A Fast and Robust TSVM for Pattern Classification

Bin-Bin Gao[a], Jian-Jun Wang[b,]

[a] *YouTu Lab, Tencent, Shenzhen 518000, P.R. China.*
[b] *School of Mathematics and Statistics, Southwest University, Chongqing 400715, P.R. China.*

Abstract

Twin support vector machine (TSVM) is a powerful learning algorithm by solving a pair of smaller SVM-type problems. However, there are still some specific issues such as low efficiency and weak robustness when it is faced with some real applications. In this paper, we propose a Fast and Robust TSVM (FR-TSVM) to deal with the above issues. In order to alleviate the effects of noisy inputs, 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 separating hyperplanes. 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, theoretical foundations of the proposed model 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: `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 [1], is a very popular method in machine learning algorithms. By adopting the principle of structural risk minimization for generalization, SVM is capable of handling excellently with classifications and regressions of many engineering problems, such as machine fault-diagnosis [2], image identification [4], text classification [5], biomedicine [6] and financial forecast [7], *etc.* and had become a prominent highlight of machine learning research. However, the application of SVM is occasionally restricted by some practical issues, especially computational speed and noisy inputs. To overcome the limitations, varieties of solutions have been discovered and studied.

Mangasarian *et al.* [9] changed the proximal planes which are originally parallel to each other to generate a maximal spaced separating hyperplane into those nonparallel ones, and proposed a generalized eigenvalue proximal SVM (GEPSVM). By solving a pair of simple generalized eigenvalue problems, the GEPSVM can speed-up the training speed with slighter accurate performance. Following this concept, Jayadeva *et al.* [10] proposed the twin SVM (TSVM). The objectives of TSVM 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 algorithm of TSVM converted GEPSVM into a pair of SVM-like convex programs. Due to the progress of solving the small-sized SVM-type problems, the cost of the computation time was thus reduced satisfactorily to one fourth of that of a conventional SVM. Recently, numerous non-parallel proximal-plane learning methods were proposed as offshoots of the TSVM concept, such as TBSVM [11], Structural TSVM [12], Robust TSVM [13], Laplacian Smooth TSVM [14], Least-square TSVM(Ls-TSVM) [15], Ls-TSVM for multi-class [16], Twin Parametric-margin SVM [17, 18], Multi-label TSVM [19], RPTWSVM [20] and Pin-TSVM [21]. In additional, some regression model based TSVM are also proposed such as TSVR [22], TPSVR[23], ϵ-TSVR [24], TWSVR [25] and v-TWSVR [26]. All of the proposed variants share the same merits of the TSVM. However, one challenge is that the training instances from the real-world applications often carry information with a significant noise. These TSVM methods are sensitive to outliers or noises.

The noise data often can deteriorate the SVM and TSVM generalization ability. From the technical viewpoint, the term of *noisy information* can be causally converted into the *fuzzy information* to meet the theory of fuzzy inference. The training of SVM would be too sensitive to the noisy inputs if all the training instances are treated equivalently at the training stage. A conceptual way to alleviate the sensitive deterioration is to contract the influence of the noisy inputs. This means that a conventional SVM, which intrinsically treats every input in-

Email addresses: `csgaobb@gmail.com` (Bin-Bin Gao), `wjj@swu.edu.cn` (Jian-Jun Wang)

stance in equivalence, can be improved by introducing the fuzzy membership functions to soften the input information. A category of fuzzy SVMs are hence developed, such as Lin and Wang [27, 28], Wu and Liu [29], Inoue and Abe [30], Yang *et al.* [31], and Tang [32] *etc.* The elementary concept of fuzzy SVM is to allocate a small confident membership for each noise 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 the noise instances and leads to a robust classifier. To utilize the fuzzy concept, variants of fuzzy TSVM have been developed in [33, 34]. However, these fuzzy SVM methods equivalently assigned the fuzzy membership to each instance and they cannot distinguish the support vectors and the outliers effectively.

From the viewpoint of computational efficiency, the plenty of input instances from the real-world applications require abundant quadratic programming computations, which slow down the training efficiency. This computational cost significantly limits the applications of SVM. There are various solutions for improvement, including *chunking* [35], *decomposition* [36], *sequential minimal optimization* (SMO) [37] *etc.* However, they need to optimize the entire set of non-zero Lagrangian multipliers, and the generated kernel matrix may still be too large to adapt to memory. Recently, Chang *et al.* proposed a primal *coordinate descent* (CD) method to deal with the large-scale linear SVM [38, 39]. The CD method showed a straightforward merit in computational efficiency [40].

In this paper, we propose a fast and robust twin support vector machine (FR-TSVM) for classification problem. First, in order to decrease the effect of outliers, we construct a novel fuzzy membership function. Each training instance is assigned a fuzzy membership according to the structural information of training instances in the input space or the feature space. Then, we embed the fuzzy concept into TSVM and formulate a robust twin SVM model. In FR-TSVM, different input instances can make different contributions to the learning of decision hyperplanes. Hence, the proposed model can effectively alleviate the effect of the noisy instances and obtain the robust performance. Finally, we develop further the coordinate descent algorithm with the shrinking to speed-up the computations of the linear and nonlinear FR-TSVM. In addition, this paper presents the FR-TSVM in details including corresponding theoretical fundamentals, related properties and optimization algorithms. Experiments on artificial and benchmark datasets show that the proposal brings more satisfactory performances on both classification accuracy and learning efficiency, compared with the traditional SVM, FSVM and TSVM.

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

2. Background

In this paper, we mainly focus on a binary classification problem in the n-dimensional space $\mathcal{R}^n$. We denote the set of training data as $D = \{(\boldsymbol{x}_i, y_i)|i = 1, 2, \ldots, l\}$, where $\boldsymbol{x}_i \in \mathcal{R}^n$ 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 the all training instances, and the matrices $X_\pm$ with sizes of $l_\pm \times n$ represent the training instances belonging to classes "+1" and "-1", 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:

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

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

$$\begin{aligned} &\min_{\boldsymbol{w},b,\xi_i} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + c\sum_{i=1}^{l}\xi_i \\ &\text{s.t.} \ \ y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) + \xi_i \geq 1, \xi_i \geq \boldsymbol{0}, i = 1, 2, \ldots, l. \end{aligned} \tag{2}$$

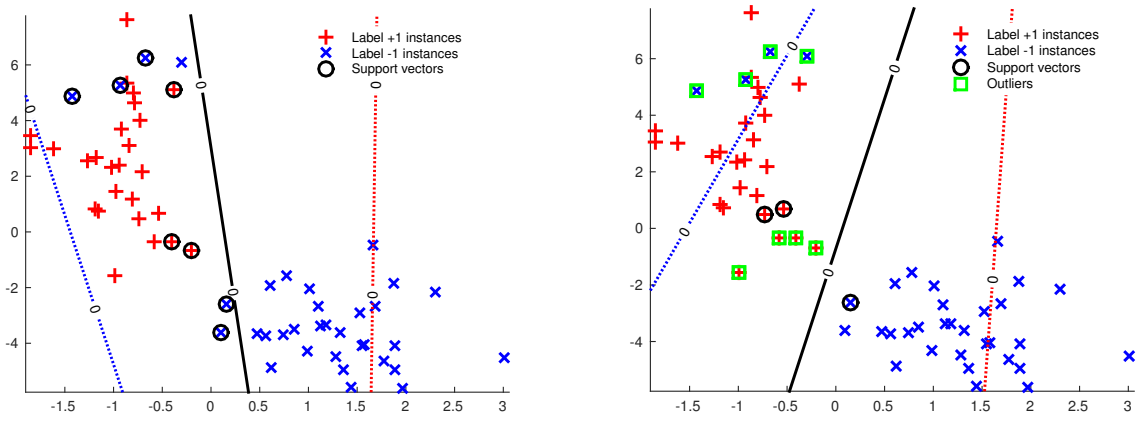
where $\|.\|$ stands for the ℓ_2-norm, ξ_i is called as the slack variable which denotes the misclassification error associated with the i-th input instance and $c > 0$ is the regularization factor that balances the importance between the maximization of the margin width (*i.e.*, the minimization of $\frac{1}{2}\|\boldsymbol{w}\|^2$) and the minimization of the training error. The dual QPP of the problem Eq.(2) is:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \ \sum_{i=1}^{l}\alpha_i + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \alpha_i \alpha_j \\ &\text{s.t.} \ \ \sum_{i=1}^{l}\alpha_i y_i = 0, 0 \leq \alpha_i \leq 1, i = 1, 2, \ldots, l\,, \end{aligned} \tag{3}$$

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

$$f(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x} + b^*) \tag{4}$$

where $(\boldsymbol{w}^*, b^*)$ and α_i^* are the solution of Eq.(2) and Eq.(3), respectively.



(a) TSVM (b) FR-TSVM

Figure 1: Geometric interpretation of linear TSVM and FR-TSVM for binary classification.

