text{s.t.} \quad & 0 \leq \beta \leq c_4 s_+,
\end{align*}
\]

3.2.2. Nonlinear FR-TSVM

In nonlinear case, the classification problem is intuitively solved by mapping input instance \( \mathbf{x} \) from the input space \( \mathcal{R}^n \) to a high-dimensional feature space \( \mathcal{H} \) through transformation \( \varphi(x) \). Using alternative kernel function \( \kappa(\cdot, \cdot) \), which implicitly calculates the dot-product of a pair of transformations \( \kappa(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle \), similarity manipulation of transformed \( x_1 \) and \( x_2 \) can be resolved and utilized to deal with the nonlinear FR-TSVM. The fact, which makes the transformation helpful for the nonlinear FR-TSVM, is that the optimal separating hyperplane can be constructed linearly in the high-dimensional feature space [11]. With the kernel function, the nonlinear dual proximal hyperplanes of FR-TSVM can be stated as:

\[
\kappa(x, X^T) w_+ + b_+ = 0 \quad \text{and} \quad \kappa(x, X^T) w_- + b_- = 0.
\]

To obtain the above two hyperplanes, the primal problems of nonlinear FR-TSVM can be expressed as:

\[
\begin{align*}
\min_{w_+, b_+, \xi_+} \quad & \frac{1}{2} c_1 \| w_+ \|^2 + \frac{1}{2} \| \kappa(X_+, X^T) w_+ + e_+ b_+ \|^2 \\
\text{s.t.} \quad & -\kappa(X_-, X^T) w_+ + e_- b_+ + \xi_+ \geq e_-, \xi_+ \geq 0,
\end{align*}
\]

\[
\begin{align*}
\min_{w_-, b_-, \xi_-} \quad & \frac{1}{2} c_2 \| w_- \|^2 + \frac{1}{2} \| \kappa(X_+, X^T) w_- + e_+ b_- \|^2 \\
\text{s.t.} \quad & -\kappa(X_-, X^T) w_- + e_+ b_- + \xi_- \geq e_+, \xi_- \geq 0.
\end{align*}
\]

By solving the dual forms of Eq.(41) and Eq.(42), one can obtain the optimal solutions \( \alpha^* \) and \( \beta^* \) of the dual problems, and furthermore \( \mathbf{u}^*_\pm \) of the corresponding primal problems. The non-parallel proximal hyperplanes can thus be subsequently obtained. For a testing instance \( x \in \mathcal{R}^n \), the classification decision function can be given as:

\[
f(x) = \arg\min_{\pm} \frac{\mathbf{w}_\pm^T x + b^*_\pm}{\| \mathbf{w}_\pm^* \|}.
\]
Algorithm 1: A dual CD method for FR-TSVM

1: Compute $Q = (H^T L + c_1 E_1)^{-1} H^T$ and $\overline{Q}_{ii} = H_{-i} Q_i$
2: Initial $\alpha \leftarrow 0$ and $\mathbf{u}_+ \leftarrow 0$
3: while $\alpha$ is not optimized do
4:   for $i = 1, 2, \ldots, l_-$ do
5:     $\nabla_i f(\alpha) = -H_{-i} \mathbf{u}_+ - 1$
6:     Compute $\nabla_i^T f(\alpha)$ by Eq. (54)
7:     if $\nabla_i^T f(\alpha) \neq 0$ then
8:         $\pi_i \leftarrow \alpha_i$
9:         $\alpha_i \leftarrow \min(\alpha_i - \nabla_i^T f(\alpha)/\overline{Q}_{ii}, 0), c_3 s_i-$
10:        $u_{+i} \leftarrow u_{+i} - q_i(\pi_i - \pi_i)$
11:     end if
12:   end for
13: end while

Once solutions of the dual problems Eq. (13) and Eq. (44) are obtained, solutions of the primal problems Eq. (11) and Eq. (12) can be obtained through Eq. (15), and the decision function for classifying a testing instance $x \in \mathbb{R}^n$ is eventually given by:

$$f(x) = \arg \min_\alpha \left| \kappa(x, X^T) w_+^T + b_+^T \right| / \sqrt{w_+^T \kappa(x, X^T) w_+^T}.$$  \hfill (46)

3.3. A fast optimization method for FR-TSVM

3.3.1. Solving FR-TSVM with the pure coordinate descent

Based on the quadratic differentiable expressions of the FR-TSVM’s objective functions Eq. (26)-(27) and Eq. (43)-(44), a coordinate descent method can be further employed for solving the FR-TSVM. There are pairwise similarities between Eq. (26)-(27) and Eq. (43)-(44). The intuition is that if either one of these functions can be reformulated as a quadratic expression, the coordinate descent method would be applied accordingly with the algorithms proposed by [38, 39, 42], and be easily extended to the other three objective functions. By the motivation, we initially show the dual FR-TSVM with the first objective function of Eq. (26) as below.

