Pattern Recognition 121 (2022) 108195

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Indefinite twin support vector machine with DC functions programming

Yuexuan An[a,b], Hui Xue[a,b,*]

[a] School of Computer Science and Engineering, Southeast University, Nanjing, 210096, China
[b] MOE Key Laboratory of Computer Science and Information Integration(Southeast University), China

ARTICLE INFO

Article history:
Received 17 October 2019
Revised 12 June 2021
Accepted 20 July 2021
Available online 21 July 2021

Keywords:
SVM
TWSVM
Indefinite kernel
DC Programming
Structural risk minimization principle

ABSTRACT

Twin support vector machine (TWSVM) is an efficient algorithm for binary classification. However, the lack of the structural risk minimization principle restrains the generalization of TWSVM and the guarantee of convex optimization constraints TWSVM to only use positive semi-definite kernels (PSD). In this paper, we propose a novel TWSVM for indefinite kernel called indefinite twin support vector machine with difference of convex functions programming (ITWSVM-DC). The indefinite TWSVM (ITWSVM) leverages a maximum margin regularization term to improve the generalization of TWSVM and a smooth quadratic hinge loss function to make the model continuously differentiable. The representer theorem is applied to the ITWSVM and the convexity of the ITWSVM is analyzed. In order to address the non-convex optimization problem when the kernel is indefinite, a difference of convex functions (DC) is used to decompose the non-convex objective function into the subtraction of two convex functions and a line search method is applied in the DC algorithm to accelerate the convergence rate. A theoretical analysis illustrates that ITWSVM-DC can converge to a local optimum and extensive experiments on indefinite and positive semi-definite kernels show the superiority of ITWSVM-DC.

1. Introduction

Support vector machine (SVM) [1–4] is a machine learning method based on the theory of statistical learning and the principle of structural risk minimization (reducing the VC dimension of learning machine and seeking the minimum sum of experience risk and confidence risk). The learning strategy of SVM is “maximum margin”, that is, solving the optimal separating hyperplane with the maximal margin, which gives impetus to have good generalization. In fact, SVM aims to address a constrained quadratic programming (QP) problem. By introducing kernel learning, the samples in low dimension feature space can be implicitly mapped into the high dimensional feature space and the complexity of inner product operations in SVM can be avoided [5]. Therefore, it overcomes the problems of the “curse of dimensionality” and “over-fitting” to a great extent. Since SVM was proposed, it has attracted extensive attention for its superior performance [6–8] and has been widely used in anomaly detection [9], image retrieval [10], sequence-based prediction of protein [11], etc.

Jayadeva et al. proposed a twin support vector machine (TWSVM) as a useful extension of the traditional SVM. TWSVM generates two nonparallel hyperplanes by solving a pair of smaller-sized QP problems instead of a single larger-sized QP problem [12]. Therefore, compared with SVM, TWSVM accelerates the learning speed for the smaller-sized model and is more resilient to “Cross Planes” datasets for the solution of two nonparallel hyperplanes. However, TWSVM only takes into account the empirical risk minimization principle and lacks structural risk minimization principle which is a significant advantage of SVM. Some scholars solve the problem by modifying the loss function to ensure the structural risk minimization principle and improve the generalization performance [13,14]. However, in order to ensure the convexity of the modified TWSVM to reduce the dual gap and satisfy Mercer’s condition, the kernel in TWSVM is limited to positive semi-definite (PSD) kernels. In fact, verifying the property of PSD for a given kernel can be a challenging task beyond the ability of most scholars. Moreover, indefinite kernels (i.e. kernel matrix contains a mix of positive and negative eigenvalues) play an important role in machine learning and real-world applications [15]. Some functions such as hyperbolic tangent kernel are indefinite [16] and most kernels as similarity measures directly utilized in real-world applications are indefinite [17]. Unfortunately, to the best of our knowl-

* Corresponding author at: School of Computer Science and Engineering, Southeast University, Nanjing, 210096, China.
E-mail address: hxue@seu.edu.cn (H. Xue).

https://doi.org/10.1016/j.patcog.2021.108195
0031-3203/

edge, TWSVM has not exploited the study of indefinite kernels and cannot elegantly deal with indefinite kernels.

However, indefinite kernel SVM (IKSVM) has been studied extensively and many algorithms have been proposed for dealing with indefinite kernels in SVMs. One direction is "Kernel transformation" which applies direct spectral transformations to indefinite kernels. These methods are represented by "Clip" (set all negative eigenvalues to zero) [18], "Flip" (set negative eigenvalues to their absolute value) [19] and "Shift" (add all eigenvalues with a positive constant to make sure all eigenvalues are non-negative after shifting) [20]. The other direction is "Reformulate problems" which is solving the non-convex problem directly. However, these methods may lose useful information in samples and have adverse effects on modeling a function [21,22]. In 2017, Xu et al. [23] directly focus on the non-convex primal form of IKSVM by decomposing the primal problem into two convex functions.

In this paper, we construct a bridge between TWSVM and indefinite kernel and propose a novel algorithm called indefinite twin support vector machine with difference of convex functions programming (ITWSVM-DC). In order to consider the confidence interval which is ignored by TWSVM and be free from complex matric inversion, we add a regularized item into TWSVM. We further introduce the smooth quadratic hinge loss function to make the regularized TWSVM (ITWSVM) model continuously differentiable and more resilient to indefinite kernels. Then, we analyze the convexity of the proposed ITWSVM. In order to solve the non-convex problem existing in indefinite kernels, DC algorithm [24] is used to decompose the objective function into the subtraction of two convex functions on ITWSVM. Therefore, ITWSVM can both use PSD and indefinite kernels. A line search along the descent direction under the Armijo type rule is used in the DC algorithm to accelerate the convergence rate. We also implement a theoretical analysis to illustrate that ITWSVM-DC can converge to the local optimum and various experiments on both PSD and indefinite kernels show that our algorithm is superior to the state-of-the-art algorithms.

This paper is organized as follows. Section 2 outlines the related works including TWSVM and DC programming. Section 3 expounds the mechanisms of the ITWSVM-DC in detail including the model and convexity of ITWSVM with Representer Theorem, the decomposition of the ITWSVM with DC, the convergence of ITWSVM-DC. Section 4 is the experimental results and analysis. The superiority and convergence of ITWSVM-DC are verified through experiments on real-world and artificial datasets. Conclusions are given in the last section.

2. Related work

2.1. TWSVM

For a binary classification problem, given a training set $(\boldsymbol{x}_i, y_i), i = 1, 2, \ldots, n$ where $\boldsymbol{x}_i \in \mathcal{X}$ and $y_i \in \{-1, +1\}$. n is the number of training samples and m is the dimension of training samples. There are n_1 samples belonging to class $+1$ and n_2 samples belonging to class -1 in the n-dimensional real space $\mathcal{X}$. For the linear separable binary classification problem, the goal of TWSVM is to find two non-parallel hyperplanes

$$\boldsymbol{x}_1^T \boldsymbol{w}_1 + b_1 = 0 \quad and \quad \boldsymbol{x}_2^T \boldsymbol{w}_2 + b_2 = 0. \tag{1}$$

The model of TWSVM makes each hyperplane closer to the pattern of one class and as far as possible from the other. The hyperplanes are generally obtained by solving the following QP problems

$$\begin{aligned} (\text{TWSVM1}) \quad & \min_{\boldsymbol{w}_1, b_1} \frac{1}{2}(\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}_1 b_1)^T (\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}_1 b_1) + c_1 \boldsymbol{e}_2^T \boldsymbol{\xi}, \\ & \text{s.t.} \ -(\boldsymbol{B}\boldsymbol{w}_1 + \boldsymbol{e}_2 b_1) + \boldsymbol{\xi} \geq \boldsymbol{e}_2, \ \ \boldsymbol{\xi} \geq \boldsymbol{0}. \end{aligned} \tag{2}$$

$$\begin{aligned} (\text{TWSVM2}) \quad & \min_{\boldsymbol{w}_2, b_2} \frac{1}{2}(\boldsymbol{B}\boldsymbol{w}_2 + \boldsymbol{e}_2 b_2)^T (\boldsymbol{B}\boldsymbol{w}_2 + \boldsymbol{e}_2 b_2) + c_2 \boldsymbol{e}_1^T \boldsymbol{\eta}, \\ & \text{s.t.} \ (\boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}_1 b_2) + \boldsymbol{\eta} \geq \boldsymbol{e}_1, \ \ \boldsymbol{\eta} \geq \boldsymbol{0}. \end{aligned} \tag{3}$$

where c_1 and c_2 are penalty variables, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are column vectors of ones, $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are slack variables and the matrices $\boldsymbol{A}$ in $R^{n_1 \times m}$ and $\boldsymbol{B}$ in $R^{n_2 \times m}$ are training sample matrices composed of positive class and negative class respectively.

For non-linear problem, by using kernel functions, data samples can be implicitly mapped from low-dimensional space to high-dimensional feature space, thus transforming the linear inseparable problem in low-dimensional space into a linear separable problem in high-dimensional space. $\phi(\boldsymbol{x})$ is defined as the mapping function from the input space $\mathcal{X}$ to the feature space $\mathcal{H}$. $K(\boldsymbol{x}, \boldsymbol{z})$ is defined as $K(\boldsymbol{x}, \boldsymbol{z}) = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{z})$. Generally, we use the Radial Basis Function (RBF) as the kernel function.