2.2. Twin support vector machine

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

$$\boldsymbol{w}_+^{\mathrm{T}}\boldsymbol{x} + b_+ = 0 \text{ and } \boldsymbol{w}_-^{\mathrm{T}}\boldsymbol{x} + b_- = 0\,. \tag{5}$$

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

$$\begin{aligned} \min_{\boldsymbol{w}_+, b_+, \boldsymbol{\xi}_-} \quad & \frac{1}{2} \parallel X_+\boldsymbol{w}_+ + \boldsymbol{e}_+ b_+ \parallel^2 + c_1 \boldsymbol{e}_-^{\mathrm{T}} \boldsymbol{\xi}_- \\ \text{s.t.} \quad & -(X_-\boldsymbol{w}_+ + \boldsymbol{e}_- b_+) + \boldsymbol{\xi}_- \geq \boldsymbol{e}_-, \boldsymbol{\xi}_- \geq \boldsymbol{0}\,, \end{aligned} \tag{6}$$

and

$$\begin{aligned} \min_{\boldsymbol{w}_-, b_-, \boldsymbol{\xi}_+} \quad & \frac{1}{2} \parallel X_-\boldsymbol{w}_- + \boldsymbol{e}_- b_- \parallel^2 + c_2 \boldsymbol{e}_+^{\mathrm{T}} \boldsymbol{\xi}_+ \\ \text{s.t.} \quad & (X_+\boldsymbol{w}_- + \boldsymbol{e}_+ b_-) + \boldsymbol{\xi}_+ \geq \boldsymbol{e}_+, \boldsymbol{\xi}_+ \geq \boldsymbol{0}\,, \end{aligned} \tag{7}$$

where $c_1 > 0$ and $c_2 > 0$ are parameters, $\boldsymbol{\xi}_+$ and $\boldsymbol{\xi}_-$ denote the vectors of slack variables for positive and negative classes, respectively, and $\boldsymbol{e}_+, \boldsymbol{e}_-$ correspond to unit vectors with $l_\pm$ dimensions. By introducing the Lagrangian multipliers, the dual QPPs of Eq.(6) and Eq.(7) can be represented as followings

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{e}_-^{\mathrm{T}}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}} H_-(H_+^{\mathrm{T}}H_+)^{-1}H_-^{\mathrm{T}}\boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{0} \leq \boldsymbol{\alpha} \leq c_1\boldsymbol{e}_-, \end{aligned} \tag{8}$$

$$\begin{aligned} \max_{\boldsymbol{\beta}} \quad & \boldsymbol{e}_+^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}} H_+(H_-^{\mathrm{T}}H_-)^{-1}H_+^{\mathrm{T}}\boldsymbol{\beta} \\ \text{s.t.} \quad & \boldsymbol{0} \leq \boldsymbol{\beta} \leq c_2\boldsymbol{e}_+, \end{aligned} \tag{9}$$

where $H_+ = [X_+, \boldsymbol{e}_+], H_- = [X_-, \boldsymbol{e}_-]$. The non-parallel hyperplanes Eq.(5) can be obtained from the solutions $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ of Eq.(8) and Eq.(9) by

$$\boldsymbol{u}_+^* = -(H_+^{\mathrm{T}}H_+)^{-1}H_-^{\mathrm{T}}\boldsymbol{\alpha}^*, \boldsymbol{u}_-^* = (H_-^{\mathrm{T}}H_-)^{-1}H_+^{\mathrm{T}}\boldsymbol{\beta}^*, \tag{10}$$

where $\boldsymbol{u}_\pm^* = [\boldsymbol{w}_\pm^{*\mathrm{T}}, b_\pm^*]^{\mathrm{T}}$, $w_\pm^{*\mathrm{T}}$ and $b_\pm^*$ are the solutions of Eq.(6) and Eq.(7). TSVM then can easily assign a label "+1" or "-1" to a testing instance $\boldsymbol{x}$ by

$$f(\boldsymbol{x}) = \operatorname*{argmin}_{\pm} \frac{\mid \boldsymbol{w}_\pm^{*\,\mathrm{T}}\boldsymbol{x} + b_\pm^* \mid}{\parallel \boldsymbol{w}_\pm^* \parallel}\,, \tag{11}$$

where $\mid \cdot \mid$ is a function taking its absolute value. If the matrix $H_+^{\mathrm{T}}H_+$ or $H_-^{\mathrm{T}}H_-$ 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 λ 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) [27]. In FSVM, training instance $\boldsymbol{x}_i$ was assigned a fuzzy membership $0 \leq s_i \leq 1$ besides a label $y_i \in \{+1, -1\}$. The input dataset T is thus modified as $T' = \{(\boldsymbol{x}_i, y_i, s_i) | i = 1, 2, \ldots, l\}$. These fuzzy memberships $\{s_i | i = 1, 2, \ldots, l\}$ are used to reduce the influence of the noisy instances for generating the decision function, which induces the fuzzy SVM as follow:

$$\begin{aligned} \min_{\boldsymbol{w}, b, \xi} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + c\sum_{i=1}^{l} s_i\xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) + \xi_i \geq 1, \xi_i \geq 0, i = 1, 2, \ldots, l. \end{aligned} \tag{12}$$

It is noted that a smaller s_i can reduce the effect of the parameter ξ_i in problem Eq.(12) so that the corresponding instance $\boldsymbol{x}_i$ can be treated as less important. The classification of any testing instance $\boldsymbol{x}$ can be obtained by determining the sign of $\boldsymbol{w}^{*\mathrm{T}}\boldsymbol{x} + b^*$ where $\boldsymbol{w}^*$ and b^* are the solution of Eq.(12).

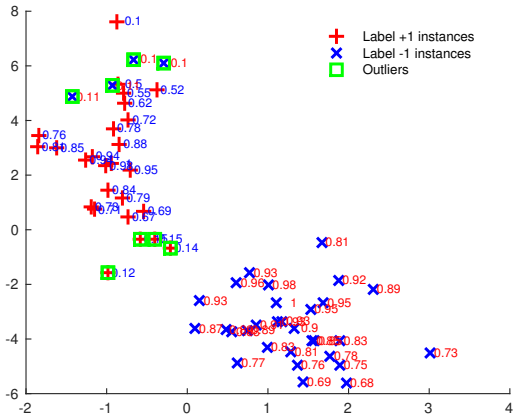


Figure 2: Linear fuzzy membership value for binary classification. The red color and blue color numbers are fuzzy membership value for negetive and postive instances, respectively.

3. The proposed FR-TSVM approach

In this section, we will propose the approach of FR-TSVM. To this end, we first introduce the fuzzy membership construction. Then, we propose the 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, the support vectors geometrically locate near the boundary area of two adjacent classes. The noise data points reside in this area typically and unfortunately. This means that support vectors and noisy data points are frequently mixed together. Inspired by Tang [32], 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 the support vectors.

3.1.1. Linear case

The construction of fuzzy membership considers firstly linear kernel case, and defines directly positive and negative class centers, $\boldsymbol{x}_{c+}$ and $\boldsymbol{x}_{c-}$, as the mean points in the input space for the categorized positive and negative instances,

$$\boldsymbol{x}_{c+} = \frac{1}{l_+}\sum_{y_i=+1}\boldsymbol{x}_i\,, \boldsymbol{x}_{c-} = \frac{1}{l_-}\sum_{y_i=-1}\boldsymbol{x}_i\,. \tag{13}$$

By measuring the distance of farthest scattering positive and negative instances, hypersphere radii, r_+ and r_-, for respective classes are defined and given as

$$r_+ = \max\{\parallel \boldsymbol{x}_i - \boldsymbol{x}_{c+} \parallel | y_i = +1\}\,, \tag{14}$$

$$r_- = \max\{\parallel \boldsymbol{x}_i - \boldsymbol{x}_{c-} \parallel | y_i = -1\}\,. \tag{15}$$

With known $\boldsymbol{x}_{c+}, \boldsymbol{x}_{c-}$, r_+, and r_-, membership s_i of an instance can be assigned according to the relationship between $\parallel \boldsymbol{x}_i - \boldsymbol{x}_{c+} \parallel$ and $\parallel \boldsymbol{x}_i - \boldsymbol{x}_{c-} \parallel$. For example, s_i of a positive instance can be given as:

$$s_i = \begin{cases} \mu\big(1 - \parallel \boldsymbol{x}_i - \boldsymbol{x}_{c+} \parallel/(r_+ + \delta)\big), \\ \quad \text{if } \parallel \boldsymbol{x}_i - \boldsymbol{x}_{c+} \parallel \geq \parallel \boldsymbol{x}_i - \boldsymbol{x}_{c-} \parallel \ \&\ y_i = +1 \\ (1-\mu)\big(1 - \parallel \boldsymbol{x}_i - \boldsymbol{x}_{c+} \parallel/(r_+ + \delta)\big), \\ \quad \text{if } \parallel \boldsymbol{x}_i - \boldsymbol{x}_{c+} \parallel < \parallel \boldsymbol{x}_i - \boldsymbol{x}_{c-} \parallel \ \&\ y_i = +1 \end{cases} \tag{16}$$

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.* $\parallel \boldsymbol{x}_i - \boldsymbol{x}_{c+} \parallel \geq \parallel \boldsymbol{x}_i - \boldsymbol{x}_{c-} \parallel$). In order to decrease outliers' effect toward the hyperplane, we assign a small positive real number for μ. Therefore, we set $u = 0.1$ for FR-TSVM in practice. Similar rule can be applied to the fuzzy membership of those negative instances. As shown in Fig. 2, we intuitively show linear fuzzy membership value for training instances.