With $Q = (H^T L + c_1 E_1)^{-1} H^T$ and $\overline{Q} = H L Q$, the problem Eq. (26) can be abbreviated as a quadratic expression:

$$\min_{\alpha} f(\alpha) = \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \quad \text{s.t.} \quad 0 \leq \alpha \leq c_3 s_-. \quad (47)$$

As an iterative scheme, the FR-TSVM generates a sequence of updating vectors $\{\alpha_{k1}, \ldots, \alpha_{kl} \}$ to consecutively optimize the objective function where $\alpha^{k} = [\alpha_{k1}^T, \ldots, \alpha_{kl}^T]^T \in \mathbb{R}^l$. There are two levels of iterations. An integer $k$ first is used to index the 2nd-level of outer iterations updated from $\alpha^{k}$ to $\alpha^{k+1}$. In every $k$-th outer iteration, the update of $\alpha^{k}$ is further subdivided into 1st-level of $l_-$ inner iterations, indexed by $i$, to generate a series of intermediate vectors

$$\{\alpha_{k1}, \alpha_{k2}, \ldots, \alpha_{ki}, \ldots, \alpha_{kl}, \alpha_{kl+1} \}. \quad (48)$$

The two-level updated vector $\alpha^{ki}$ is thus expressed as:

$$\alpha^{ki} = [\alpha_{k1}^{i+1}, \ldots, \alpha_{ki}^{i+1}, \alpha_{ki}^{i}, \ldots, \alpha_{kl}^{i+1}]^T, \quad \forall i = 1, 2, \ldots, l_-. \quad (49)$$

and

$$\alpha^{k+1} = \alpha^{k} + \alpha^{ki+1} - \alpha^{ki}. \quad (50)$$

To update the intermediate $\alpha^{ki+1}$ to $\alpha^{ki+1}$, the following single variable subproblem should be solved:

$$\min_{d} f(\alpha^{ki} + d e_i) \quad \text{s.t.} \quad 0 \leq \alpha^{ki} + d \leq c_3 s_-, \quad (51)$$

where $e_i$ is the $i$-th orthogonal basis vector of $\mathbb{R}^{l_-}$. Indeed, the objective function Eq. (51) corresponds to a quadratic function of $d$:

$$f(\alpha^{ki} + d e_i) = \frac{1}{2} \overline{Q}_{ii} d^2 + \nabla_i f(\alpha^{ki}) d + c, \quad (52)$$

where $\nabla_i f$ denotes the $i$-th component of gradient $\nabla f$, and $c$ is an arbitrary constant. Apparently, Eq. (52) has an optimum at $d = 0$ if and only if

$$\nabla_i^T f(\alpha^{ki}) = 0, \quad (53)$$

where $\nabla_i^T f(\alpha^{ki})$ is a projected gradient. To gain the possibility to refine the optimum, the project gradient should be satisfactory with:

$$\nabla_i^T f(\alpha^{ki}) = \left\{ \begin{array}{ll}
0, & \text{if } \alpha_{i} = 0 \\
\nabla_i f(\alpha), & \text{if } 0 < \alpha_i < c_3 s_-
\end{array} \right. \quad \text{max}(0, \nabla_i f(\alpha)), \quad \text{if } \alpha_i = c_3 s_. \quad (54)$$

The key for computational reduction is that we can directly move forward to next $i + 1$ iteration without updating $\alpha_i^{ki}$ in the $l_-$ length inner-iteration updates if Eq. (53) has been fulfilled. In other words, we only update $\alpha_i^{ki}$ to temporarily meet the optimal solution of Eq. (51). By introducing Lipschitz continuity [33], the optimum of Eq. (52) can be reached by:

$$\alpha_i^{k+1} = \min(\max(\alpha_i^{ki} - \nabla_i f(\alpha^{ki})/\overline{Q}_{ii}, 0), c_3 s_-). \quad (55)$$

However $\alpha_i^{k+1}$ is updated or not in the $l_-$ length inner iterations; the process would be repeated in the outer iterations once and once again until the presumed termination condition is reached. In the update of Eq. (55), $\overline{Q}_{ii}$ can be pre-calculated by $\overline{Q}_{ii} = H_{-i} Q_i$, where $Q = (H^T L + c_1 E_1)^{-1} H^T$, and preserved through all the iterations, and $\nabla_i f(\alpha^{ki})$ can be obtained by

$$\nabla_i f(\alpha) = (\overline{Q}(\alpha))_i - 1 = \sum_{j=1}^{l_-} \overline{Q}_{ij} \alpha_j - 1. \quad (56)$$