By introducing the kernel function to TWSVM and constructing matric $\boldsymbol{C}$, i.e., $\boldsymbol{C}^T = [\boldsymbol{A} \ \boldsymbol{B}]^T$, the counterpart of the problem (2) and (3) should be

$$\begin{aligned} (\text{TWSVM1}) \quad & \min_{\boldsymbol{u}_1, b_1} \frac{1}{2}(K(\boldsymbol{A}, \boldsymbol{C}^T)\boldsymbol{u}_1 + \boldsymbol{u}_1 b_1)^T (K(\boldsymbol{A}, \boldsymbol{C}^T)\boldsymbol{u}_1 + \boldsymbol{e}_1 b_1) + c_1 \boldsymbol{e}_2^T \boldsymbol{\xi}, \\ & \text{s.t.} \ (K(\boldsymbol{B}, \boldsymbol{C}^T)\boldsymbol{u}_1 + \boldsymbol{e}_2 b_1) + \boldsymbol{\xi} \geq \boldsymbol{e}_2, \ \ \boldsymbol{\xi} \geq \boldsymbol{0}. \end{aligned} \tag{4}$$

$$\begin{aligned} (\text{TWSVM2}) \quad & \min_{\boldsymbol{u}_2, b_2} \frac{1}{2}(K(\boldsymbol{B}, \boldsymbol{C}^T)\boldsymbol{u}_2 + \boldsymbol{e}_2 b_2)^T (K(\boldsymbol{B}, \boldsymbol{C}^T)\boldsymbol{u}_2 + \boldsymbol{e}_2 b_2) + c_2 \boldsymbol{e}_1^T \boldsymbol{\eta}, \\ & \text{s.t.} \ (K(\boldsymbol{A}, \boldsymbol{C}^T)\boldsymbol{u}_2 + \boldsymbol{e}_1 b_2) + \boldsymbol{\eta} \geq \boldsymbol{e}_1, \ \ \boldsymbol{\eta} \geq \boldsymbol{0}. \end{aligned} \tag{5}$$

where c_1 and c_2 are penalty variables, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are column vectors of ones, and $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ are slack variables.

Take Eq. (4) for example, the Lagrangian of Eq. (4) is

$$\begin{aligned} L(\boldsymbol{u}_1, b_1, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = & \frac{1}{2}\left\|K(\boldsymbol{A}, \boldsymbol{C}^T)\boldsymbol{u}_1 + \boldsymbol{e}_1 b_1\right\|^2 + c_1 \boldsymbol{e}_2^T \boldsymbol{\xi} + \boldsymbol{\alpha}^T (K(\boldsymbol{B}, \boldsymbol{C}^T)\boldsymbol{u}_1 \\ & + \boldsymbol{e}_2 b_1 - \boldsymbol{\xi} + \boldsymbol{e}_2) - \boldsymbol{\beta}^T \boldsymbol{\xi}, \end{aligned} \tag{6}$$

where $\boldsymbol{\alpha}$ is Lagrangian multiplier.

By utilizing KKT Conditions, we can achieve

$$\begin{aligned} (\text{TWSVM1}) \quad & \max_{\boldsymbol{\alpha}} \ \boldsymbol{e}_2^T \boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{V}\left(\boldsymbol{S}^T \boldsymbol{S}\right)^{-1} \boldsymbol{V}^T \boldsymbol{\alpha} \\ & \text{s.t.} \ \boldsymbol{0} \leq \boldsymbol{\alpha} \leq c_1 \boldsymbol{e}_2, \end{aligned} \tag{7}$$

where $S = [K(\boldsymbol{A}, \boldsymbol{C}^T) \ \ \boldsymbol{e}_1]$, $V = [K(\boldsymbol{B}, \boldsymbol{C}^T) \ \ \boldsymbol{e}_2]$.

Similarly,

$$\begin{aligned} (\text{TWSVM2}) \quad & \max_{\boldsymbol{\gamma}} \ \boldsymbol{e}_1^T \boldsymbol{\gamma} - \frac{1}{2}\boldsymbol{\gamma}^T \boldsymbol{S}\left(\boldsymbol{V}^T \boldsymbol{V}\right)^{-1} \boldsymbol{S}^T \boldsymbol{\gamma} \\ & \text{s.t.} \ \boldsymbol{0} \leq \boldsymbol{\gamma} \leq c_2 \boldsymbol{e}_1, \end{aligned} \tag{8}$$

where $\boldsymbol{\gamma}$ is Lagrange multiplier similar to $\boldsymbol{\alpha}$ in TWSVM1.

Thus, each class corresponds to a hyperplane, and the class where the sample point belonging to is determined by the following formula.

$$\text{Class } (\boldsymbol{x}^*) = \arg\min_{\text{i}=1,2} \frac{\left|\text{K}(\boldsymbol{x}^{*\text{T}}, \boldsymbol{C}^{\text{T}})\boldsymbol{u}_{\text{i}} + \text{b}_{\text{i}}\right|}{\sqrt{\boldsymbol{u}_{\text{i}}^{\text{T}} \text{K}(\boldsymbol{C}, \boldsymbol{C}^{\text{T}})\boldsymbol{u}_{\text{i}}}}, \tag{9}$$

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

2.2. DC programming

DC Algorithm (DCA) is widely applied to many non-differentiable nonconvex optimization problems. In these problems, DCA is often adopted for global solutions and proved to be more robust and more efficient than related standard methods [24]. The particular structure of DC programming has been permitted as a good deal of development both in qualitative and quantitative studies [25].

The DC programming and DCA can address the non-convex problem by decomposing it into two convex functions, which can be written as:

$$f(x) = g(x) - h(x), \tag{10}$$

where g, h are lower semi-continuous proper convex functions on R^n.

A DC program is in the form of

$$(P_{dc})\ \alpha = \inf\{f(x) := g(x) - h(x) : x \text{ in } X\}, \tag{11}$$

where g and h belong to $\Gamma_o(X)$ which is a set of all proper lower semi-continuous convex functions on X.

By introducing conjugate functions, we have

$$\begin{aligned}\alpha &= \inf\{g(x) - h(x) : x \text{ in } X\}\\ &= \inf\{g(x) - \sup\{\langle x, y\rangle - h^*(y) : y \text{ in } Y\} : x \text{ in } X\},\end{aligned} \tag{12}$$

where Y is the dual space of X. We state the dual problem of Eq. (11)

$$(D_{dc})\ \alpha = \inf\{h^*(x) - g^*(x) : y \text{ in } Y\}, \tag{13}$$

where g^*, h^* denote the conjugate functions of g and h, respectively.

The transportation of global solutions between (P_{dc}) and (D_{dc}) is expressed as:

1. If x^* is an optimal solution of (P_{dc}), then y^* in $\partial h(x^*)$ is an optimal solution of (D_{dc}).
2. If y^* is an optimal solution of (D_{dc}), then x^* in $\partial g^*(y^*)$ is an optimal solution of (P_{dc}).

The variables x and y satisfy

$$y \in \partial h(x), \tag{14}$$

$$x \in \partial g^*(y), \tag{15}$$

where $y \in \partial h(x)$ and $x \in \partial g^*(y)$ are the sub-gradients [26] of h and g^* respectively. Then, DCA consists in the construction of two sequences $\{x_k\}$ and $\{y_k\}$, which are candidates to be optimal solutions of primal and dual programs respectively. Therefore, the sequences $\{g(x_k) - h(x_k)\}$ and $\{h^*(y_k) - g^*(y_k)\}$ are decreasing, $\{x_k\}$ (resp. $\{y_k\}$) converges to a primal feasible solution x^* (resp. a dual feasible solution y^*) verifying local optimality conditions and x^* in $\partial g^*(y^*)$, y^* in $\partial h(x^*)$.

3. ITWSVM-DC

3.1. The regularized TWSVM

3.1.1. The model of the regularized TWSVM

In this section, we introduce a regularization item to TWSVM to make sure that the model is structural risk minimization. We modify the QP problems (4) and (5) with an additional "margin" between the proximal hyperplanes ($\boldsymbol{x}^T\boldsymbol{w}_i + b_i = 0\ (i = 1, 2)$) to ensure hyperplane of one class as far as possible away from the other class. In order to make the regularized TWSVM (ITWSVM) continuously differentiable and more resilient to indefinite kernels, we introduce the smooth quadratic hinge loss function to our model.