3.1.2. Nonlinear case

For a nonlinear kernel case, the fuzzy membership function must be consistently reconstructed in the feature space $\mathcal{H}$. Similar to the linear case, the positive and negative class centers $\boldsymbol{\varphi}_{c+}$ and $\boldsymbol{\varphi}_{c-}$ in the feature space are defined as:

$$\boldsymbol{\varphi}_{c+} = \frac{1}{l_+}\sum_{y_i=+1}\varphi(\boldsymbol{x}_i)\,, \boldsymbol{\varphi}_{c-} = \frac{1}{l_-}\sum_{y_i=-1}\varphi(\boldsymbol{x}_i)\,. \tag{17}$$

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

$$\begin{aligned} &\parallel \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c+} \parallel^2 \\ &= \|\varphi(\boldsymbol{x}_i)\|^2 - 2\langle\varphi(\boldsymbol{x}_i), \boldsymbol{\varphi}_{c+}\rangle + \|\boldsymbol{\varphi}_{c+}\|^2 \\ &= \langle\varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_i)\rangle - \frac{2}{l_+}\sum_{y_j=+1}\langle\varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j)\rangle \\ &\quad + \frac{1}{l_+^2}\sum_{y_i=+1}\sum_{y_k=+1}\langle\varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_k)\rangle \\ &= \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \frac{2}{l_+}\sum_{y_j=+1}\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &\quad + \frac{1}{l_+^2}\sum_{y_i=+1}\sum_{y_k=+1}\kappa(\boldsymbol{x}_i, \boldsymbol{x}_k), \end{aligned} \tag{18}$$

and

$$\begin{aligned}&\| \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c-} \|^2 \\ &= \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \frac{2}{l_-}\sum_{y_j=-1}\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &\quad + \frac{1}{l_-^2}\sum_{y_i=-1}\sum_{y_k=-1}\kappa(\boldsymbol{x}_i, \boldsymbol{x}_k),\end{aligned} \tag{19}$$

where $\kappa(\boldsymbol{x}_i.\boldsymbol{x}_j)$ is the kernel function which implicitly calculates the high-dimensional dot-product of $\varphi(\boldsymbol{x}_i)$ and $\varphi(\boldsymbol{x}_j)$. The scattering hypersphere radii in the feature space are

$$r_{\varphi+} = \max\{\| \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c+} \| \, | y_i = +1\}, \tag{20}$$

$$r_{\varphi-} = \max\{\| \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c-} \| \, | y_i = -1\}. \tag{21}$$

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

$$s_i = \begin{cases} \mu\big(1 - \sqrt{\| \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c+} \|^2/(r_{\varphi+}^2 + \delta)}\big), \\ \text{if } \| \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c+} \| \geq \| \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c-} \| \ \& \ y_i = +1 \\ \big(1-\mu\big)\big(1 - \sqrt{\| \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c+} \|^2/(r_{\varphi+}^2 + \delta)}\big), \\ \text{if } \| \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c+} \| < \| \varphi(\boldsymbol{x}_i) - \boldsymbol{\varphi}_{c-} \| \ \& \ y_i = +1 \end{cases} \tag{22}$$

where δ is similarly defined as a small positive constant to avoid the vanishing of s_i. Of course, a fuzzy membership function s_i for a 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 the fast and robust twin support vector machine (FR-TSVM). As mentioned earlier, the FR-MSVM is similar to the TSVM, as it also derives a pair of non-parallel planes through two QPPs. What is more, FR-TSVM is more robust than TSVM.

3.2.1. Linear FR-TSVM

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

$$\boldsymbol{w}_+^{\mathrm{T}}\boldsymbol{x} + b_+ = 0, \quad \textit{and} \quad \boldsymbol{w}_-^{\mathrm{T}}\boldsymbol{x} + b_- = 0. \tag{23}$$

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

$$\begin{aligned}\min_{\boldsymbol{w}_+, b_+, \boldsymbol{\xi}_-} \quad & \frac{1}{2}c_1 \| \boldsymbol{w}_+ \|^2 + \frac{1}{2} \| X_+\boldsymbol{w}_+ + \boldsymbol{e}_+ b_+ \|^2 + c_3 \boldsymbol{s}_-^{\mathrm{T}}\boldsymbol{\xi}_- \\ \text{s.t.} \quad & -(X_-\boldsymbol{w}_+ + \boldsymbol{e}_- b_+) + \boldsymbol{\xi}_- \geq \boldsymbol{e}_-, \ \boldsymbol{\xi}_- \geq \boldsymbol{0},\end{aligned} \tag{24}$$

$$\begin{aligned}\min_{\boldsymbol{w}_-, b_-, \boldsymbol{\xi}_+} \quad & \frac{1}{2}c_2 \| \boldsymbol{w}_- \|^2 + \frac{1}{2} \| X_-\boldsymbol{w}_- + \boldsymbol{e}_- b_- \|^2 + c_4 \boldsymbol{s}_+^{\mathrm{T}}\boldsymbol{\xi}_+ \\ \text{s.t.} \quad & (X_+\boldsymbol{w}_- + \boldsymbol{e}_+ b_-) + \boldsymbol{\xi}_+ \geq \boldsymbol{e}_+, \ \boldsymbol{\xi}_+ \geq \boldsymbol{0},\end{aligned} \tag{25}$$

where $c_i > 0 (i = 1, 2, 3, 4)$ are trade-off parameters for weighting the regularization, $\boldsymbol{\xi}_+$ and $\boldsymbol{\xi}_-$ denote the subsets of misclassification error for positive and negative classes respectively, both $\boldsymbol{s}_+ \in R^{l_+}$ and $\boldsymbol{s}_- \in R^{l_-}$ are the fuzzy-number vectors sequentially associated with the positive and negative input instances, and $\boldsymbol{e}_+, \boldsymbol{e}_-$ 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 the samples to its corresponding hyperplane. An intuitive geometric interpretation for the linear FR-TSVM is shown in Fig. 1b.

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

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \quad & \boldsymbol{e}_-^{\mathrm{T}}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}} H_-(H_+^{\mathrm{T}}H_+ + c_1E_1)^{-1}H_-^{\mathrm{T}}\boldsymbol{\alpha} \\ s.t. \quad & \boldsymbol{0} \leq \boldsymbol{\alpha} \leq c_3\boldsymbol{s}_-,\end{aligned} \tag{26}$$

$$\begin{aligned}\max_{\boldsymbol{\beta}} \quad & \boldsymbol{e}_+^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}} H_+(H_-^{\mathrm{T}}H_- + c_2E_2)^{-1}H_+^{\mathrm{T}}\boldsymbol{\beta} \\ s.t. \quad & \boldsymbol{0} \leq \boldsymbol{\beta} \leq c_4\boldsymbol{s}_+,\end{aligned} \tag{27}$$

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

$$\begin{aligned}\boldsymbol{u}_+^* &= -(H_+^{\mathrm{T}}H_+ + c_1E_1)^{-1}H_-^{\mathrm{T}}\boldsymbol{\alpha}^*, \\ \boldsymbol{u}_-^* &= (H_-^{\mathrm{T}}H_- + c_2E_2)^{-1}H_+^{\mathrm{T}}\boldsymbol{\beta}^*,\end{aligned} \tag{28}$$

where $\boldsymbol{u}_\pm^* = [\boldsymbol{w}_\pm^{*\mathrm{T}}, b_\pm^*]^{\mathrm{T}}$, $\boldsymbol{\alpha}^*$ *and* $\boldsymbol{\beta}^*$ *denote the optimal values of* $\boldsymbol{\alpha}$ *and* $\boldsymbol{\beta}$, *respectively.*

Proof. Taking Lagrangian of the primal problem Eq.(24), the problem becomes:

$$\begin{aligned}\mathrm{L}(\boldsymbol{w}_+, b_+, \boldsymbol{\xi}_-) = &\frac{1}{2}c_1 \| \boldsymbol{w}_+ \|^2 + \frac{1}{2} \| X_+\boldsymbol{w}_+ + \boldsymbol{e}_+ b_+ \|^2 \\ &+ \alpha^{\mathrm{T}}(X_-\boldsymbol{w}_+ + \boldsymbol{e}_- b_+ - \boldsymbol{\xi}_- + \boldsymbol{e}_-) \\ &+ c_3\boldsymbol{s}_-^{\mathrm{T}}\boldsymbol{\xi}_- - \boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\xi}_-,\end{aligned} \tag{29}$$

where non-negative $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ are Lagrange multipliers. According to the KKT conditions, we have

$$\nabla_{\boldsymbol{w}_+} L = c_1\boldsymbol{w}_+ + X_+^{\mathrm{T}}(X_+^{\mathrm{T}}\boldsymbol{w}_+ + \boldsymbol{e}_+b_+) + X_-^{\mathrm{T}}\boldsymbol{\alpha} = 0, \quad (30)$$

$$\nabla_{b_+} L = \boldsymbol{e}_+^{\mathrm{T}}(X_+^{\mathrm{T}}\boldsymbol{w}_+ + \boldsymbol{e}_+b_+) + \boldsymbol{e}_-^{\mathrm{T}}\boldsymbol{\alpha} = 0, \quad (31)$$

$$\nabla_{\boldsymbol{\xi}_-} L = c_3\boldsymbol{s}_- - \boldsymbol{\alpha} - \boldsymbol{\eta} = 0, \quad (32)$$

$$-(X_-\boldsymbol{w}_+ + \boldsymbol{e}_-b_+) + \boldsymbol{\xi}_- \geq \boldsymbol{e}_-, \boldsymbol{\xi}_- \geq \boldsymbol{0}, \quad (33)$$

$$\boldsymbol{\alpha}^{\mathrm{T}}(X_-\boldsymbol{w}_+ + \boldsymbol{e}_-b_+ - \boldsymbol{\xi}_- + \boldsymbol{e}_-) = 0, \boldsymbol{\eta}^{\mathrm{T}}\boldsymbol{\xi}_- = 0. \quad (34)$$