Here, the computation of $\nabla_i f(\alpha^{ki})$ by Eq. (56), which is approximated as $O(l_- \pi)$ where $\pi$ is the average count of non-zero elements in $\overline{Q}$ per instance, is expensive. In order
to reduce the computation, $\nabla_i f(\alpha^{k,i})$ can alternatively be calculated by [39]:

$$\nabla_i f(\alpha) = -H_{-i}u_+ - 1,$$  

(57)

with a pre-defined $u_+$

$$u_+ = -Q\alpha,$$  

(58)

where $H_{-i}$ is the $i$-th row of matrix $H_-$. With this alternative, the time complexity of computing $\nabla_i f(\alpha^{k,i})$ can be reduced as $O(\pi)$. In order to employ Eq. (57) for calculating $\nabla_i f(\alpha^{k,i})$, it is required to maintain $u_+$ throughout the whole coordinate descent procedure by an update policy:

$$u_{+i} \leftarrow u_{+i} - Q_i(\alpha_i - \pi_i).$$  

(59)

Here, the time consumption by the iterative maintaining of $u_+$ requires only $O(\pi)$ rather than that by the direct calculation by Eq. (58), where $\pi_i$ and $\alpha_i$ denote the values of the primal optimizer before and after the corresponded update iteration, respectively. With a zero initial value for the first $u_+$ due to the generally adopted $\alpha^0 = 0$, the optimal solution of $u_+$ for the primal problem Eq. (26) can be eventually obtained by iterative updates of Eq. (59). Algorithm 1 describes the entire process.

#### 3.3.2. Speeding-up FR-TSVM with heuristic shrinking

Although the quadratic expressions of FR-TSVM inherit most essential merits from the convex quadratic optimization, the solutions of FR-TSVM, even the temporal solutions $\alpha^k$, are still constrained in a specified range, for example, $0 \leq \alpha_i \leq c_3 s_i$, for the problem Eq. (26). With the bounded Lagrange multipliers, $\alpha_i = 0$ or $\alpha_i = c_3 s_i$, further iterative effort may vanish and remain the temporal objective function outcome in a steady state. Since FR-TSVM produces relatively considerable amounts of bounded Lagrange multipliers in the iterative process, a policy to early stop the update of such bounded multipliers of reducing the scale of optimization programming is indeed beneficial to the subsequent computation, and speed-ups the FR-TSVM.

At the same time, for simplicity, we examine only the coordinate descent of Eq. (26) in the twin constrained optimization problems Eq. (26)-(27) and Eq. (43)-(44) with the heuristic shrinking technique [4]. Once the heuristic shrinking is applicable, the examination of Eq. (27) and Eq. (43)-(44) can be equivalently conducted and a comparable result can be assessed with the similarity to each other.

To employ the heuristic shrinking [4], a subset removing some elements from $\{1, 2, \cdots, l\}$ is defined as an active set $A$, and the complimentary subset is defined contradictorily as an inactive set $\overline{A} = \{1, \cdots, l\} \setminus A$. The use of the active set is for dynamically collecting those non-bound Lagrange multiplier still effective to the optimization. With the separation of $A$ from $\overline{A}$, the optimization problem Eq. (26) can be decomposed and reorganized as:

$$\min_{\alpha} \frac{1}{2} \alpha^T \overline{Q}_A^T\alpha_A + (\overline{Q}_A^T\alpha_{\overline{A}} - e_A)^T\alpha_A \text{ s.t. } 0 \leq \alpha_A \leq c_A, \alpha_{\overline{A}} = 0,$$  

(60)

where $\overline{Q}_A$ and $\overline{Q}_A^T$ are sub-matrices of $\overline{Q}$, and $\alpha_A$ and $\alpha_{\overline{A}}$ are Lagrange multiplier sub-vectors corresponding to subsets $A$ and $\overline{A}$, respectively. As illustrated previously, $\alpha_{\overline{A}}$ contains only those inactive bounded multipliers which can not contribute furthermore to the optimization. A divide-and-conquer strategy is thus used to achieve the optimization more efficiently due to the eliminated computations of the second part of Eq. (60). As described in the theory of FR-TSVM, the gradient $\nabla f(\alpha)$ is a key to update the optimizer. Following the subdivision strategy, $\nabla f(\alpha)$ in Eq. (60) can also be decomposed as:

$$\nabla f(\alpha) = \overline{Q}_{i,A} \alpha_A + \overline{Q}_{i,\overline{A}} \alpha_{\overline{A}} - 1 \quad (61)$$

and only the gradient of those $i \in A$ would be paid attention to because they would effectively update the corresponding Lagrange multiplier. The gradients of those $i \notin A$ are never required to be recalculated and of course the updates of $u_+$ are no longer needed.