More precisely, our QP problems are

$$\text{(ITWSVM1)}\quad \min_{\boldsymbol{w}_1, b_1}\ \frac{1}{2}\|\boldsymbol{w}_1\|^2 + \frac{1}{2}(\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}_1 b_1)^T(\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}_1 b_1) + c_1\boldsymbol{\xi}^T\boldsymbol{\xi},\quad \text{s.t. } (\boldsymbol{B}\boldsymbol{w}_1 + \boldsymbol{e}_2 b_1) + \boldsymbol{\xi} \geq \boldsymbol{e}_2,\ \ \boldsymbol{\xi} \geq \boldsymbol{0}. \tag{16}$$

$$\text{(ITWSVM2)}\quad \min_{\boldsymbol{w}_2, b_2}\ \frac{1}{2}\|\boldsymbol{w}_2\|^2 + \frac{1}{2}(\boldsymbol{B}\boldsymbol{w}_2 + \boldsymbol{e}_2 b_2)^T(\boldsymbol{B}\boldsymbol{w}_2 + \boldsymbol{e}_2 b_2) + c_2\boldsymbol{\eta}^T\boldsymbol{\eta},\quad \text{s.t. } (\boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}_1 b_2) + \boldsymbol{\eta} \geq \boldsymbol{e}_1,\ \ \boldsymbol{\eta} \geq \boldsymbol{0}. \tag{17}$$

From Eqs. (16) and (17), the distance between proximal hyperplanes $\boldsymbol{x}^T\boldsymbol{w}_i + b_i = 0\ \ (i = 1, 2)$ and the bounding hyperplanes $\boldsymbol{x}^T\boldsymbol{w}_i + b_i = \pm 1\ \ (i = 1, 2)$ is $\frac{1}{\|\boldsymbol{w}_i\|}\ \ (i = 1, 2)$. Therefore, the extra term in the objective function implies to separate the proximal and the bounding hyperplanes away as far as possible [27]. Finally, ITWSVM has the same advantages as the standard SVM, this strategy leads our method to be more theoretically sound than the original TWSVM. In the model of the ITWSVM, we also use the smooth quadratic hinge loss function on slack term $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ to make this model continuously differentiable. Then, we reformulate Eqs. (16) and (17) as unconstrained optimization problems:

$$\begin{aligned}\text{(ITWSVM1)}\quad \min_{\boldsymbol{w}_1, b_1}\ & \gamma\langle\boldsymbol{w}_1, \boldsymbol{w}_1\rangle + \frac{1}{2}\|\boldsymbol{A}\boldsymbol{w}_1 + \boldsymbol{e}_1 b_1\|^2\\ & + c_1\|\max(0, \boldsymbol{e}_2 + \boldsymbol{B}\boldsymbol{w}_1 + \boldsymbol{e}_2 b_1)\|^2,\\ & = \gamma\langle\boldsymbol{w}_1, \boldsymbol{w}_1\rangle + \sum_{i=1}^{n} V_1(\langle\boldsymbol{w}_1, \boldsymbol{x}_i\rangle + b_1).\end{aligned} \tag{18}$$

$$\begin{aligned}\text{(ITWSVM2)}\quad \min_{\boldsymbol{w}_2, b_2}\ & \gamma\langle\boldsymbol{w}_2, \boldsymbol{w}_2\rangle + \frac{1}{2}\|\boldsymbol{B}\boldsymbol{w}_2 + \boldsymbol{e}_2 b_2\|^2\\ & + c_2\|\max(0, \boldsymbol{e}_1 + \boldsymbol{A}\boldsymbol{w}_2 + \boldsymbol{e}_1 b_2)\|^2,\\ & = \gamma\langle\boldsymbol{w}_2, \boldsymbol{w}_2\rangle + \sum_{i=1}^{n} V_2(\langle\boldsymbol{w}_2, \boldsymbol{x}_i\rangle + b_2).\end{aligned} \tag{19}$$

From Eqs. (18) and (19), for each of ITWSVM, it can be divided into two parts: the regularized term $\gamma\langle\boldsymbol{w}, \boldsymbol{w}\rangle$ and loss function term $\sum_{i=1}^{n} V(\langle\boldsymbol{w}, \boldsymbol{x}_i\rangle + b)$.

3.1.2. The regularized TWSVM with representer theorem

According to the Representer Theorem [28], we can extend (18) and (19) with kernel in Reproducing Kernel Hilbert Spaces(RKHS) which can be rewritten as

$$\text{(ITWSVM1)}\quad \min_{\boldsymbol{f}_1, b_1} \gamma\langle\boldsymbol{f}_1, \boldsymbol{f}_1\rangle_{\mathcal{K}} + \sum_{i=1}^{n} V_1(\boldsymbol{f}_1(\boldsymbol{x}_i) + b_1). \tag{20}$$

$$\text{(ITWSVM2)}\quad \min_{\boldsymbol{f}_2, b_2} \gamma\langle\boldsymbol{f}_2, \boldsymbol{f}_2\rangle_{\mathcal{K}} + \sum_{i=1}^{n} V_2(\boldsymbol{f}_2(\boldsymbol{x}_i) + b_2). \tag{21}$$

Take ITWSVM1 for example, $\gamma\langle\boldsymbol{w}_1, \boldsymbol{w}_1\rangle$ can be represented as $\gamma\langle\boldsymbol{f}_1, \boldsymbol{f}_1\rangle_{\mathcal{K}}$ and V_1 is a loss function.

When the kernel is indefinite, (20) and (21) can be extended in a wilder Reproducing Kernel Kreĭn Spaces (RKKS) [29]. In RKKS, the Representer Theorem is verified to still hold and the problem of minimizing a regularized risk function can be expanded as

$$f^* = \sum_{i=1}^{n} \beta_i K(\boldsymbol{x}_i, \cdot), \tag{22}$$

where the coefficient $\beta_i \in \mathbb{R}$ and K is a kernel function in RKKS.

We can further attain the model of ITWSVM1 in RKKS:

$$\text{(ITWSVM1)}\quad \min_{\boldsymbol{\beta}, b_1} \gamma\boldsymbol{\beta}^T\boldsymbol{K}\boldsymbol{\beta} + \sum_{i=1}^{n} V_1\left(\boldsymbol{K}^i\boldsymbol{\beta} + b_1\right), \tag{23}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, .\beta_n]^T$, $\boldsymbol{K}$ is the indefinite kernel matrix derived from corresponding kernel function $\boldsymbol{K}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{K}^i$ represents the ith row of $\boldsymbol{K}$.

Note that:

$$\sum_{i=1}^{n} V_1\left(\boldsymbol{K}^i\boldsymbol{\beta} + b_1\right) = \sum_{i=1}^{n_1}\left(\sum_{j=1}^{n}\beta_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b_1\right)^2$$

$$+\sum_{i=n_1+1}^{n_1+n_2} c_1 \max\left(0, 1+\sum_{j=1}^{n}\beta_j K(\boldsymbol{x}_i,\boldsymbol{x}_j)+b_1\right)^2$$
$$=\sum_{i=1}^{n_1}\left(\boldsymbol{K}^i\boldsymbol{\beta}+b_1\right)^2+\sum_{i=n_1+1}^{n_1+n_2} c_1\max\left(0,\boldsymbol{K}^i\boldsymbol{\beta}+b_1+1\right)^2, \quad (24)$$

where n_1 is the number of samples belonging to class $+1$ and n_2 is the number of samples belonging to class -1, $n=n_1+n_2$. To distinguish $\boldsymbol{\beta}$ in ITWSVM1 and ITWSVM2, we set $\boldsymbol{\beta}$ as $\boldsymbol{\beta}_1$ in ITWSVM1 and $\boldsymbol{\beta}_2$ in ITWSVM2 respectively. The optimization problem by the scaling constant $1/2$ is given by

$$(\text{ITWSVM1}) \quad \min_{\boldsymbol{\beta}_1,b_1}\frac{1}{2}\gamma\boldsymbol{\beta}_1^T\boldsymbol{K}\boldsymbol{\beta}_1 + \underbrace{\frac{1}{2}\left(\sum_{i=1}^{n_1}\left(\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1\right)^2+\sum_{i=n_1+1}^{n_1+n_2}c_1\max\left(0,\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1+1\right)^2\right)}_{\sum_{i=1}^{n}V_1(\boldsymbol{f}_1(\boldsymbol{x}_i)+b_1)}. \quad (25)$$

$$(\text{ITWSVM2}) \quad \min_{\boldsymbol{\beta}_2,b_2}\frac{1}{2}\gamma\boldsymbol{\beta}_2^T\boldsymbol{K}\boldsymbol{\beta}_2 + \underbrace{\frac{1}{2}\left(\sum_{i=1}^{n_1}\left(\boldsymbol{K}^i\boldsymbol{\beta}_2+b_2\right)^2+\sum_{i=n_1+1}^{n_1+n_2}c_2\max\left(0,\boldsymbol{K}^i\boldsymbol{\beta}_2+b_2+1\right)^2\right)}_{\sum_{i=1}^{n}V_2(\boldsymbol{f}_2(\boldsymbol{x}_i)+b_2)}. \quad (26)$$

3.1.3. Analysis of convexity

In this section, we will present a theoretical analysis for the convexity of ITWSVM. In order to better solve the problem, we also divide ITWSVM into two parts: the regularized term $\frac{1}{2}\gamma\boldsymbol{\beta}^T\boldsymbol{K}\boldsymbol{\beta}$ and loss function term $\sum_{i=1}^{n}V(\boldsymbol{f}(\boldsymbol{x}_i)+b)$.

By introducing the convex optimization theory [30], we have the convex Theorem 1.

Theorem 3.1. *If f is twice differentiable, that is, its Hessian or second derivative $\nabla^2 f$ exists at each point in $\mathbf{dom} f$, which is open. Then f is convex if and only if $\mathbf{dom} f$ is convex and its Hessian is positive semidefinite: for all $x\in\mathbf{dom} f$,*

$$\nabla^2 f \succeq 0.$$

According to Theorem 3.1, we can deduce that

Proposition 3.1. *The convexity of ITWSVM model is determined by the regularized term $\frac{1}{2}\gamma\boldsymbol{\beta}^T\boldsymbol{K}\boldsymbol{\beta}$ according to kernel $\boldsymbol{K}$.*

Proof. Take ITWSVM1 for example, for the regularized term $\frac{1}{2}\gamma\boldsymbol{\beta}_1^T\boldsymbol{K}\boldsymbol{\beta}_1$, its Hessian or second derivative is $\boldsymbol{K}$. Therefore, the convexity is determined by kernel $\boldsymbol{K}$. If $\boldsymbol{K}$ is positive semi-definite, $\frac{1}{2}\gamma\boldsymbol{\beta}_1^T\boldsymbol{K}\boldsymbol{\beta}_1$ is convex and non-convex otherwise.