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

$$\boldsymbol{0} \leq \boldsymbol{\alpha} \leq c_3\boldsymbol{s}_-. \quad (35)$$

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

$$([X_+, \boldsymbol{e}_+]^{\mathrm{T}}[X_+, \boldsymbol{e}_+] + c_1E_1)[\boldsymbol{w}_+^{\mathrm{T}}, b_+]^{\mathrm{T}} + [X_-, \boldsymbol{e}_-]^{\mathrm{T}}\boldsymbol{\alpha} = 0 \quad (36)$$

$$i.e., (H_+^{\mathrm{T}}H_+ + c_1E_1)^{-1}\boldsymbol{u}_+ + H_-^{\mathrm{T}}\boldsymbol{\alpha} = 0. \quad (37)$$

Substituting Eq.(37) into the Lagrange function Eq.(29) yields:

$$\begin{aligned} \text{Ł}(\boldsymbol{w}_+, b_+, \boldsymbol{\xi}_-) &= \frac{1}{2}c_1E_1\boldsymbol{u}_+^{\mathrm{T}}\boldsymbol{u}_+ + \frac{1}{2}(H_+\boldsymbol{u}_+)^{\mathrm{T}}(H_+\boldsymbol{u}_+) \\ &\quad + \boldsymbol{\alpha}^{\mathrm{T}}H_-\boldsymbol{u}_+ + \boldsymbol{e}_-^{\mathrm{T}}\boldsymbol{\alpha} \\ &= \boldsymbol{e}_-^{\mathrm{T}}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}H_-(H_+^{\mathrm{T}}H_+ + c_1E_1)^{-1}H_-^{\mathrm{T}}\boldsymbol{\alpha}. \end{aligned} \quad (38)$$

Combine the maximization objective in Eq.(38) and the constraints in Eq.(35), and we eventually obtain the Wolfe dual form of the problem as that in Eq.(26). Similarly, the Wolfe dual form Eq.(27) of the primal problem Eq.(25) can also be proved accordingly. Despite of these dual forms, the relationships between the optimal solutions $\boldsymbol{u}_\pm^*$ of the primal problems and those $\boldsymbol{\alpha}^*$and $\boldsymbol{\beta}^*$ of the dual problems illustrated in Eq.(28) can also be derived from Eq.(37) and the related expressions. □

By solving the dual forms of Eq.(26) and Eq.(27), one can obtain the optimal solutions $\boldsymbol{\alpha}^*$ and $\boldsymbol{\beta}^*$ of the dual problems, and furthermore $\boldsymbol{u}_\pm^*$ of the corresponding primal problems. The non-parallel proximal hyperplanes can thus be subsequently obtained. For a testing instance $\boldsymbol{x} \in \mathcal{R}^n$, the classification decision function can be given as:

$$f(\boldsymbol{x}) = \underset{\pm}{\operatorname{argmin}} \frac{\mid \boldsymbol{w}_\pm^{*\,\mathrm{T}}\boldsymbol{x} + b_\pm^* \mid}{\parallel \boldsymbol{w}_\pm^* \parallel}. \quad (39)$$

3.2.2. Nonlinear FR-TSVM

In nonlinear case, the classification problem is intuitively solved by mapping the input instance $\boldsymbol{x}$ from the input space $\mathcal{R}^n$ to a high-dimensional feature space $\mathcal{H}$ through transformation $\varphi(\boldsymbol{x})$. Using alternative kernel function $\kappa(\cdot,\cdot)$, which implicitly calculates the dot-product of a pair of transformations $\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = \langle\varphi(\boldsymbol{x}_1), \varphi(\boldsymbol{x}_2)\rangle$, similarity manipulation of transformed $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ 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 [41]. With the kernel function, the nonlinear dual proximal hyperplanes of FR-TSVM can be stated as:

$$\kappa(\boldsymbol{x}, X^{\mathrm{T}})\boldsymbol{w}_+ + b_+ = 0 \text{ and } \kappa(\boldsymbol{x}, X^{\mathrm{T}})\boldsymbol{w}_- + b_- = 0\,. \quad (40)$$

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

$$\begin{aligned} \min_{\boldsymbol{w}_+, b_+, \boldsymbol{\xi}_-} \quad & \frac{1}{2}c_1 \parallel \boldsymbol{w}_+ \parallel^2 + \frac{1}{2} \parallel \kappa(X_+, X^{\mathrm{T}})\boldsymbol{w}_+ + \boldsymbol{e}_+b_+ \parallel^2 \\ & + c_3\boldsymbol{s}_-^{\mathrm{T}}\boldsymbol{\xi}_- \\ \text{s.t.} \quad & -(\kappa(X_-, X^{\mathrm{T}})\boldsymbol{w}_+ + \boldsymbol{e}_-b_+) + \boldsymbol{\xi}_- \geq \boldsymbol{e}_-, \boldsymbol{\xi}_- \geq 0, \end{aligned} \quad (41)$$

$$\begin{aligned} \min_{\boldsymbol{w}_-, b_-, \boldsymbol{\xi}_+} \quad & \frac{1}{2}c_2 \parallel \boldsymbol{w}_- \parallel^2 + \frac{1}{2} \parallel \kappa(X_-, X^{\mathrm{T}})\boldsymbol{w}_- + \boldsymbol{e}_-b_- \parallel^2 \\ & + c_4\boldsymbol{s}_+^{\mathrm{T}}\boldsymbol{\xi}_+ \\ \text{s.t.} \quad & (\kappa(X_+, X^{\mathrm{T}})\boldsymbol{w}_- + \boldsymbol{e}_+b_-) + \boldsymbol{\xi}_+ \geq \boldsymbol{e}_+, \boldsymbol{\xi}_+ \geq 0. \end{aligned} \quad (42)$$

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

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{e}_-^{\mathrm{T}}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}S_+(S_-^{\mathrm{T}}S_- + c_1E_1)^{-1}S_+^{\mathrm{T}}\boldsymbol{\alpha} \\ s.t. \quad & \boldsymbol{0} \leq \boldsymbol{\alpha} \leq c_3\boldsymbol{s}_-, \end{aligned} \quad (43)$$

$$\begin{aligned} \max_{\boldsymbol{\beta}} \quad & \boldsymbol{e}_+^{\mathrm{T}}\boldsymbol{\beta} - \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}S_-(S_+^{\mathrm{T}}S_+ + c_2E_2)^{-1}S_-^{\mathrm{T}}\boldsymbol{\beta} \\ s.t. \quad & \boldsymbol{0} \leq \boldsymbol{\beta} \leq c_4\boldsymbol{s}_+, \end{aligned} \quad (44)$$

where $S_+ = [\kappa(X_+, X^{\mathrm{T}}), \boldsymbol{e}_+]$ *and* $S_- = [\kappa(X_-, X^{\mathrm{T}}), \boldsymbol{e}_-]$. *By designating* $\boldsymbol{v}_\pm^* = [\boldsymbol{w}_\pm^{*\mathrm{T}}, b_\pm^*]^{\mathrm{T}}$ *for solutions of the primal problems of Eq.*(41) *and Eq.*(42)*, there are parametric relationships between the optimal* $\boldsymbol{v}_\pm^*$ *and the optimal solutions* $\boldsymbol{\alpha}^*$ *and* $\boldsymbol{\beta}^*$ *of their corresponding dual forms Eq.*(43) *and Eq.*(44)*:*

$$\begin{aligned} \boldsymbol{v}_+^* &= -(S_+^{\mathrm{T}}S_+ + c_1E_1)^{-1}S_-^{\mathrm{T}}\boldsymbol{\alpha}^*, \\ \boldsymbol{v}_-^* &= (S_-^{\mathrm{T}}S_- + c_2E_2)^{-1}S_+^{\mathrm{T}}\boldsymbol{\beta}^*. \end{aligned} \quad (45)$$

Proof. Referring to the proof of Theorem 1 for linear FR-TSVM, the proof of Theorem 2 for nonlinear FR-TSVM can be derived accordingly following the steps in Eq.(29)-(38). □

Once solutions of the dual problems Eq.(43) and Eq.(44) are obtained, solutions of the primal problems Eq.(41) and Eq.(42) can be obtained through Eq.(45), and the decision function for classifying a testing instance $\boldsymbol{x} \in \mathcal{R}^n$ is eventually given by:

$$f(\boldsymbol{x}) = \underset{\pm}{\operatorname{argmin}} \frac{\mid \kappa(\boldsymbol{x}, X^{\mathrm{T}})\boldsymbol{w}_\pm^{*\,\mathrm{T}} + b_\pm^* \mid}{\sqrt{\boldsymbol{w}_\pm^{*\mathrm{T}}\kappa(X, X^{\mathrm{T}})\boldsymbol{w}_\pm^*}}. \quad (46)$$