Here, we should note that the length-variable active set, chosen to handle constraints, is dynamically managed by the coordinate descent procedure. It should be kept in mind that a nonzero gradient is a necessary condition for bound and shrinking of the active set according to the original theorem proposed by Hsieh et al., [39].

**Theorem 3.** Taking the list $\{\alpha^{k,i}\}$ in the solution space, let $\ast$ be the final convergent point. The FR-TSVM sustains the following properties:

a). If $\alpha^*_{\ast} = 0$, and $\nabla_i f(\alpha^*) > 0$, there is a $\exists k_1$ such that

$$\forall k \geq k_1, \forall r, \alpha_{k,r} = 0.$$  

b). If $\alpha^*_{\ast} = c_3 s_{\ast i}$ and $\nabla_i f(\alpha^*) < 0$, there is a $\exists k_1$ such that

$$\forall k \geq k_1, \forall r, \alpha_{k,r} = c_3 s_{\ast i}.$$  

c). $\lim_{k \to \infty} \min_i \nabla_i^p f(\alpha^{k,i}) = \lim_{k \to \infty} \min_i \nabla_i^p f(\alpha^{k-1,i}) = 0.$

Proof. Referring to the proof in Appendix 7.3 in [39], imitations are taken to obtain Theorem 3.

Based on measures $\max_i \nabla_i^p f(\alpha^{k,i}) > 0, \min_i \nabla_i^p f(\alpha^{k,i}) < 0$ which are used to evaluate the solution violation level of a certain outer iteration $k$, a pair of bounds, $M^{k-1} = \max_i \nabla_i^p f(\alpha^{k-1,i})$ and $m^{k-1} = \min_i \nabla_i^p f(\alpha^{k-1,i})$, are asserted for bounding $\nabla_i f(\alpha^{k,i})$ at the $(k-1)$-th outer iteration. The assertion is used to seek a more specified
range for rejecting more inactive member from current A, in which members are allowed to participate in the optimization, at the k-th outer iteration. Basically, the active-set shrinkage relies on the pair of floating bounds to reject those violated participants. According to properties (a) and (b) of Theorem 3, the corresponding i is excluded from A to $\tilde{A}$ at the inner iteration for updating component $\alpha_i$ from $\alpha_{k,i}$ to $\alpha_{k,i+1}$ while next two conditions are hold:

$$
\alpha_{k,i} = 0 \text{ and } \nabla_i f(\alpha_{k,i}) > \bar{M}^{-k-1},
$$

$$
\alpha_{k,i} = c_3 s_{i-} \text{ and } \nabla_i f(\alpha_{k,i}) < \underline{m}^{-k-1},
$$

where

$$
\bar{M}^{-k-1} = \begin{cases} M^{k-1}, & \text{if } M^{k-1} > 0 \\ \infty, & \text{if } M^{k-1} \leq 0 \end{cases}
$$

and

$$
\underline{m}^{-k-1} = \begin{cases} m^{k-1}, & \text{if } m^{k-1} > 0 \\ -\infty, & \text{if } m^{k-1} \leq 0 \end{cases}
$$

We use the temporal $M$ and $m$ to catch maximal and minimal projected gradient in every cycle of inner iterations, and keep the maximal and minimal values in $\bar{M}$ and $\underline{m}$, respectively, to exclude inactive members in next outer iteration. According to property (c) of Theorem 3, bounds $\bar{M}$ and $\underline{m}$ would become closer after iterations, and would theoretically meet with each other finally:

$$
\bar{M}^k = \underline{m}^k \text{ if } k \to \infty.
$$

Although Eq. (66) shows the ideal condition for terminating the procedure, the exact meeting of $\bar{M}$ and $\underline{m}$ in the numerical iterations is difficult. An alternative allowing a sufficiently small gap $\epsilon$ and setting the following inequality:

$$
\bar{M}^k - \underline{m}^k < \epsilon
$$

for termination after k finite iterations is much more practical. While the gapped termination condition is reached, the optimal solution $\alpha^*$ of the sub-problem Eq. (69) is also possessed. Actually, this optimal $\alpha^*$ is only optimized for the sub-problem Eq. (69), not for the full problem Eq. (26). Hence, we ignore the current active set and set it to the full set $\{1, \ldots, l_\ast\}$ to get back all the $\alpha_i$ from the cached historical $\alpha_{k,i}^{\ast}$ in the final pass at the end of the procedure to ensure the recomposed $\alpha^*$ to fulfill Eq. (67).