For the loss function term $\sum_{i=1}^{n}V_1(\boldsymbol{f}_1(\boldsymbol{x}_i)+b_1)$, we carry out convex analysis for its two parts $\sum_{i=1}^{n_1}\left(\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1\right)^2$ and $\sum_{i=n_1+1}^{n_1+n_2}c_1\max\left(0,\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1\right)^2$ respectively.

$$\sum_{i=1}^{n_1}\left(\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1\right)^2=\sum_{i=1}^{n_1}\left(\left(\boldsymbol{K}^i\boldsymbol{\beta}_1\right)^T\left(\boldsymbol{K}^i\boldsymbol{\beta}_1\right)+2b_1\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1^2\right)$$
$$=\sum_{i=1}^{n_1}\left(\boldsymbol{\beta}_1^T\boldsymbol{K}^{iT}\boldsymbol{K}^i\boldsymbol{\beta}_1+2b_1\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1^2\right). \quad (27)$$

Its Hessian or second derivative is $\boldsymbol{K}^{iT}\boldsymbol{K}^i$.

$$\sum_{i=n_1+1}^{n_1+n_2}c_1\max\left(0,\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1+1\right)^2$$
$$=\sum_{i=n_1+1}^{n_1+n_2}c_1\max\left(0,\left(\boldsymbol{K}^i\boldsymbol{\beta}_1\right)^T\left(\boldsymbol{K}^i\boldsymbol{\beta}_1\right)+2(b_1+1)\boldsymbol{K}^i\boldsymbol{\beta}_1+(b_1+1)^2\right)$$
$$=\sum_{i=n_1+1}^{n_1+n_2}c_1\max\left(0,\boldsymbol{\beta}_1^T\boldsymbol{K}^{iT}\boldsymbol{K}^i\boldsymbol{\beta}_1+2(b_1+1)\boldsymbol{K}^i\boldsymbol{\beta}_1+(b_1+1)^2\right). \quad (28)$$

Its Hessian or second derivative is $\boldsymbol{K}^{iT}\boldsymbol{K}^i$.

Noted that $\boldsymbol{K}^{iT}\boldsymbol{K}^i\succeq 0$ is positive semi-definite, therefore, the quadratic form $\boldsymbol{\beta}_1^T\boldsymbol{K}^{iT}\boldsymbol{K}^i\boldsymbol{\beta}_1+2b_1\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1^2$ and $\max(0,\boldsymbol{\beta}_1^T\boldsymbol{K}^{iT}\boldsymbol{K}_1^i\boldsymbol{\beta}_1+2(b_1+1)\boldsymbol{K}^i\boldsymbol{\beta}_1+(b_1+1)^2)$ in loss function is convex. Then, the convexity of the two part of loss function $\sum_{i=1}^{n_1}(\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1)^2$ and $\sum_{i=n_1+1}^{n_1+n_2}c_1\max(0,\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1)^2$ can be proved. Therefore, the loss function term $\sum_{i=1}^{n}V_1(\boldsymbol{f}_1(\boldsymbol{x}_i)+b_1)$ is convex.

Therefore, the convexity of ITWSVM1 is determined by the regularized term $\frac{1}{2}\gamma\boldsymbol{\beta}_1^T\boldsymbol{K}\boldsymbol{\beta}_1$ according to kernel $\boldsymbol{K}$. Similarly, the convexity of ITWSVM2 is determined by the regularized term $\frac{1}{2}\gamma\boldsymbol{\beta}_2^T\boldsymbol{K}\boldsymbol{\beta}_2$ according to kernel $\boldsymbol{K}$. □

3.2. ITWSVM with DC algorithm

In the last section, we analyze the convexity of ITWSVM. However, If the kernel $\boldsymbol{K}$ is indefinite, the ITWSVM is non-convex and traditional methods for solving the dual problem of TWSVM is not suitable for ITWSVM and there is a dual gap between the primal problem and the dual problem.

In this section, we optimize the ITWSVM model obtained in Section 3.1 with DC algorithm. Both PSD kernels and indefinite kernel can be applied to our algorithm. ITWSVM model can be noted as:

$$\begin{cases}(\text{ITWSVM1}) & \min\limits_{\boldsymbol{\beta}_1,b_1}\frac{1}{2}\gamma\boldsymbol{\beta}_1^T\boldsymbol{K}\boldsymbol{\beta}_1 + \frac{1}{2}\left(\sum\limits_{i=1}^{n_1}\left(\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1\right)^2+\sum\limits_{i=n_1+1}^{n_1+n_2}c_1\max\left(0,\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1+1\right)^2\right)\\ (\text{ITWSVM2}) & \min\limits_{\boldsymbol{\beta}_2,b_2}\frac{1}{2}\gamma\boldsymbol{\beta}_2^T\boldsymbol{K}\boldsymbol{\beta}_2 + \frac{1}{2}\left(\sum\limits_{i=1}^{n_1}\left(\boldsymbol{K}^i\boldsymbol{\beta}_2+b_2\right)^2+\sum\limits_{i=n_1+1}^{n_1+n_2}c_2\max\left(0,\boldsymbol{K}^i\boldsymbol{\beta}_2+b_2+1\right)^2\right)\end{cases}. \quad (29)$$

The objective functions of ITWSVM are

$$\begin{cases}f(\boldsymbol{\beta}_1)=\frac{1}{2}\gamma\boldsymbol{\beta}_1^T\boldsymbol{K}\boldsymbol{\beta}_1 + \frac{1}{2}\left(\sum\limits_{i=1}^{n_1}\left(\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1\right)^2+\sum\limits_{i=n_1+1}^{n_1+n_2}c_1\max\left(0,\boldsymbol{K}^i\boldsymbol{\beta}_1+b_1+1\right)^2\right)\\ f(\boldsymbol{\beta}_2)=\frac{1}{2}\gamma\boldsymbol{\beta}_2^T\boldsymbol{K}\boldsymbol{\beta}_2 + \frac{1}{2}\left(\sum\limits_{i=1}^{n_1}\left(\boldsymbol{K}^i\boldsymbol{\beta}_2+b_2\right)^2+\sum\limits_{i=n_1+1}^{n_1+n_2}c_2\max\left(0,\boldsymbol{K}^i\boldsymbol{\beta}_2+b_2+1\right)^2\right)\end{cases}. \quad (30)$$

The eigenspectrum of the indefinite kernel matrix can be noted as $\boldsymbol{K}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{U}$ represents the orthogonal column eigenvector matrix and $\boldsymbol{\Lambda}$ represent the diagonal eigenvalue matrix respectively. Due to the kernel matrix is indefinite, $\boldsymbol{\Lambda}$ contains both positive and negative eigenvalues. After shifting the eigenspectrum of the indefinite kernels, we can achieve several equivalent decompositions on Eq. (30). The basic idea adopted in this paper is to decompose the objective function into $f(\boldsymbol{\beta})=g(\boldsymbol{\beta})-h(\boldsymbol{\beta})$. Specifically, the following two decomposition methods are adopted:

$$① \begin{cases}g_1(\boldsymbol{\beta}_1)=\frac{1}{2}\left(\gamma\boldsymbol{\beta}_1^T\boldsymbol{U}_1(\rho_1\boldsymbol{I}+\boldsymbol{\Lambda}_1)\boldsymbol{U}_1^T\boldsymbol{\beta}_1\right)+\sum\limits_{i=1}^{n}V_1(\boldsymbol{f}_1(\boldsymbol{x}_i)+b_1)\\ h_1(\boldsymbol{\beta}_1)=\frac{1}{2}\gamma\boldsymbol{\beta}_1^T\boldsymbol{U}_1(\rho_1\boldsymbol{I})\boldsymbol{U}_1^T\boldsymbol{\beta}_1\\ g_2(\boldsymbol{\beta}_2)=\frac{1}{2}\left(\gamma\boldsymbol{\beta}_2^T\boldsymbol{U}_2(\rho_2\boldsymbol{I}+\boldsymbol{\Lambda}_2)\boldsymbol{U}_2^T\boldsymbol{\beta}_2\right)+\sum\limits_{i=1}^{n}V_2(\boldsymbol{f}_2(\boldsymbol{x}_i)+b_2)\\ h_2(\boldsymbol{\beta}_2)=\frac{1}{2}\gamma\boldsymbol{\beta}_2^T\boldsymbol{U}_2(\rho_2\boldsymbol{I})\boldsymbol{U}_2^T\boldsymbol{\beta}_2\end{cases}, \quad (31)$$

$$\textcircled{2}\begin{cases} g_1(\boldsymbol{\beta}_1) = \frac{1}{2}\left(\gamma \boldsymbol{\beta}_1^T \boldsymbol{U}_1(\rho_1' \boldsymbol{I})\boldsymbol{U}_1^T \boldsymbol{\beta}_1\right) + \sum_{i=1}^{n} V_1(\boldsymbol{f}_1(\boldsymbol{x}_i) + b_1) \\ h_1(\boldsymbol{\beta}_1) = \frac{1}{2}\gamma \boldsymbol{\beta}_1^T \boldsymbol{U}_1(\rho_1' \boldsymbol{I}_1 - \boldsymbol{\Lambda}_1)\boldsymbol{U}_1^T \boldsymbol{\beta}_1 \\ g_2(\boldsymbol{\beta}_2) = \frac{1}{2}\left(\gamma \boldsymbol{\beta}_2^T \boldsymbol{U}_2(\rho_2' \boldsymbol{I})\boldsymbol{U}_2^T \boldsymbol{\beta}_2\right) + \sum_{i=1}^{n} V_2(\boldsymbol{f}_2(\boldsymbol{x}_i) + b_2) \\ h_2(\boldsymbol{\beta}_2) = \frac{1}{2}\gamma \boldsymbol{\beta}_2^T \boldsymbol{U}_2(\rho_2' \boldsymbol{I} - \boldsymbol{\Lambda}_2)\boldsymbol{U}_2^T \boldsymbol{\beta}_2 \end{cases}, \tag{32}$$