Algorithm 1 A dual CD method for FR-TSVM

Compute $Q = (H_+^{\mathrm{T}} H_+ + c_1 E_1)^{-1} H_-^{\mathrm{T}}$ and $\overline{Q}_{ii} = H_{-i} Q_i$
Initial $\boldsymbol{\alpha} \leftarrow \mathbf{0}$ and $\boldsymbol{u}_+ \leftarrow \mathbf{0}$
while $\boldsymbol{\alpha}$ is not optimized **do**
  for $i = 1, 2, \ldots, l_-$ **do**
    $\nabla_i f(\boldsymbol{\alpha}) = -H_{-i}\boldsymbol{u}_+ - 1$
    Compute$\nabla_i^p f(\boldsymbol{\alpha})$ by Eq.(54)
    if $\nabla_i^p f(\boldsymbol{\alpha}) \neq 0$ **then**
      $\overline{\alpha}_i \leftarrow \alpha_i$
      $\alpha_i \leftarrow \min(\max(\alpha_i - \nabla_i^p f(\boldsymbol{\alpha})/\overline{Q}_{ii}, 0), c_3 s_{i-})$
      $u_{+i} \leftarrow u_{+i} - Q_i(\alpha_i - \overline{\alpha}_i)$
    end if
  end for
end while

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 [39] can be further employed for solving the FR-TSVM. As compared with the paired objective functions in Eq.(26)-(27) and Eq.(43)-(44), pairwise similarities, *i.e.*, similarities between Eq.(26) and Eq.(43), and between Eq.(27)and Eq.(44), can be easily found. 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 [39, 40, 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 $Q = (H_+^{\mathrm{T}} H_+ + c_1 E_1)^{-1} H_-^{\mathrm{T}}$ and $\overline{Q} = H_- Q$, the problem Eq.(26) can be first abbreviated as a quadratic expression:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & f(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}\overline{Q}\boldsymbol{\alpha} - \boldsymbol{e}_-^{\mathrm{T}}\boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq c_3 \boldsymbol{s}_-. \end{aligned} \tag{47}$$

As an iterative scheme, the FR-TSVM generates subsequently a sequence of updating vectors $\{\boldsymbol{\alpha}^0, \ldots, \boldsymbol{\alpha}^k, \boldsymbol{\alpha}^{k+1}\}$ to consecutively optimize the objective function where $\boldsymbol{\alpha}^k = [\boldsymbol{\alpha}_1^k, \boldsymbol{\alpha}_2^k, \ldots, \boldsymbol{\alpha}_{l_-}^k]^{\mathrm{T}} \in \mathcal{R}^{l_-}$. There are two levels of iterations. An integer k first is used to index the 2nd-level of outer iterations updated from $\boldsymbol{\alpha}^k$ to $\boldsymbol{\alpha}^{k+1}$. In every k-th outer iteration, the update of $\boldsymbol{\alpha}^k$ is further subdivided into 1st-level of l_- inner iterations, indexed by i, to generate a series of intermediate vectors

$$\{\boldsymbol{\alpha}^{k,1}, \boldsymbol{\alpha}^{k,2}, \ldots, \boldsymbol{\alpha}^{k,i}, \ldots, \boldsymbol{\alpha}^{k,l_-}, \boldsymbol{\alpha}^{k,l_-+1}\}. \tag{48}$$

The two-level updated vector $\boldsymbol{\alpha}^{k,i}$ is thus expressed as:

$$\begin{aligned} \boldsymbol{\alpha}^{k,i} = [\alpha_1^{k+1}, \ldots, \alpha_{i-1}^{k+1}, \alpha_i^k, \ldots, \alpha_{l_-}^k]^{\mathrm{T}}, \\ \forall i = 1, 2, \ldots, l_- \end{aligned} \tag{49}$$

and

$$\boldsymbol{\alpha}^{k,1} = \boldsymbol{\alpha}^k \; \textit{and} \; \boldsymbol{\alpha}^{k,l_-+1} = \boldsymbol{\alpha}^{k+1}. \tag{50}$$

To update the intermediate $\boldsymbol{\alpha}^{k,i}$ to $\boldsymbol{\alpha}^{k,i+1}$, the following single variable sub-problem should be solved:

$$\min_d f(\boldsymbol{\alpha}^{k,i} + d\boldsymbol{e}_i) \quad \text{s.t.} \quad 0 \leq \alpha_i^k + d \leq c_3 s_{i-}, \tag{51}$$

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

$$f(\boldsymbol{\alpha}^{k,i} + d\boldsymbol{e}_i) = \frac{1}{2}\overline{Q}_{ii} d^2 + \nabla_i f(\boldsymbol{\alpha}^{k,i}) d + c, \tag{52}$$

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

$$\nabla_i^p f(\boldsymbol{\alpha}^{k,i}) = 0, \tag{53}$$

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

$$\nabla_i^p f(\boldsymbol{\alpha}^{k,i}) = \begin{cases} \min(0, \nabla_i f(\boldsymbol{\alpha})), & \text{if } \alpha_i = 0 \\ \nabla_i f(\boldsymbol{\alpha}), & \text{if } 0 < \alpha_i < c_3 s_{i-} \\ \max(0, \nabla_i f(\boldsymbol{\alpha})), & \text{if } \alpha_i = c_3 s_{i-} \end{cases} \tag{54}$$

The key for computational reduction is that we can directly move forward to next $i+1$ iteration without updating $\alpha_i^{k,i}$ in the l_-length inner-iteration updates if Eq.(53) has been fulfilled. In other words, we only update $\alpha_i^{k,i}$ to temporally meet the optimal solution of Eq.(51). By introducing Lipschitz continuity [43], the optimum of Eq.(52) can be reached by:

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

However $\alpha_i^{k,i+1}$ 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 $Q = (H_+^{\mathrm{T}} H_+ + c_1 E_1)^{-1} H_-^{\mathrm{T}}$, and preserved through all the iterations, and $\nabla_i f(\boldsymbol{\alpha}^{k,i})$ can be obtained by

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

Here, the computation of $\nabla_i f(\boldsymbol{\alpha}^{k,i})$ by Eq.(56), which is approximated as $O(l_-\overline{n})$ where $\overline{n}$ is the average count of non-zero elements in $\overline{Q}$ per instance, is expensive. In order to reduce the computation, $\nabla_i f(\boldsymbol{\alpha}^{k,i})$ can alternatively be calculated by [40]:

$$\nabla_i f(\boldsymbol{\alpha}) = -H_{-i}\boldsymbol{u}_+ - 1, \tag{57}$$

with a pre-defined $\boldsymbol{u}_+$

$$\boldsymbol{u}_+ = -Q\boldsymbol{\alpha}, \tag{58}$$

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

$$u_{+i} \leftarrow u_{+i} - Q_i(\alpha_i - \overline{\alpha}_i). \tag{59}$$

Here, the time consumption by the iterative maintaining of $\boldsymbol{u}_+$ requires only $O(\overline{n})$ rather than that by the direct calculation by Eq.(58), where $\overline{\alpha}_i$ and α_i denote the values of the primal optimizer before and after the corresponded update iteration, respectively. With a zero initial value for the first $\boldsymbol{u}_+$ due to the generally adopted $\boldsymbol{\alpha}^0 = \mathbf{0}$, the optimal solution of $\boldsymbol{u}_+$ 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 α^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 speeds-up more 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 [5]. 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 [5], 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 $\overline{A} = \{1, \cdots, l_-\} \backslash A$. The use of the active set is for dynamically collecting those non-bounded 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:

$$\begin{aligned} \min_{\boldsymbol{\alpha}_A} \quad & \frac{1}{2}\boldsymbol{\alpha}_A^{\mathrm{T}} \overline{Q}_{AA} \boldsymbol{\alpha}_A + (\overline{Q}_{A\overline{A}} \boldsymbol{\alpha}_{\overline{A}} - \boldsymbol{e}_A)^{\mathrm{T}} \boldsymbol{\alpha}_A \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha}_A \leq c_3 \boldsymbol{s}_{A-}, \end{aligned} \tag{60}$$

where $\overline{Q}_{AA}$ and $\overline{Q}_{A\overline{A}}$ are sub-matrices of $\overline{Q}$, and $\boldsymbol{\alpha}_A$ and $\boldsymbol{\alpha}_{\overline{A}}$ are Lagrange multiplier sub-vectors corresponding to subsets A and $\overline{A}$, respectively. As illustrated previously, $\boldsymbol{\alpha}_{\overline{A}}$ 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 $\nabla f(\boldsymbol{\alpha})$ is a key to update the optimizer. Following the subdivision strategy, $\nabla f(\boldsymbol{\alpha})$ in Eq.(56) can also be decomposed as:

$$\nabla_i f(\boldsymbol{\alpha}) = \overline{Q}_{i,A} \boldsymbol{\alpha}_A + \overline{Q}_{i,\overline{A}} \boldsymbol{\alpha}_{\overline{A}} - 1 \tag{61}$$

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 $\boldsymbol{u}_+$ 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 an ongoing optimization whether the optimization is constrained or not. The rule is still true for a *coordinate descent* optimization. In FR-TSVM, what we have are cyclical gradients $\nabla_i f(\boldsymbol{\alpha}^{k,i})$ for $i = 1, 2, \ldots, l_-$. With the subsequent cyclical $\nabla_i f(\boldsymbol{\alpha}^{k,i})$, Theorem 3 is established for bound and shrinking of the active set according to the original theorem proposed by Hsieh *et al.*, [40].