The heuristic shrinking might raise a failure risk with the mismatch $\bar{M}$ and $\underline{m}$ even with a tolerable gapped mismatch, for example, $\bar{M} \leq 0$, or $\underline{m} \geq 0$. If such a condition happens, the whole FR-TSVM of Eq. (66) should be re-optimized with a different set of initial guests of $\alpha$. Additionally, the shrinking is in general applied heuristically in a fixed sequence of the $l_\ast$-dimensional gradients. A random update scheme performed a more rapid convergent rate than a sequential update scheme [38].

According to the separation of active from inactive set, $u_+$ defined in Eq. (65) can be re-expressed as:

$$
u_+ = -(Q_+ \alpha_A + Q_\tau \alpha_\tau),
$$

Algorithm 2 The optimization of FR-TSVM

1. Compute $Q = (H_T^T H_+ + c_1 E)^{-1} H_T^T$ and $\overline{Q}_{ii} = H_+ Q_i$
2. Let $A \leftarrow \{1, \ldots, l_\ast\}$
3. Given $\epsilon$ and initialized $\alpha_i \leftarrow 0, u_i \leftarrow 0$
4. Let $\bar{M} \leftarrow \infty$ and $\underline{m} \leftarrow -\infty$
5. while do
6. Let $M \leftarrow -\infty, m \leftarrow \infty$
7. for all $i \in A$ (a randomly and exclusively selected)
8. $\nabla_i f(\alpha) = -H_+ u_i - 1$
9. $\nabla_i f(\alpha) \leftarrow 0$
10. if $\alpha_i = 0$ then
11. if $\nabla_i f(\alpha) > \bar{M},$ then $A = A \setminus \{i\}$ end if
12. if $\nabla_i f(\alpha) < 0,$ then $\nabla_i f(\alpha) \leftarrow \nabla_i f(\alpha) \leftarrow \nabla_i f(\alpha)$ end if
13. else if $\alpha_i = c_3 s_{i-}$ then
14. if $\nabla_i f(\alpha) < \underline{m},$ then $A = A \setminus \{i\}$ end if
15. if $\nabla_i f(\alpha) > 0,$ then $\nabla_i f(\alpha) \leftarrow \nabla_i f(\alpha) \leftarrow \nabla_i f(\alpha)$ end if
16. else
17. $\nabla_i f(\alpha) \leftarrow \nabla_i f(\alpha)$
18. end if
19. $M \leftarrow \max(M, \nabla_i f(\alpha))$
20. $m \leftarrow \min(m, \nabla_i f(\alpha))$
21. if $\nabla_i f(\alpha) \neq 0$ then
22. $s_i \leftarrow \alpha_i$
23. $\alpha_i \leftarrow \min(\max(\alpha_i - \nabla_i f(\alpha)/Q_{ii}, 0), c_3 s_{i-})$
24. $u_{i+} \leftarrow u_i + Q_{ii}(\alpha_i - s_i)$
25. end if
26. end for
27. if $M - m < \epsilon$ then
28. if $A = \{1, \ldots, l_\ast\}$, break
29. else
30. $A \leftarrow \{1, \ldots, l_\ast\}, \bar{M} \leftarrow \infty, \underline{m} \leftarrow -\infty$
31. end if
32. if $M \leq 0$, then $\bar{M} \leftarrow \infty$. else $\bar{M} \leftarrow M$ end if
33. if $M \geq 0$, then $\underline{m} \leftarrow -\infty$. else $\underline{m} \leftarrow m$ end if
34. end while

which means that some elements coincided in the $\alpha_{i}\tau$ and $\overline{\alpha}_{i}\tau$ would remain the same before and after the update iteration $i$, and can be prevented in the update of $u_{i+} \leftarrow u_{i+} - Q_{ii}(\alpha_i - s_i)$. Algorithm 2 describes the update procedure of Algorithm 3 with active set shrinking in a random scheme.

4. Experiments

To validate the learning efficiency and generalization ability of FR-TSVM, some experiments are implemented on artificial datasets and publicly available benchmark datasets. All experiments are conducted in MATLAB (R2014a) on a PC with an Intel Core i7 processor (3.6GHz) with 32GB RAM. The execution time
and classification accuracy are fairly compared. Since the quadratic programming of SVM, TSVM, or FSVM has similar corresponding dual form, a Matlab optimization toolbox [44] is equally adapted for optimization. The proposed FR-TSVM is optimized by the coordinate decent with heuristic shrinking [2]. The Matlab code of FR-TSVM is released at https://github.com/gaobb/FR-TSVM.