where

$$\sum\nolimits_{i=1}^{n} V_1(\boldsymbol{f}_1(\boldsymbol{x}_i) + b_1) = \frac{1}{2}\left(\sum\nolimits_{i=1}^{n_1} \left(\boldsymbol{K}^i \boldsymbol{\beta}_1 + b_1\right)^2 + \sum\nolimits_{i=n_1+1}^{n_1+n_2} c_1 \max\left(0, \boldsymbol{K}^i \boldsymbol{\beta}_1 + b_1 + 1\right)^2\right) \tag{33}$$

and

$$\sum\nolimits_{i=1}^{n} V_2(f_2(x_i) + b_2) = \frac{1}{2}\left(\sum\nolimits_{i=1}^{n_1} \left(\boldsymbol{K}^i \boldsymbol{\beta}_2 + b_2\right)^2 + \sum\nolimits_{i=n_1+1}^{n_1+n_2} c_2 \max\left(0, \boldsymbol{K}^i \boldsymbol{\beta}_2 + b_2 + 1\right)^2\right) \tag{34}$$

have been proved convex in Section 3.1.3. $\{\lambda_i^1\}_{i=1}^n$ are noted as the eigenvalues in the eigenvalue matrix $\boldsymbol{\Lambda}_1$, $\{\lambda_i^2\}_{i=1}^n$ are noted as the eigenvalues in the eigenvalue matrix $\boldsymbol{\Lambda}_2$, $\rho_1 \geq -\min(\{\lambda_i^1\}_{i=1}^n)$, $\rho_2 \geq -\min(\{\lambda_i^2\}_{i=1}^n)$, $\rho_1' \geq \max(\{\lambda_i^1\}_{i=1}^n)$, $\rho_2' \geq \max(\{\lambda_i^2\}_{i=1}^n)$. Theses positive numbers ρ_1, ρ_2, ρ_1' and ρ_2' are used to ensure the convexity of these four functions $g_1(\boldsymbol{\beta}_1)$, $h_1(\boldsymbol{\beta}_1)$, $g_2(\boldsymbol{\beta}_2)$ and $h_1(\boldsymbol{\beta}_2)$.

In order to avoid the repetitive complex solving process, we use $\boldsymbol{\beta}$ to simultaneously represent $\boldsymbol{\beta}_1$ in ITWSVM1 and $\boldsymbol{\beta}_2$ in ITWSVM2.

According to the theory of DC programming, we can get the conjugate dual problem [31,32] of function $f(\boldsymbol{\beta}) : \inf\{f^*(\boldsymbol{\theta}) = h^*(\boldsymbol{\theta}) - g^*(\boldsymbol{\theta})\}$. According to Eqs. (14) and (15), we can obtain:

$$\begin{cases} \boldsymbol{\theta} \in \partial h(\boldsymbol{\beta}) \\ \boldsymbol{\beta} \in \partial g^*(\boldsymbol{\theta}) \end{cases}. \tag{35}$$

Function $h(\boldsymbol{\beta})$ and $g^*(\boldsymbol{\theta})$ can be noted as:

$$\begin{cases} h(\boldsymbol{\beta}) = h(\boldsymbol{\beta}^t) + \left\langle \boldsymbol{\beta} - \boldsymbol{\beta}^t, \boldsymbol{\theta}^t \right\rangle \\ g^*(\boldsymbol{\theta}) = g^*(\boldsymbol{\theta}^t) + \left\langle \boldsymbol{\theta} - \boldsymbol{\theta}^t, \boldsymbol{\beta}^{t+1} \right\rangle \end{cases} \tag{36}$$

in $\boldsymbol{\beta}^t, \boldsymbol{\theta}$. In Eq. (36), $\boldsymbol{\theta}^t \in \partial h(\boldsymbol{\beta}^t)$ and $\boldsymbol{\beta}^{t+1} \in \partial g^*(\boldsymbol{\theta}^t)$. In this way, the problem is transformed into an iterative solution method to the sequences $\{\boldsymbol{\beta}^t\}$ and $\{\boldsymbol{\theta}^t\}$:

$$\begin{cases} \left\{\boldsymbol{\beta}^t\right\} = \arg\min\left\{\boldsymbol{\beta}^{t+1} : g(\boldsymbol{\beta}) - \left\langle \boldsymbol{\beta}, \boldsymbol{\theta}^t \right\rangle, \boldsymbol{\beta} \in R^n\right\} \\ \left\{\boldsymbol{\theta}^t\right\} = \arg\min\left\{\boldsymbol{\theta}^{t+1} : h^*(\boldsymbol{\theta}) - \left\langle \boldsymbol{\theta}, \boldsymbol{\beta}^{t+1} \right\rangle, \boldsymbol{\theta} \in R^n\right\} \end{cases}. \tag{37}$$

According to the research result of DC programming [33], the model requires to optimize six parameters: $\boldsymbol{\beta}_1$, b_1, $\boldsymbol{\theta}_1$, $\boldsymbol{\beta}_2$, b_2 and $\boldsymbol{\theta}_2$, where the optimal iteration formulas of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$ are:

$$\begin{cases} \boldsymbol{\theta}^t \in \partial h(\boldsymbol{\beta}^t) \\ \boldsymbol{\beta}^{t+1} \in \arg\min_{\boldsymbol{\beta} \in R^n} g(\boldsymbol{\beta}^t) - \left\langle \boldsymbol{\beta}^t, \boldsymbol{\theta}^t \right\rangle \end{cases}. \tag{38}$$

In each iteration, the sequence $\{\boldsymbol{\beta}^t\}$ can generate one descent direction. In order to accelerate the convergence rate of the algorithm, the Armijo type rule along the descent direction is used to search the smallest non-negative integer l_t to further reduce the value of the objective function:

$$f(\boldsymbol{\beta}^{t+1} + \eta^{l_t} d(\boldsymbol{\beta})) \leq f(\boldsymbol{\beta}^{t+1}) - \mu\eta^{l_t} \left\| d(\boldsymbol{\beta}) \right\|^2. \tag{39}$$

3.3. Algorithm description

Algorithm 1 The pseudo code of ITWSVM-DC algorithm is given in Algorithm 1.

Input:
D: the training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$
$\bar{\upsilon}$: the step size of Armijo Rule ($\bar{\upsilon} > 0$)
μ, η: the parameters of Armijo Rule ($0 < \mu < \eta < 1$)
T: the maximize number of iterations
$\boldsymbol{x}^*$: the test sample
Output:
$\boldsymbol{y}^*$: the predicted class label of the sample $\boldsymbol{x}^*$
Process:
Initialize the kernel coefficient β_0 and $t = 0$;
Implement DC decomposition for ITWSVM1: $f_1(\beta_1) = g_1(\beta_1) - h_1(\beta_1)$ and ITWSVM2: $f_2(\beta_2) = g_2(\beta_2) - h_2(\beta_2)$;
while $t < T$ **do**
  for ITWSVMi $i \in \{1, 2\}$ **do**
    Obtain a solution for conjugate dual problem: $\theta_i^t = \nabla h\left(\beta_i^t\right)$;
    Solve convex optimization method $\beta_i^{t+1} \in \arg\min_{\beta_i \in R^n} g(\beta_i^t) - \left\langle \beta_i^t, \theta_i^t \right\rangle$ to obtain the solution β_i^{t+1} of the primal ITWSVMi problem;
    Calculate $d(\beta_i) = \beta_i^{t+1} - \beta_i^t$;
    if $||d(\beta_i)||^2 \leq \delta$ **then**
      The model converges to the local minimum and Stop iteration;
    end if
    Set $\upsilon^t = \bar{\upsilon}$;
    while $f_i(\beta_i^{t+1} + \eta^{l_t} d(\beta_i)) \leq f_i(\beta_i^{t+1}) - \mu\eta^{l_t} \|d(\beta_i)\|^2$ **do**
      $\upsilon^t = \eta\upsilon^t$;
    end while
    Update the solution of ITWSVMi: $\beta_i^{t+1} = \beta_i^{t+1} + \upsilon^t d(\beta_i)$ and the number of iterations $t = t + 1$;
  end for
end while
return Class$(\boldsymbol{x}^*) = \arg\min_{i=1,2} \frac{\left|K(\boldsymbol{x}^{*T}, \boldsymbol{C}^T)\beta_i + b_i\right|}{\sqrt{\beta_i^T K(\boldsymbol{C}, \boldsymbol{C}^T)\beta_i}}$)

3.4. Convergence analysis

In this section, we implement a theoretical analysis for the convergence of ITWSVM-DC. Like Section 3.2, we use unified $\boldsymbol{\beta}$ to represent $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$.