Theorem 3. *By taking the list $\{\alpha^{k,i}\}$ in the solution space, let $*$ be the final convergent point. The FR-TSVM sustains the following properties:*
a). If $\alpha_i^* = 0$, and $\nabla_i f(\boldsymbol{\alpha}^*) > 0$, there is a $\exists k_i$ such that

$$\forall k \geq k_i, \forall r, \alpha_i^{k,r} = 0.$$

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

$$\forall k \geq k_i, \forall r, \alpha_i^{k,r} = c_3 s_{i-}.$$

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

Proof. Referring to the proof in Appendix 7.3 in [40], imitations are taken to obtain Theorem 3 . □

Based on measures $\max_i \nabla_i^p f(\boldsymbol{\alpha}^k) > 0, \min_i \nabla_i^p f(\boldsymbol{\alpha}^k) < 0$ which are used to evaluate the solution violation level of a certain outer iteration k, a pair of bounds, $M^{k-1} = \max_i \nabla_i^p f(\boldsymbol{\alpha}^{k-1,i})$ and $m^{k-1} = \min_i \nabla_i^p f(\boldsymbol{\alpha}^{k-1,i})$, are asserted for bounding $\nabla_i f(\boldsymbol{\alpha}^{k,i})$ 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 $\overline{A}$ at the inner iteration for updating component

α_i from $\alpha_i^{k,i}$ to $\alpha_i^{k,i+1}$ while next two conditions are hold:

$$\alpha_i^{k,i} = 0 \text{ and } \nabla_i f(\boldsymbol{\alpha}^{k,i}) > \overline{M}^{k-1}, \tag{62}$$

$$\alpha_i^{k,i} = c_3 s_{i-} \text{ and } \nabla_i f(\boldsymbol{\alpha}^{k,i}) < \overline{m}^{k-1}, \tag{63}$$

where

$$\overline{M}^{k-1} = \begin{cases} M^{k-1}, & \text{if } M^{k-1} > 0 \\ \infty, & \text{if } M^{k-1} \leq 0 \end{cases} \tag{64}$$

and

$$\overline{m}^{k-1} = \begin{cases} m^{k-1}, & \text{if } m^{k-1} > 0 \\ -\infty, & \text{if } m^{k-1} \leq 0 \end{cases}. \tag{65}$$

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 $\overline{M}$ and $\overline{m}$, respectively, to exclude inactive members in next outer iteration. According to property (c) of Theorem 3, bounds $\overline{M}$ and $\overline{m}$ would become closer after iterations, and would theoretically meet with each other finally:

$$\overline{M}^k = \overline{m}^k \text{ if } k \to \infty. \tag{66}$$

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

$$\overline{M}^k - \overline{m}^k < \varepsilon \tag{67}$$

for termination after k finite iterations is much more practical. While the gapped termination condition is reached, the optimal solution $\boldsymbol{\alpha}^k$ of the sub-problem Eq.(60) is also possessed. Actually, this optimal $\boldsymbol{\alpha}^k$ is only optimized for the sub-problem Eq.(60), not for the full problem Eq.(26). Hence, we ignore the current active set and set it to the full set $\{1, \ldots, l_-\}$ to get back all the α_i from the cached historical $\alpha_i^{k,i}$ in the final pass at the end of the procedure to ensure the recomposed $\boldsymbol{\alpha}^*$ to fulfill Eq.(60).

The heuristic shrinking might raise a failure risk with the mismatch $\overline{M}$ and $\overline{m}$ even with a tolerable gapped mismatch, for example, $\overline{M} \leq 0$, or $\overline{m} \geq 0$. If such a condition happens, the whole FR-TSVM of Eq.(60) should be re-optimized with a different set of initial guests of $\boldsymbol{\alpha}$. Additionally, the shrinking is in general applied heuristically in a fixed sequence of the l_--dimensional gradients. A random update scheme performed a more rapid convergent rate than a sequential update scheme [39].

According to the separation of active from inactive set, u_+ defined in Eq.(58) can be re-expressed as:

$$\boldsymbol{u}_+ = -(Q_A \boldsymbol{\alpha}_A + Q_{\overline{A}} \boldsymbol{\alpha}_{\overline{A}}), \tag{68}$$

which means that some elements coincided in the $\alpha_{i\overline{A}}$ and $\overline{\alpha}_{i\overline{A}}$ would remain the same before and after the update iteration i, and can be prevented in the update of $u_{+i} \leftarrow u_{+i} - Q_i(\alpha_i - \overline{\alpha_i})$. Algorithm 2 describes the update procedure of Algorithm 1 with active set shrinking in a random scheme.

Algorithm 2 The optimization of FR-TSVM

Compute $Q = (H_+^{\mathrm{T}} H_+ + c_1 E)^{-1} H_-^{\mathrm{T}}$ and $\overline{Q}_{ii} = H_{-i} Q_i$
Let $A \leftarrow \{1, \ldots, l_-\}$
Given ϵ and initialized $\boldsymbol{\alpha} \leftarrow \mathbf{0}$, $\boldsymbol{u}_+ \leftarrow \mathbf{0}$
Let $\overline{M} \leftarrow \infty$ and $\overline{m} \leftarrow -\infty$
while **do**
  Let $M \leftarrow -\infty$,$m \leftarrow \infty$
  for all $i \in A$ (a randomly and exclusively selected) **do**
    $\nabla_i f(\boldsymbol{\alpha}) = -H_{-i} \boldsymbol{u}_+ - 1$
    $\nabla_i^p f(\boldsymbol{\alpha}) \leftarrow 0$
    if $\alpha_i = 0$ **then**
      if $\nabla_i^p f(\boldsymbol{\alpha}) > \overline{M}$, **then** $A = A \backslash \{i\}$ **end if**
      if $\nabla_i^p f(\boldsymbol{\alpha}) < 0$, **then** $\nabla_i^p f(\boldsymbol{\alpha}) \leftarrow \nabla_i f(\boldsymbol{\alpha})$ **end if**
    else if $\alpha_i = c_3 s_{i-}$ **then**
      if $\nabla_i^p f(\boldsymbol{\alpha}) < \overline{m}$, **then** $A = A \backslash \{i\}$ **end if**
      if $\nabla_i^p f(\boldsymbol{\alpha}) > 0$, **then** $\nabla_i^p f(\boldsymbol{\alpha}) \leftarrow \nabla_i f(\boldsymbol{\alpha})$ **end if**
    else
      $\nabla_i^p f(\boldsymbol{\alpha}) \leftarrow \nabla_i f(\boldsymbol{\alpha})$
    end if
    $M \leftarrow \max(M, \nabla_i^p f(\boldsymbol{\alpha}))$
    $m \leftarrow \min(m, \nabla_i^p f(\boldsymbol{\alpha}))$
    if $\nabla_i^p f(\boldsymbol{\alpha}) \neq 0$ **then**
      $\overline{\alpha}_i \leftarrow \alpha_i$
      $\alpha_i \leftarrow \min(\max(\alpha_i - \nabla_i f(\boldsymbol{\alpha})/\overline{Q}_{ii}, 0), c_3 s_{i-})$
      $u_{+i} \leftarrow u_{+i} - Q_i(\alpha_i - \overline{\alpha}_i)$
    end if
  end for
  if $M - m < \epsilon$ **then**
    if $A = \{1, \ldots, l_-\}$, **break**
  else
    $A \leftarrow \{1, \ldots, l_-\}, \overline{M} \leftarrow \infty, \overline{m} \leftarrow -\infty$
  end if
  if $M \leq 0$, **then** $\overline{M} \leftarrow \infty$. **else** $\overline{M} \leftarrow M$ **end if**
  if $M \geq 0$, **then** $\overline{m} \leftarrow -\infty$. **else** $\overline{m} \leftarrow m$ **end if**
end while

4. Experiments

To show the learning efficiency and generalization ability of the FR-TSVM, some experiments are implemented on the 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. Under identical hardware environment and software environment, 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 adopted for optimization. The proposed FR-TSVM is optimized by the coordinate decent of Algorithm 2. The Matlab code of all experiments are released at `https://github.com/gaobb/FR-TSVM`.

Due to the employment or unemployment of the transformation function $\varphi(\cdot)$, 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 $\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2/g^2)$ 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 first implement experiments on two 2-dimensional artificial datasets and compare the proposed method with the standard SVM, FSVM and TSVM.

The first dataset is the artificial-generated *Ripleys synathetic* dataset [45]. The *Ripleys synathetic*, often adopted for evaluating a classifier's performance, is a dataset with $n = 2$ comprising 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. 3, respectively. As shown in Fig. 3, compared to those instances locating near the class center, the fuzzy membership values of the instances which are far from the center of class are always smaller.

Table 4 reports the classification performance, including the tesing accuracies and learning CPU time, for the comparative methods with or without the Gaussian kernel. We can see that the linear SVM and the nonlinear FR-TSVM achieve the highest prediction accuracy under the linear and nonlinear case, respectively. The reason for the outperformance of the linear SVM is more likely from the dataset itself than from the classifier because there may be no noise existing on the *Ripleys synathetic* dataset. The noiseless fact of the test 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 this table. In terms of execution time, FR-TSVM shows its excellence in computational efficiency for both linear and nonlinear learning, especially under the linear case. The excellence manifests the remarkable potential of employing FR-TSVM for a fast classification.

Panels in Fig. 4 show the linear and nonlinear separating hyperplanes produced by the comparative methods with equivalent settings. In the panels, while the standard SVM and FSVM produce only a single hyperplane (Fig. 4a and 4b), TSVM and FR-TSVM produce a paired proximal hyperplanes (Fig. 4c and 4d) for class separation. Instead of the decision boundary identical to the single hyperplane in the standard SVM and FSVM, the bisector of the proximal hyperplanes can be used for a more accurate discrimination in TSVM and FR-TSVM. Comparing more of the FR-TSVM and TSVM, the positions of the proximal hyperplanes of FR-TSVM is relatively exact than ones of TSVM. The fact reveals that FR-TSVM is more capable of producing an unbiased accuracy than TSVM.