The fuzzy membership assignments are different by utilizing Eq. (16) for linear kernel case and Eq. (22) for nonlinear kernel case. In addition, the fuzzy parameter $u = 0.1$ is set in Eq. (16) and Eq. (22). Under nonlinear case, Gaussian kernel $k(x_1, x_2) = \exp(-\|x_1 - x_2\|^2/g^2)$ is taken as kernel function, which in general outperforms other kinds of kernel functions. The model parameters $c_i(i = 1, 2, 3, 4)$ are carefully searched in the grids $\{2^i | i = -8, -7, \ldots, 8\}$ by setting $c_1 = c_2$ for TSVM, and $c_1 = c_2, c_3 = c_4$ for FR-TSVM. The grid-searching is conducted in 10-folds cross-validations, randomly selecting 30% of the whole samples for learning with the equivalent conditions mentioned above.

4.1. Experiments on artificial datasets

To intuitively validate the FR-TSVM’s classification performance, we firstly implement experiments on two 2-dimensional artificial datasets and compare the proposed method with standard SVM, FSVM, and TSVM.

The first dataset is artificial-generated Ripleys synthetic [15]. The Ripleys synthetic comprise 250 training instances and 1000 testing instances. We visualize the distribution of fuzzy membership value for training instances under linear and nonlinear case in Fig. 4 respectively. As shown in Fig. 4, compared to those instances locating near the class center, the fuzzy membership values of the instances which are far from their class center are always smaller.

Table 1 reports testing accuracies and learning CPU time of all methods with linear and Gaussian kernel. We can see that linear SVM and nonlinear FR-TSVM achieve the highest prediction accuracy under linear and nonlinear case, respectively. The reason for the outperformance of linear SVM is more likely from the dataset itself than from the classifier because there may be no noise existing on the Ripleys synthetic dataset. The noiseless fact of this testing dataset suppresses the outstanding ability of FR-TSVM in the experiment. The ability of FR-TSVM is confirmed if we examine the accuracy in the nonlinear classification. In terms of execution time, FR-TSVM shows its excellence in computational efficiency for both linear and nonlinear learning, especially linear case. The excellence manifests the remarkable potential of employing FR-TSVM for fast classification.

Fig. 4 shows linear and nonlinear decision hyperplanes produced by comparative methods with equivalent settings. In these panels, while standard SVM and FSVM produce only a single hyperplane (Fig. 4a and 4b), TSVM and FR-TSVM produce a pair of proximal hyperplanes (Fig. 4c and 4d) for class separation. Instead of single decision hyperplane in standard SVM and FSVM, a pair of nonparallel decision hyperplanes are used in TSVM and FR-TSVM. Comparing FR-TSVM with TSVM, the final decision hyperplane of FR-TSVM is more exact than one of TSVM. The fact reveals that FR-TSVM is more capable of producing an unbiased prediction than TSVM.

The second dataset is also a 2-dimensional artificial-generated dataset. In this dataset, the positive class of instances are generated by a uniform distribution satisfying $x_1 \in [-\pi/2, 2\pi]$, and $\sin(x_1) - 0.25 \leq x_2 \leq \sin(x_1) + 0.25$, while the negative class of instances consist of uniform points satisfying $x_1 \in [-\pi/2, 2\pi]$, and $0.6\sin(x_1/1.05 + 0.5) - 1.3 \leq x_2 \leq 0.6\sin(x_1/1.05 + 0.5) + 0.8$, where $x = [x_1, x_2]$. In our experiments, we generate 3000 instances according to the above rules. 600 instances are randomly chosen for training, and the remaining 2400 for testing.

In Fig. 5, the first row shows the results of the nonlinear SVM, FSVM, TSVM and the proposed FR-TSVM on this dataset. We can see that the separating hyperplanes of these classifiers obtain similar results. To further validate the robust performance of the FR-TSVM, we add a noise $\varepsilon \in \mathcal{N}(0,0.2)$ for each training instance of this dataset, where $\mathcal{N}(0,0.2)$ is a normal distribution with mean 0 and variance 0.2. In Fig. 5 the second row shows the results of the nonlinear SVM, FSVM, TSVM and the proposed FR-TSVM on this dataset with noise. It can be seen that the proposed FR-TSVM can still effectively determine the separating hyperplane compared with the other three methods, especially TSVM. This indicates that the FR-TSVM can effectively alleviate the effect of noise points.