Theorem 3.2. *If the sequence $\boldsymbol{\beta}^t$ satisfies $d(\boldsymbol{\beta}) = \boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t = 0$, that is, $\boldsymbol{\beta}^* = \boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t$. Then, for $\forall \boldsymbol{\beta} \in U(\boldsymbol{\beta}^*, \delta)$, we have*

$$g(\boldsymbol{\beta}) - h(\boldsymbol{\beta}) \geq g(\boldsymbol{\beta}^*) - h(\boldsymbol{\beta}^*). \tag{40}$$

Proof. For the DC programming and DCA, we can decompose the non-convex objective function into two convex function $f(x) = g(x) - h(x)$. If an additional term $\frac{\tau}{2}x^2 (\tau > 0)$ is added to the convex function g and h, it can make them strongly convex. Then

$$(g - h)(x) = \left(g(x) + \frac{\tau}{2}x^2\right) - \left(h(x) + \frac{\tau}{2}x^2\right). \tag{41}$$

Set

$$G(x) = g(x) + \frac{\tau}{2}x^2, \tag{42}$$

$$H(x) = h(x) + \frac{\tau}{2}x^2. \tag{43}$$

Then we introduce the functions to our objective function. For the strongly convexity of function, we can get

$$G\left(\boldsymbol{\beta}^t\right) \geq G\left(\boldsymbol{\beta}^{t+1}\right) + \nabla G\left(\boldsymbol{\beta}^{t+1}\right)\left(\boldsymbol{\beta}^t - \boldsymbol{\beta}^{t+1}\right)^T, \tag{44}$$

$$H\left(\boldsymbol{\beta}^{t+1}\right) \geq H\left(\boldsymbol{\beta}^t\right) + \nabla H\left(\boldsymbol{\beta}^t\right)\left(\boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t\right)^T, \tag{45}$$

$$H\left(\boldsymbol{\beta}^t\right) \geq H\left(\boldsymbol{\beta}^{t+1}\right) + \nabla H\left(\boldsymbol{\beta}^{t+1}\right)\left(\boldsymbol{\beta}^t - \boldsymbol{\beta}^{t+1}\right)^T. \tag{46}$$

According to the iteration formula $\begin{cases}\boldsymbol{\theta}^t \in \partial h(\boldsymbol{\beta}^t) \\ \boldsymbol{\beta}^{t+1} \in \arg\min_{\boldsymbol{\beta}\in R^n} g(\boldsymbol{\beta}^t) - \langle \boldsymbol{\beta}^t, \boldsymbol{\theta}^t \rangle \end{cases}$, we have $\begin{cases}\boldsymbol{\theta}^t = \partial h(\boldsymbol{\beta}^t) \\ \partial g \boldsymbol{\beta}^{t+1} = \boldsymbol{\theta}\end{cases}$, that is

$$\nabla g(\boldsymbol{\beta}^{t+1}) = \boldsymbol{\theta}^t = \nabla h(\boldsymbol{\beta}^t). \tag{47}$$

By substituting Eqs. (42) and (43) into Eqs. (44) and (45) respectively and combine Eq. (47), we have

$$(g(\boldsymbol{\beta}^t) - h(\boldsymbol{\beta}^t)) - (g(\boldsymbol{\beta}^{t+1}) - h(\boldsymbol{\beta}^{t+1})) \geq \tau \left\| \boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t \right\|^2. \tag{48}$$

The equality holds if and only if $\tau \left\| \boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t \right\|^2 = 0$, which means ITWSVM-DC can reduce the value of objective function in each iteration. When $\tau \left\| \boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t \right\|^2 = 0$, ITWSVM-DC converges. According to Eqs. (43) and (46), function $h(\boldsymbol{\beta})$ is strongly convex in R^n. According to the theory of reference [34], we have □

Theorem 3.3. *A function f is strongly convex if and only if it is continuously differentiable and for any $x, y \in R^n$, we have*

$$\left\langle f'(x) - f'(y), x - y \right\rangle \geq \mu \|x - y\|^2, \quad \mu > 0. \tag{49}$$

Proof. According to Eq. (49), we have

$$\left\langle \nabla h(\boldsymbol{\beta}^t) - \nabla h(\boldsymbol{\beta}^{t+1}), \boldsymbol{\beta}^t - \boldsymbol{\beta}^{t+1} \right\rangle \geq \tau \left\| \boldsymbol{\beta}^t - \boldsymbol{\beta}^{t+1} \right\|^2. \tag{50}$$

Substitute Eq. (47) into Eq. (50), we have

$$\langle \nabla g(\boldsymbol{\beta}^{t+1}) - \nabla h(\boldsymbol{\beta}^{t+1}), \boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t \rangle \leq \tau \left\| \boldsymbol{\beta}^t - \boldsymbol{\beta}^{t+1} \right\|^2 \leq 0. \tag{51}$$

The equality holds if and only if $\tau \left\| \boldsymbol{\beta}^t - \boldsymbol{\beta}^{t+1} \right\|^2 = 0$, which demonstrates that $d(\boldsymbol{\beta}) = \boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t = 0$ is a descent direction for the objective function $f = g - h$ at $\boldsymbol{\beta}^{t+1}$.

Setting the optimal solution of the function as $\boldsymbol{\beta}^*$, when $d(\boldsymbol{\beta}) = \boldsymbol{\beta}^{t+1} - \boldsymbol{\beta}^t = 0$, according to Eq. (47), we have $\nabla g(\boldsymbol{\beta}^*) = \nabla g(\boldsymbol{\beta}^{t+1}) = \boldsymbol{\theta}^t$, that is $\exists \boldsymbol{\theta} \in \partial g(\boldsymbol{\beta}^*)$.

So the conjugate function $\boldsymbol{g}^*$ of g at $\boldsymbol{\beta}^*$ is

$$g^*(\boldsymbol{\theta}) = \sup\left\{\left\langle \boldsymbol{\beta}^*, \boldsymbol{\theta} \right\rangle - g(\boldsymbol{\beta}^*)\right\} = \left\langle \boldsymbol{\beta}^*, \boldsymbol{\theta} \right\rangle - g(\boldsymbol{\beta}^*). \tag{52}$$

Similar to Eq. (52), $\forall \boldsymbol{\theta} \in R^n$, the conjugate function h^* of h at $\boldsymbol{\beta}^*$ is

$$h^*(\boldsymbol{\theta}) = \sup\left\{\left\langle \boldsymbol{\beta}^*, \boldsymbol{\theta} \right\rangle - h(\boldsymbol{\beta}^*)\right\} \geq \left\langle \boldsymbol{\beta}^*, \boldsymbol{\theta} \right\rangle - h(\boldsymbol{\beta}^*). \tag{53}$$

Combining Eqs. (52) and (53), we have

$$g(\boldsymbol{\beta}^*) - h(\boldsymbol{\beta}^*) \leq h^*(\boldsymbol{\theta}) - g^*(\boldsymbol{\theta}). \tag{54}$$

Due to $\boldsymbol{\theta} = \nabla h(\boldsymbol{\beta})$, that is $\exists \boldsymbol{\theta} \in \partial h(\boldsymbol{\beta})$. the conjugate function h^* of h at $\boldsymbol{\beta}^*$ is

$$h^*(\boldsymbol{\theta}) = \sup\left\{\left\langle \boldsymbol{\beta}, \boldsymbol{\theta} \right\rangle - h(\boldsymbol{\beta})\right\} = \left\langle \boldsymbol{\beta}, \boldsymbol{\theta} \right\rangle - h(\boldsymbol{\beta}). \tag{55}$$

Similar to Eq. (55), $\forall \boldsymbol{\theta} \in R^n$, the conjugate function g^* of g at $\boldsymbol{\beta}$ is

$$g^*(\boldsymbol{\theta}) = \sup\left\{\left\langle \boldsymbol{\beta}, \boldsymbol{\theta} \right\rangle - g(\boldsymbol{\beta})\right\} \geq \left\langle \boldsymbol{\beta}, \boldsymbol{\theta} \right\rangle - g(\boldsymbol{\beta}). \tag{56}$$

Combining Eqs. (55) and (56), we have

$$g(\boldsymbol{\beta}) - h(\boldsymbol{\beta}) \geq h^*(\boldsymbol{\theta}) - g^*(\boldsymbol{\theta}). \tag{57}$$

According to Eqs. (54) and (57), we obtain

$$g(\boldsymbol{\beta}) - h(\boldsymbol{\beta}) \geq g(\boldsymbol{\beta}^*) - h(\boldsymbol{\beta}^*). \tag{58}$$

Therefore, the function converges to the optimal solution $\boldsymbol{\beta}^*$. □

3.5. ITWSVM-DC for multi-class classification

In this section, we use "one-versus-rest" strategy for multi-class ITWSVM-DC [35]. For a K-class classification problem, the approach generates K hyperplanes, one hyperplane for each class. When constructing the kth hyperplane for the kth class, multi-class ITWSVM-DC takes the kth class as the positive class and considers the rest classes as negative class to construct an ITWSVM-DC-type QP Problem. Each QP problem of multi-class ITWSVM-DC is trained on all samples and generates one hyperplane. In the stage of prediction, multi-class ITWSVM-DC calculates the distances between the new sample and these hyperplanes. Then, multi-class ITWSVM-DC signs the new sample to the class corresponding to the hyperplane that the new sample is closest to. For a K-class classification problem, the model of multi-class ITWSVM-DC for the kth hyperplane is written as follows:

$$\min_{\boldsymbol{\beta}_k, b_k} \frac{1}{2}\gamma \boldsymbol{\beta}_k^T \boldsymbol{K} \boldsymbol{\beta}_k + \frac{1}{2}\left(\sum_{i=1}^{n_k}\left(\boldsymbol{K}^i \boldsymbol{\beta}_k + b_k\right)^2 + \sum_{i=n_k+1}^{n_k+n_{k'}} c_k \max\left(0, \boldsymbol{K}^i \boldsymbol{\beta}_k + b_k\right)^2\right), \tag{59}$$

where $\boldsymbol{\beta}_k$ and b_k are the parameters of the kth separating hyperplane, c_k is the penalty parameter. Then the multi-class ITWSVM-DC model can be optimized with DC algorithm as described in Section 3.2.