Table 1: Classification performance comparison on artificial Ripleys dataset.

Methods	Description	SVM	FSVM	TSVM	FR-TSVM
Linear	Acc(%)↑	**89.70**	88.80	89.20	89.10
	Time(s)↓	1.46	2.00	0.28	**0.21**
Nonlinear	Acc(%)↑	90.40	91.10	90.50	**91.30**
	Time(s)↓	1.56	1.79	0.60	**0.24**

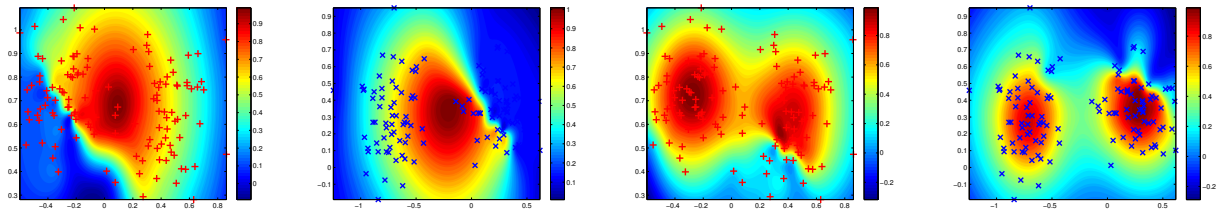
Table 2: Classification performance comparison on the second artificial dataset.

Methods	Description	SVM	FSVM	TSVM	FR-TSVM
nonoise	Acc(%)↑	99.92	99.88	99.92	**100.0**
	Time(s)↓	16.77	17.11	5.03	**0.30**
noise	Acc(%)↑	99.37	**99.63**	97.00	99.33
	Time(s)↓	16.43	16.74	4.93	**0.72**

The second dataset is also a 2-dimensional artificial-generated dataset. In the 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 $\boldsymbol{x} = [x_1, x_2]$. 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.

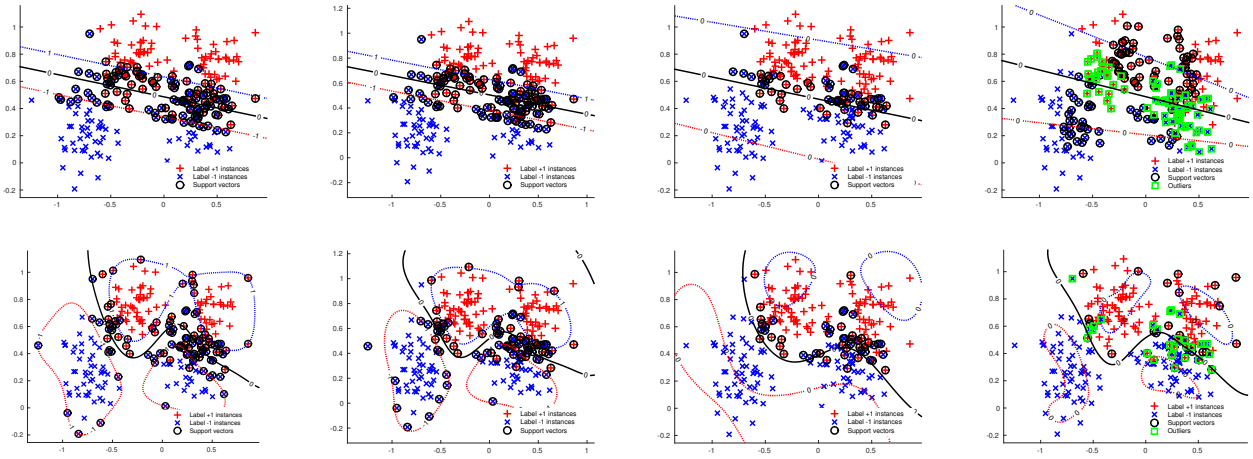
In Fig. 5, the first row shows the results of the nonlinear SVM, FSVM, TSVM and the proposed FR-TSVM on this dataset. We can see that the separating hyperplanes of these classifiers obtain the similar results. To further validate the robust performance of the FR-TSVM, we add a noise $\varepsilon \in \mathcal{N}(0, 0.2)$ 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 result 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 the prediction results including classification accuracy and training time on no-noise and noise cases. We can see that all the methods obtain similarity test accuracies on no-noise training instances, 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



(a) "+1" instances (b) "-1" instances (c) "+1" instances (d) "-1" instances

Figure 3: Fuzzy membership distribution of training instances on Ripley dataset. (a) and (b) for linear case, (c) and (d) for nonlinear case.



(a) SVM (b) FSVM (c) TSVM (d) FR-TSVM

Figure 4: Results of linear and nonlinear SVM, FSVM, TSVM and FR-TSVM on the first artificial dataset (*Ripleys synthetic*). The first and second row show the results used linear kernel and Gaussian kernel, respectively.

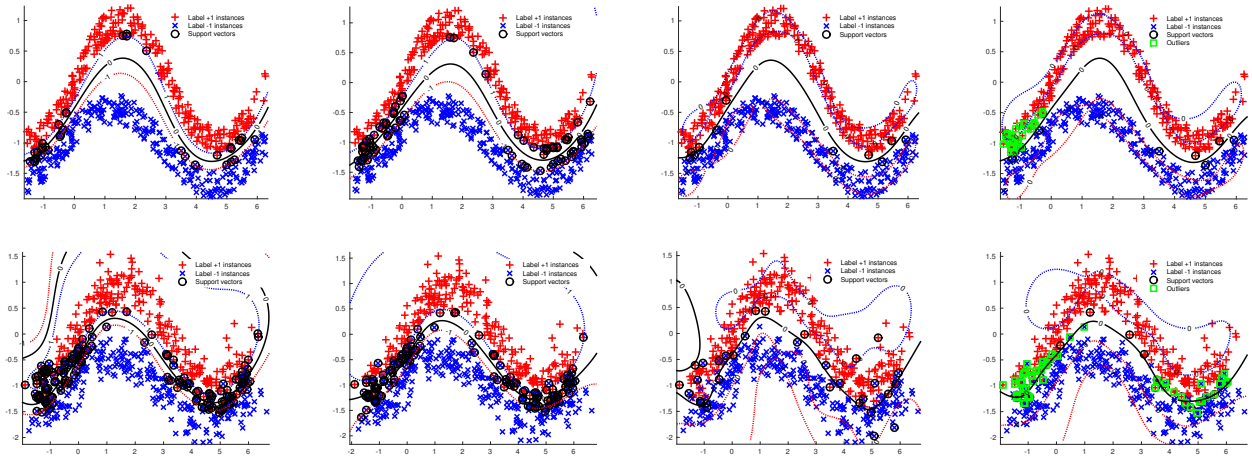
in training model with noise instances. It is noteworthy that TSVM's test accuracy is reduced by almost 3% (from 99.88% to 97.00%), whereas FR-TSVM has only a slight slowdown (less than 1%). It suggests that FR-TSVM is more robust than TSVM when training instances consist of noise data. As for the training 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 [46]. 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 4 and 5 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 4 and 5. 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 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 reason is that these datasets may not have outliers. Fig. 6



(a) SVM (b) FSVM (c) TSVM (d) FR-TSVM

Figure 5: Results of nonlinear SVM, FSVM, TSVM and FR-TSVM on the second artificial dataset. The first row and second show the results on the dataset without noises and with noises, respectively.

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	Data statistics			
	$\#l$	$\#l_+$	$\#l_-$	$\#n$
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	83.53±6.48	85.75±4.06	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	98.33±2.68	98.89±2.34
Hepatitis	79.26±8.76	84.94±7.29	75.58±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**	75.65±4.22	76.95±3.37	75.13±3.78

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 that 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 that whether linear or nonlinear, 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.

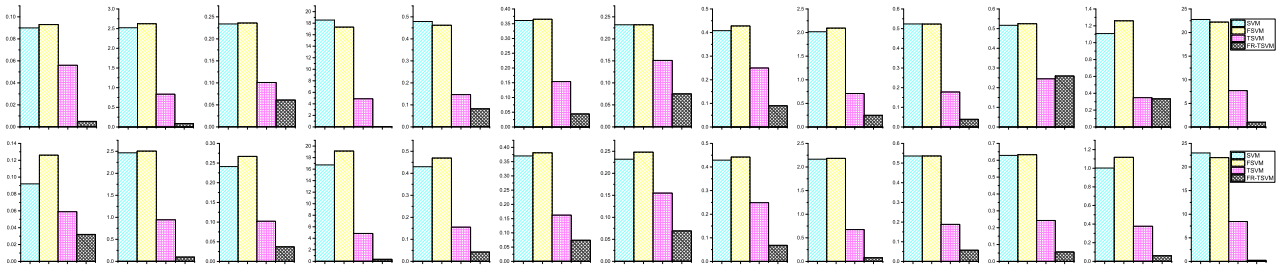
4.3. Parameter analysis

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 synathetic* dataset for example.

Fig. 7 further shows the test accuracy with varying the adjustable parameters on artificial dataset *Ripleys synathetic*. 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.