Table 2 reports prediction results including classification accuracy and training time with/without noises. We can see that all the methods obtain similarity test accuracies without noises, while our FR-TSVM achieves the highest test accuracy. Under noise case, FR-TSVM and FSVM obtain better classification accuracies than TSVM and SVM, respectively. This indicates that fuzzy membership takes an important role in training model with noisy
instances. It is noteworthy that T SVM's test accuracy is reduced by almost 3% (from 99.88% to 97.00%), whereas FR-TSVM has only a slight slowdown (less than 1%). It suggests that FR-TSVM is more robust than TSVM when training instances consist of noisy data. As for the training time, FR-TSVM and TSVM are more efficient than SVM and FSVM, while FR-TSVM is the fastest among all methods.

4.2. Experiments on benchmark datasets

To further examine the performance of FR-TSVM, 13 common datasets are gathered from the public UCI machine learning datasets [60]. To adjust values measured on different scales to a notionally common scale in the input space, all input features are normalized and scaled-down within [0,1]. The examination mainly focuses on binary classification. Two adversary classes are formed in every dataset. For the multi-classification datasets, we convert them into binary classification datasets by taking the majority class as the first class and gathering all the remainders together as the adversary class. Table 3 lists the statistics of datasets.

Table 3 and 4 report the learning results of these algorithms with linear and Gaussian kernels. To show the optimization cost, a particular quadratic programming time is recorded. To assess the performance, 10-folds cross-validation, as that in Section 4 is taken. It means every classifier is repeatedly validated in the datasets with a ratio of 90%/10% for respective training/testing phase. Ten characteristics values are collected and averaged for the assessment. In the experiments, a standard deviation of the ten classification accuracies is provided in addition to the average to reflect the classification robustness.

Comparing to baselines including SVM, FSVM and TSVM, the FR-TSVM produces more competitive and robust performances (e.g., high mean and low deviation of accuracies) on most datasets in Table 3 and 4. This suggests it is important to improve the classification performance introducing the fuzzy membership. Moreover, the tables show a higher improvement using nonlinear kernel than that of a linear kernel. This is due to the intrinsic difference between the kernels. As we know, a nonlinear Gaussian kernel commonly pertains a superior classification ability due to its flexibility. It meets the general concept of the classification. In addition, FR-TSVM obtains slightly lower performance than other methods on a few datasets (e.g, Bupa and Pima). A possible rea-
son is that these datasets may not have outliers. Fig. 6 shows the time cost of training on 13 UCI datasets. It is seen that the training time of TSVM and FR-TSVM are shorter than those of SVM and FSVM. This result is not surprising because TSVM and FR-TSVM are solved by two small QPPs rather than one large QPP in SVM and FSVM. Compared to the training time of TSVM, our FR-TSVM are faster. In short, the results in Table 4 and 5 indicate, FR-TSVM effectively improves the classification accuracy and efficiently reduces training time compared to the traditional methods. The excellence strongly reflects that the classifier is potential for future applications.

4.3. Parameter analysis

FR-TSVM’s performance may be affected by hyperparameters. The hyperparameters corresponding to the investigation are those  \( c_1 = c_2 = c_3 = c_4, u, \) and  \( g \) in Gaussian kernel for nonlinear case. Here, we take Ripleys synthetic dataset for an example.

Table 3: Detailed characteristics of the benchmark datasets.  \( l, \ l_+, \) and  \( l_- \) are the number of all instances, positive instances and negative instances, respectively.  \( n \) is the dimension of feature.

| Dataset  | #l | #l_+ | #l_- |  \( n \) |
|----------|----|------|------|--------|
| Breast   | 106| 22   | 84   | 9      |
| Ionosphere| 351| 126  | 225  | 34     |
| Iris     | 150| 100  | 50   | 4      |
| Australian| 690| 307  | 383  | 14     |
| WDBC     | 569| 357  | 212  | 30     |
| Wine     | 178| 119  | 59   | 13     |
| Hepatitis| 155| 123  | 32   | 19     |
| WPBC     | 198| 46   | 148  | 33     |
| Bupa     | 345| 200  | 145  | 6      |
| Sonar    | 208| 111  | 97   | 60     |
| Glass    | 214| 138  | 76   | 10     |
| Heart    | 270| 120  | 150  | 13     |
| Pima     | 768| 268  | 500  | 8      |

Table 4: Comparison results (mean±std) of linear SVM, FSVM, TSVM and FR-TSVM on benchmark datasets. The best result of each row is marked in bold.