4. Experiments results and analysis

In this section, all algorithms are implemented in Python 3.6.5 on a PC with an Intel i5-8300H quad core processor, 8 GB RAM and Microsoft Windows 10.

4.1. Experimental setup

We present experimental results of our algorithms on UCI datasets and IDA datasets to verify the effectiveness of our algorithms. We adopt the grid search method to optimize the parameters. We choose sigmoid kernel and Radial Basis Function (RBF) as kernel functions to compare our ITWSVM-DC with other methods respectively. The definition of kernel functions (sigmoid kernel and RBF kernel) is given by

$$K(x, z) = \tanh\left(\gamma \langle x, z \rangle + \theta\right) \tag{60}$$

and

$$K(x, z) = \exp\left(-\gamma \|x - z\|^2\right) \tag{61}$$

respectively. The regularization term parameter, the parameters in sigmoid and RBF kernels and penalty parameters in SVMs and TWSVMs are selected by grid search from the set $\{2^{-6}, 2^{-5}, \cdots, 2^6\}$.

In the experiments, twenty real-world datasets are used for training models. Tables 1 and 2 gives a brief description of the used twenty datasets. Among them, the diabetis dataset are IDA benchmark dataset, and the other nineteen datasets are UCI benchmark dataset.

For all the datasets, we randomly divide the samples into two non-overlapping training and testing sets which contain almost

Table 1
Description of the datasets for binary classification.

Datasets	Number of samples	Number of dimension
australian	690	14
blood	748	4
breast	277	9
cryotherapy	90	6
customers	440	7
haberman	306	3
heart	270	13
liver	345	6
pima	768	8
planning	182	12
voting	435	16
wpbc	198	33
diabetis	768	8

Table 2
Description of the datasets for multi-class classification.

Datasets	Number of samples	Number of dimension	Number of classes
breast-tissue	106	9	6
glass	214	9	6
iris	150	4	3
seeds	210	7	3
balance	625	4	3
soybean	47	35	4
wine	178	13	3

half of the samples in each class. The processes are repeated ten times to generate ten independent epochs for each dataset, and then the mean classification accuracies, the standard deviations and training time are reported.

In order to reflect the characteristics of different algorithms and validate the performance, we perform experiments to compare our proposed regularized TWSVM (ITWSVM) and ITWSVM-DC with the original TWSVM and several state-of-the-art IKSVMs, they are:

- Clip: Treat all negative eigenvalues as noise and replace them with zero.
- Flip: Flip the sign of negative eigenvalues in K so as to form a PSD kernel matrix.
- Diffusion [36] : Consider data distribution when computing pairwise similarity.
- Shift: Add a constant to all eigenvalues to make sure all the eigenvalues are non-negative.
- IKSVM-DC: Introduce DC programming into the solution of IKSVM, which greatly improves the classification accuracy of the model.

4.2. Experimental results on binary classification datasets

First, we perform experiments on sigmoid kernel which can be viewed as one prominent representative of indefinite kernel. we compare our algorithm with Flip, Diffusion, Shift and Clip which are common forms of SVMs for solving indefinite kernel. We also

Table 3
The classification accuracy (mean ± standard deviation) and training time of binary classification of various algorithms when using the sigmoid kernel. •/◦ indicates whether the ITWSVM-DC is statistically superior/inferior to the compared models (pairwise t-test at 0.05 significance level).

Datasets		Flip	Diffusion	Shift	Clip	TWSVM	ITWSVM	IKSVM-DC	ITWSVM-DC
australian	mean(%)	85.68•	76.41•	86.81	85.59•	82.23•	86.96	86.75	**87.39**
	±std(%)	1.44	14.23	1.25	1.46	2.33	0.64	0.64	0.76
	time(s)	0.61	0.58	0.64	0.44	0.89	0.38	2.85	5.94
blood	mean(%)	76.82•	77.59•	77.59•	77.09•	78.48	79.79	78.61	**79.89**
	±std(%)	1.54	1.29	1.13	1.39	1.74	1.14	1.44	1.24
	time(s)	2.64	0.60	0.70	4.14	0.71	0.37	3.54	5.77
breast	mean(%)	74.46•	75.90•	74.75•	73.24•	70.79•	77.70	76.76•	**78.56**
	±std(%)	2.94	1.86	1.26	2.42	2.85	2.43	1.25	2.15
	time(s)	0.14	0.12	0.17	0.14	0.14	0.09	0.71	1.08
cryotherapy	mean(%)	85.78	77.11•	86.00•	86.44	88.00	89.33	88.67	**90.22**
	±std(%)	4.99	16.27	3.73	5.01	3.87	3.27	5.11	4.12
	time(s)	0.05	0.05	0.10	0.06	0.13	0.05	0.17	0.52
customers	mean(%)	89.18•	88.64•	76.27•	89.86•	88.64•	92.18	91.55	**92.23**
	±std(%)	0.88	1.97	2.63	1.11	1.36	1.25	1.24	1.08
	time(s)	0.26	0.25	0.27	0.41	0.26	0.14	1.48	2.36
haberman	mean(%)	74.31	74.64	73.53•	72.94•	65.29	76.47	75.29	**77.06**
	±std(%)	3.09	3.17	3.25	3.55	22.07	3.78	4.12	3.55
	time(s)	0.16	0.14	0.15	0.20	0.19	0.09	0.76	1.28
heart	mean(%)	84.22	68.74•	82.96	84.30	80.44•	84.22	83.11	**84.52**
	±std(%)	2.25	10.17	2.37	2.31	2.07	1.45	3.05	1.12
	time(s)	0.13	0.13	0.14	0.13	0.22	0.08	0.72	1.06
liver	mean(%)	66.88•	61.56•	61.73•	63.99•	58.55•	70.98	61.73•	**71.27**
	±std(%)	2.65	2.82	3.53	4.43	2.94	2.52	3.95	2.88
	time(s)	0.17	0.25	0.17	0.17	0.08	0.10	1.09	1.41
pima	mean(%)	75.68•	71.51•	74.82•	76.33•	66.98•	77.84	77.45	**77.97**
	±std(%)	2.03	2.81	1.36	1.67	2.54	1.47	1.43	1.53
	time(s)	0.65	0.64	0.62	0.58	0.36	0.38	4.18	6.68
planning	mean(%)	71.10•	71.10•	71.21•	71.32•	71.10	71.10	71.10	**71.43**
	±std(%)	3.22	3.22	3.18	3.31	3.22	3.22	3.22	3.15
	time(s)	0.09	0.09	0.08	0.09	0.09	0.07	0.42	0.80
voting	mean(%)	96.54	91.49•	93.56•	95.85	96.32	96.78	96.31	**97.47**
	±std(%)	3.91	4.25	4.20	4.34	2.77	4.02	4.29	3.15
	time(s)	0.63	0.65	0.65	0.44	0.84	0.38	4.67	7.13
wpbc	mean(%)	77.98	77.58	76.06•	78.08	77.17	78.99	77.07	**79.80**
	±std(%)	2.51	2.55	2.39	2.67	2.13	3.02	3.23	3.29
	time(s)	0.09	0.09	0.09	0.10	0.11	0.06	0.34	0.81
diabetis	mean(%)	75.89•	68.05•	74.58•	76.77•	66.09•	78.52	78.10	**78.67**
	±std(%)	1.96	4.49	2.05	1.73	3.29	1.38	1.40	1.25
	time(s)	0.61	0.58	0.65	0.54	0.73	0.35	4.09	6.17