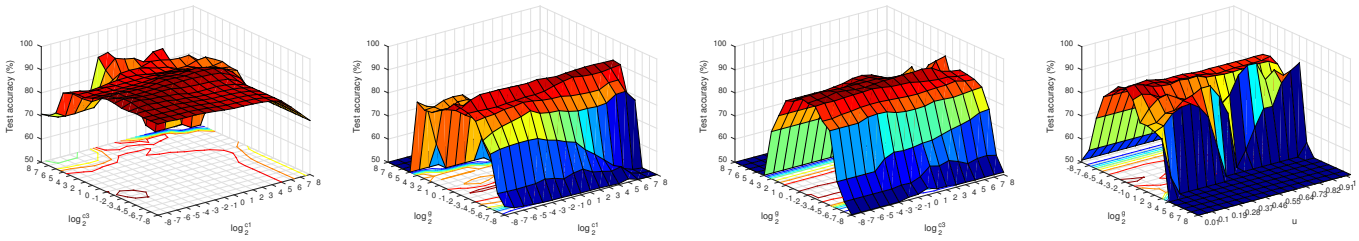
5. Conclusion

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



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m)

Figure 6: Comparison of **training time** on thirteen benchmark datasets including (a)Breast (b)Ionosphere (c)Iris (d)Australian (e)WDBC (f)Wine (g)Hepatitis (h)WPBC (i)Bupa (j)Sonar (k)Glass (l)Heart and (m)Pima.



(a) $\mu = 0.1$, $g = 1$ (b) $\mu = 0.1$, $c_1 = 32$ (c) $\mu = 0.1$, $c_3 = 0.5$ (d) $c_1 = 32$, $c_3 = 0.5$

Figure 7: Accuracy varying adjustable parameters of nonlinear FR-TSVM on Ripleys dataset.

Table 5: Comparison results (mean±std) of nonlinear 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	97.26±4.43	97.17±4.58	96.33±4.77	**98.09±4.03**
Ionosphere	94.84±4.01	94.59±4.31	92.61±6.12	**95.41±4.93**
Iris	100.0±0.00	100.0±0.00	100.0±0.00	100.0±0.00
Australian	85.50±4.53	85.50±4.59	85.50±4.59	**86.81±4.84**
WDBC	94.84±4.23	95.34±3.80	95.34±3.80	**96.39±3.52**
Wine	99.44±1.76	98.89±2.34	100.0±0.00	100.0±0.00
Hepatitis	80.51±8.32	84.28±7.42	82.00±6.84	**84.48±6.86**
WPBC	81.51±7.13	77.88±9.43	75.30±7.93	**82.51±8.05**
Bupa	70.68±8.28	**72.71±7.93**	71.86±5.71	71.84±5.67
Sonar	89.42±5.41	88.92±6.95	89.42±5.41	**89.44±5.31**
Glass	**97.68±3.95**	96.77±4.38	94.87±5.08	97.21±3.93
Heart	84.07±5.25	82.59±4.29	80.74±7.16	**84.81±4.08**
Pima	75.65±3.80	75.26±2.91	**77.34±5.16**	76.17±2.68

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. N. Vapnik, 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.-B Gao, J.-J. Wang, Y. Wang, C.-Y. Yang, Coordinate descent fuzzy twin support vector machine for classification, In IEEE 14th International Conference on Machine Learning and Applications (ICMLA), (2015), pp. 7–12.

[4] 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.

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

[6] 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.

[7] 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.

[8] H.-W. Wang, C. Qi, Y.-C. Wei, B. Li, S. Zhu, Review on data-based decision making methodologies, Acta Autom. Sinica 35 (6) (2009) 820–833.
[9] 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.
[10] R. Khemchandani, S. Chandra, et al., Twin support vector machines for pattern classification, IEEE Trans. on Pattern Analysis and Machine Intelligence 29 (5) (2007) 905–910.
[11] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, N.-Y. Deng, Improvements on twin support vector machines, IEEE Trans. on Neural Networks 22 (6) (2011) 962–968.
[12] Z. Qi, Y. Tian, Y. Shi, Structural twin support vector machine for classification, Knowledge-Based Systems 43 (2013) 74–81.
[13] Z.-Q. Qi, Y.-J. Tian, Y. Shi, Robust twin support vector machine for pattern classification, Pattern Recognition 46 (1) (2013) 305–316.
[14] W.-J. Chen, Y.-H. Shao, N. Hong, Laplacian smooth twin support vector machine for semi-supervised classification, Int'l Journal of Machine Learning and Cybernetics 5 (3) (2014) 459–468.
[15] M. A. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Systems with Applications 36 (4) (2009) 7535–7543.
[16] J. A. Nasiri, N. M. Charkari, S. Jalili, Least squares twin multi-class classification support vector machine, Pattern Recognition 48 (3) (2015) 984–992.
[17] X. Peng, TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition, Pattern Recognition 44 (10) (2011) 2678–2692.
[18] X. Peng, L. Kong, D. Chen, Improvements on twin parametric-margin support vector machine, Neurocomputing 151 (2015) 857–863.
[19] W.-J. Chen, Y.-H. Shao, C.-N. Li, N.-Y. Deng, MLTSVM: A novel twin support vector machine to multi-label learning, Pattern Recognition 52 (2016) 61–74.
[20] R. Rastogi, S. Sharma, S. Chandra, Robust parametric twin support vector machine for pattern classification, Neural Processing Letters (2017) 1–31.
[21] Y. Xu, Z. Yang, X. Pan, A novel twin support-vector machine with pinball loss, IEEE transactions on neural networks and learning systems 28 (2) (2017) 359–370.
[22] X. Peng, TSVR: an efficient twin support vector machine for regression, Neural Networks 23 (3) (2010) 365–372.
[23] X. Peng, D. Xu, J. Shen, A twin projection support vector machine for data regression, Neurocomputing 138 (2014) 131–141.
[24] Y.-F. Ye, L. Bai, X.-Y. Hua, Y.-H. Shao, Z. Wang, N.-Y. Deng, Weighted lagrange ε-twin support vector regression, Neurocomputing 197 (2016) 53–68.
[25] R. Khemchandani, K. Goyal, S. Chandra, Twsvr: regression via twin support vector machine, Neural Networks 74 (2016) 14–21.
[26] R. Rastogi, P. Anand, S. Chandra, A ν-twin support vector machine based regression with automatic accuracy control, Applied Intelligence 46 (3) (2017) 670–683.
[27] C.-F. Lin, S.-D. Wang, Fuzzy support vector machines, IEEE Trans. on Neural Networks 13 (2) (2002) 464–471.
[28] C.-F. Lin, et al., Training algorithms for fuzzy support vector machines with noisy data, Pattern recognition letters 25 (14) (2004) 1647–1656.
[29] Y. Wu, Y. Liu, Robust truncated hinge loss support vector machines, Journal of the American Statistical Association.
[30] T. Inoue, S. Abe, Fuzzy support vector machines for pattern classification, in: Proc. Int'l Joint Conf. on Neural Networks, IEEE, 2001, pp. 1449–1454.
[31] C.-Y. Yang, J.-J. Chou, F.-L. Lian, Robust classifier learning with fuzzy class labels for large-margin support vector machines, Neurocomputing 99 (2013) 1–14.
[32] W. M. Tang, Fuzzy svm with a new fuzzy membership function to solve the two-class problems, Neural Processing Letters 34 (3) (2011) 209–219.
[33] Y. Xu, L. Wang, P. Zhong, A rough margin-based ν-twin support vector machine, Neural Computing and Applications 21 (6) (2012) 1307–1317.
[34] R. Khemchandani, C. S. Jayadeva, Fuzzy twin support vector machines for pattern classification, Mathematical Programming and Game Theory for Decision Making. (2008) 131–42.
[35] T. Kudo, Y. Matsumoto, Chunking with support vector machines, in: Proc. North American Chapter of the Association for Computational Linguistics on Language technologies, Association for Computational Linguistics, 2001, pp. 1–8.
[36] J.-X. Dong, A. Krzyżak, C. Y. Suen, Fast svm training algorithm with decomposition on very large data sets, IEEE Trans. on Pattern Analysis and Machine Intelligence 27 (4) (2005) 603–618.
[37] J. C. Platt, Fast training of support vector machines using sequential minimal optimization, in: Advances in kernel methods—Support Vector Learning, MIT Press, 1999, pp. 185–208.
[38] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. on Intelligent Systems and Technology 2 (3) (2011) 27, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
[39] K.-W. Chang, C.-J. Hsieh, C.-J. Lin, Coordinate descent method for large-scale l2-loss linear support vector machines, The Journal of Machine Learning Research 9 (2008) 1369–1398.
[40] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, S. Sundararajan, A dual coordinate descent method for large-scale linear svm, in: Proc. of the 25th int'l conf. on Machine learning, ACM, 2008, pp. 408–415.
[41] A. J. Smola, B. Schölkopf, Learning with kernels, Citeseer, 1998.
[42] Y.-H. Shao, N.-Y. Deng, A coordinate descent margin based-twin support vector machine for classification, Neural Networks 25 (2012) 114–121.
[43] M. O'Searcoid, Metric spaces, Springer Science & Business Media, 2006.
[44] A. Messac, Optimization in Practice with MATLAB®: For Engineering Students and Professionals, Cambridge University Press, 2015.
[45] B. D. Ripley, Pattern recognition and neural networks, Cambridge university press, 1996.
[46] C. Blake, C. J. Merz, UCI repository of machine learning databases, available at `http://~archive.ics.uci.edu/ml/about.html` (1998).