| Dataset  | SVM   | FSVM  | TSVM  | FR-TSVM |
|----------|-------|-------|-------|---------|
| Breast   | 96.33±4.77 | 97.17±4.58 | 97.17±4.58 | 97.17±4.58 |
| Ionosphere| 85.53±6.48 | 85.75±6.06 | 82.33±5.18 | 87.19±4.22 |
| Iris     | 100.0±0.00 | 100.0±0.00 | 100.0±0.00 | 100.0±0.00 |
| Australian| 84.92±4.53 | 85.50±4.59 | 85.07±4.77 | 85.93±4.39 |
| WDBC    | 95.34±5.17 | 95.87±3.21 | 93.84±5.86 | 96.39±3.52 |
| Wine    | 98.89±2.34 | 98.89±2.34 | 98.89±2.34 | 98.89±2.34 |
| Hepatitis| 79.26±8.76 | 84.94±7.29 | 75.38±12.50 | 85.77±7.21 |
| WPBC    | 79.93±9.49 | 74.24±10.35 | 76.88±7.01 | 77.96±10.03 |
| Bupa    | 66.36±6.04 | 67.51±7.36 | 61.72±5.96 | 64.38±6.24 |
| Sonar   | 74.08±8.96 | 77.46±7.14 | 72.15±7.48 | 78.34±8.29 |
| Glass   | 70.41±10.01 | 74.18±11.01 | 67.64±11.02 | 81.17±13.56 |
| Heart   | 82.22±5.18 | 82.59±3.51 | 84.07±4.95 | 84.07±6.06 |
| Pima    | 77.21±3.75 | 73.65±4.22 | 70.95±3.37 | 75.13±3.78 |

Fig. 7 further shows the test accuracy with varying the adjustable parameters on artificial dataset Ripleys synthetic. Despite artificial dataset, two points are thus eventually asserted from the investigation. First, each adjustable parameter affects significantly the FR-TSVM’s performance, especially Gaussian kernel parameter  \( g \). Second, the inconsistent trends of the accuracy with respect to the parameters reveal that the issue of parameter selection is still a challenge for us. We thus leave the issue as a future study.

5. Conclusion

In this paper, we propose a FR-TSVM algorithm for binary classification based on FSVM, TSVM and coordinate descent methods. First, similar to FSVM classifier, the embedded fuzzy concept enhances noise-resistance capability and generalization ability. However, the fuzzy membership construction is different from that of the FSVM. Our method can effectively assign the fuzzy membership...
value to different instances by distinguishing the support vectors and the outliers. Second, as TSVM, the FR-TSVM finds a pair of nonparallel hyperplanes through two smaller sized QPPs rather than one large-sized QPP in the SVM or FSVM. As we all know, the dual problems of the TSVM may be ill-conditioned, but the proposed model’s dual form has been brought to a pair of convex quadratic programming problems and confirmed it is capable of solution uniqueness and singularity avoidance. Third, a novel coordinate descent method with shrinking is developed to solve the dual problems. Compared to TSVM, our FR-TSVM is not only faster but also needs less memory storage. This indicates that our FR-TSVM is very suitable for large-scale data. Experiments with simulated and realistic datasets reveal that an exceedingly high classification accuracy with less computation time is achieved using both linear or nonlinear FR-TSVM. However, there are 6 parameters (i.e., $c_1, c_2, c_3, c_4, \mu, g$) in our FR-TSVM, so the parameters selection is a practical problem and should be addressed in the future. Also, it is an interesting direction to extend FR-TSVM to more recognition problems such as multi-class or multi-label problem.

References

[1] V. Vapnik, Statistical learning theory, Wiley New York, 1998.

[2] A. Widodo, B.-S. Yang, Support vector machine in machine condition monitoring and fault diagnosis, Mechanical Systems and Signal Processing 21 (6) (2007) 2560–2574.

[3] B. Heisele, P. Ho, T. Poggio, Face recognition with support vector machines: Global versus component-based approach, in: Proc. IEEE Int’l Conf. Computer Vision, IEEE, 2001, pp. 688–694.

[4] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in: European Conference on Machine Learning, Springer, 1998, pp. 137–142.

[5] I. El-Naqa, Y. Yang, M. N. Wernick, N. P. Galatsanos, R. M. Nishikawa, A support vector machine approach for detection of microcalcifications, IEEE Trans. on Medical Imaging 21 (12) (2002) 1552–1563.

[6] T. B. Trafalis, H. Ince, Support vector machine for regression and applications to financial forecasting, in: Proc. Int’l Joint Conf. on Neural Networks, 2000, pp. 348–353.

[7] O. L. Mangasarian, E. W. Wild, Multisurface proximal support vector machine classification via generalized eigenvalues, IEEE Trans. on Pattern Analysis and Machine Intelligence 28 (1) (2006) 69–74.