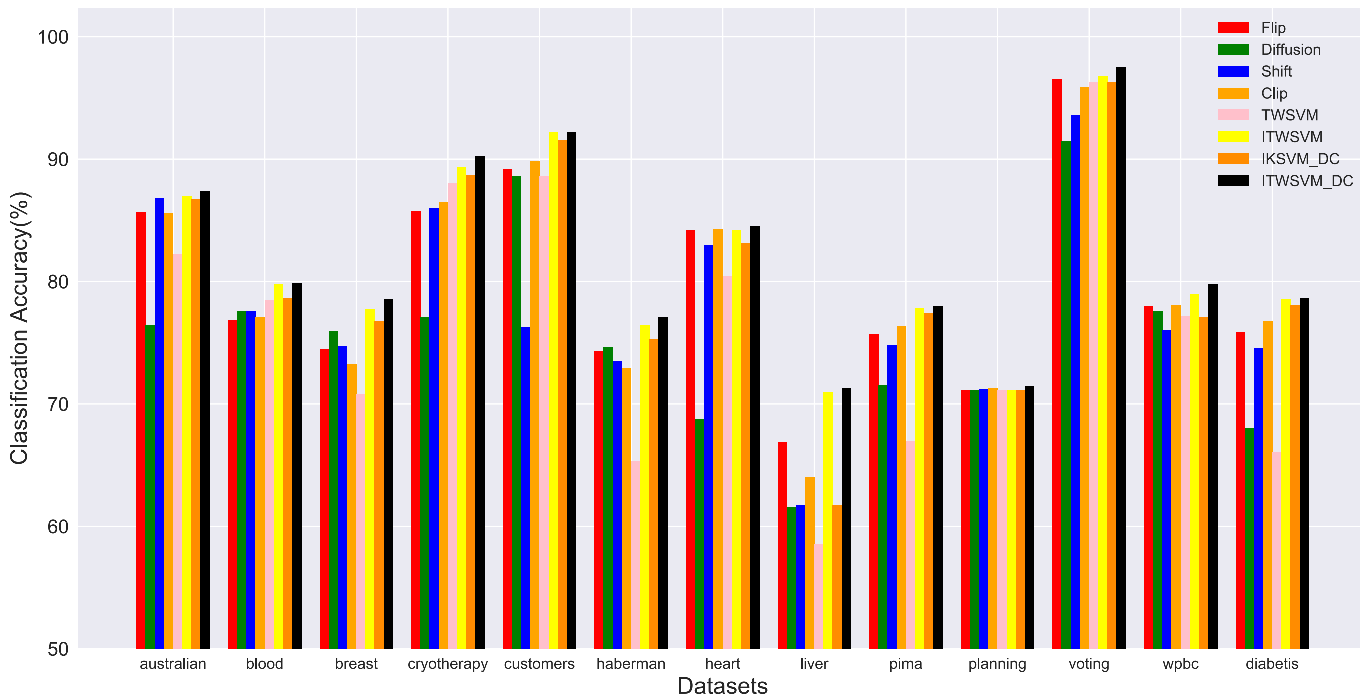


Fig. 1. The binary classification accuracy of various algorithms when using the sigmoid kernel.

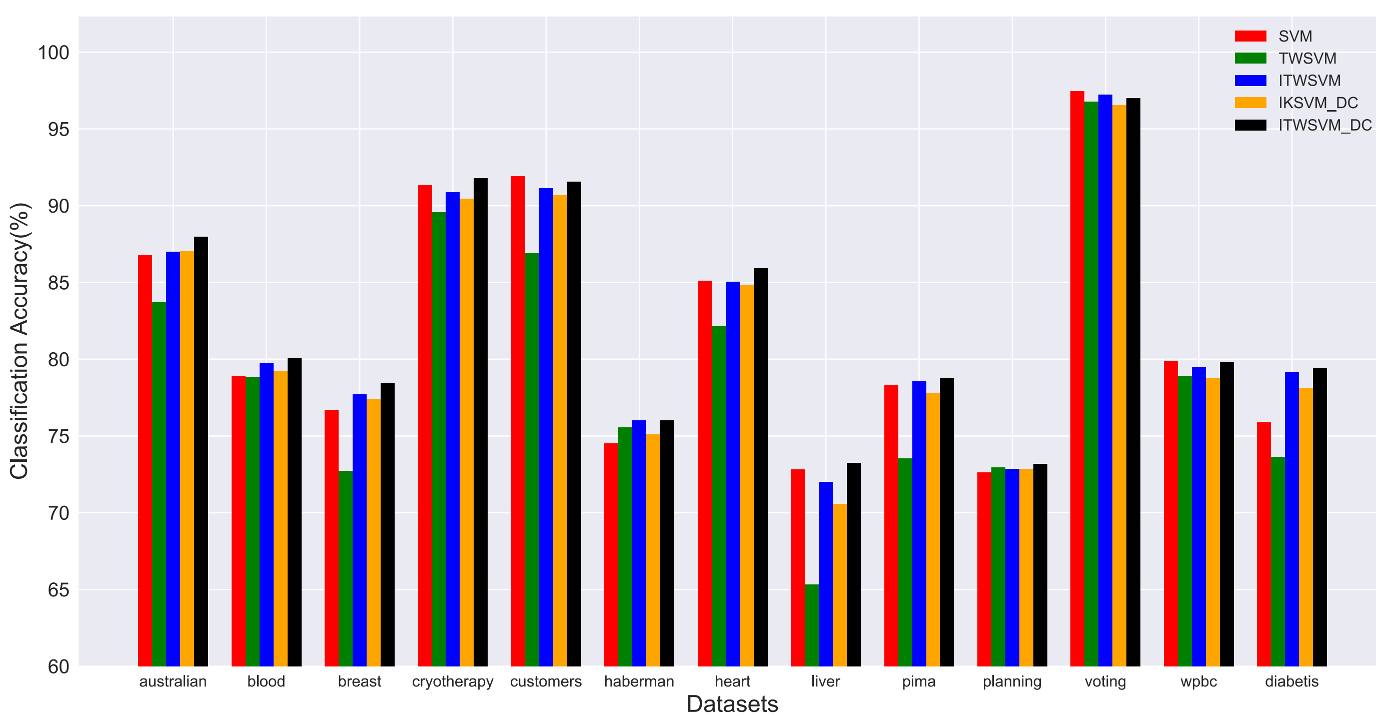


Fig. 2. The binary classification accuracy of various algorithms when using the RBF kernel.

compare our ITWSVM-DC with IKSVM-DC which is the state-of-the-art algorithm. To better illustrate performance of TWSVM with indefinite kernel, we compare our algorithm with the original TWSVM to demonstrate that directly using indefinite kernel is not favorable. For RBF kernel which is the prominent representative of PSD kernels, the kernel spectra of Flip, Shift, Clip and Diffusion do not need to transform and here we use original SVM as one comparison of our algorithm. We also compare our algorithm with the original TWSVM, ITWSVM and IKSVM-DC to test the robustness of our algorithm.

Tables 3 and 4 are the classification accuracies and training time of different algorithms when using the sigmoid kernel and RBF kernel respectively. The mean and standard deviation (std) of various algorithms are used to validate the accuracy of experimental results. Specially, when one algorithm is superior to all compared algorithms on one dataset, the accuracy of the algorithm is highlighted in bold. Furthermore, to statistically measure the performance differences of compared algorithms, we conduct pairwise t-test at 0.05 significance level between these algorithms. The maker •/◦ is shown when the ITWSVM-DC is statistically supe-

Table 4
The classification accuracy (mean ± standard deviation) and training time of binary classification of various algorithms when using the RBF kernel. •/◦ indicates whether the ITWSVM-DC is statistically superior/inferior to the compared models (pairwise t-test at 0.05 significance level).

Datasets		SVM	TWSVM	ITWSVM	IKSVM-DC	ITWSVM-DC
australian	mean(%)	86.78•	83.71•	86.99•	87.04•	**87.97**
	±std(%)	0.82	2.26	0.70	0.83	0.82
	time(s)	0.28	0.49	0.31	2.49	5.81
blood	mean(%)	78.88	78.85	79.73	79.20	**80.05**
	±std(%)	1.51	1.49	1.28	1.10	**1.33**
	time(s)	1.68	0.46	0.37	2.74	5.29
breast	mean(%)	76.69	72.73•	77.70	77.41	**78.42**
	±std(%)	2.07	2.15	2.30	1.58	**2.57**
	time(s)	0.06	0.11	0.08	0.60	1.08
cryotherapy	mean(%)	91.33	89.56	90.89	90.44	**91.78**
	±std(%)	3.06	2.64	3.51	4.10	3.98
	time(s)	0.03	0.06	0.05	0.22	0.42
customers	mean(%)	**91.91**	86.91•	91.14	90.68	91.55
	±std(%)	1.46	3.29	1.60	1.34	1.15
	time(s)	0.10	0.41	0.14	2.10	2.17
haberman	mean(%)	74.51	75.56	76.01	75.10	**76.01**
	±std(%)	3.46	3.07	3.74	3.82	3.73
	time(s)	0.30	0.16	0.09	0.63	1.31
heart	mean(%)	85.11	82.15•	85.04	84.81	**85.93**
	±std(%)	2.26	1.86	1.27	2.66	1.05
	time(s)	0.06	0.11	0.08	0.48	0.89
liver	mean(%)	72.83	65.32•	72.02	70.58•	**73.24**
	±std(%)	1.32	2.05	1.27	1.14	1.49
	time(s)	0.08	0.31	0.11	0.61	1.46
pima	mean(%)	78.31	73.54•	78.57	77.81	**78.75**
	±std(%)	1.45	1.68	1.54	1.81	1.37
	time(s)	0.27	0.71	0.36	3.52	6.13
planning	mean(%)	72.64	72.97	72.86	72.86	**73.19**
	±std(%)	3.20	2.92	3.22	3.03	2.96
	time(s)	0.04	0.09	0.06	0.33	0.82
voting	mean(%)	**97.46**	96.77	97.24	96.54	97.00
	±std(%)	3.18	3.47	3.07	3.90	3.28
	time(s)	0.27	0.41	0.35	3.19	5.91
wpbc	mean(%)	**79.90**	78.89	79.49	78.79	79.80
	±std(%)	2.98	3.42	3.50	3.38	3.26
	time(s)	0.05	0.08	0.06	0.40	0.80
diabetis	mean(%)	75.89•	73.65•	79.17	78.10•	**79.40**
	±std(%)	1.96	1.57	1.22	1.40	0.90
	time(s)	0.26	0.70	0.35	4.02	5.92

Table 6
The rank of various algorithms on binary classification datasets when using the RBF kernel.

Datasets	SVM	TWSVM	ITWSVM	IKSVM-DC	ITWSVM-DC
australian	4	5	3	2	**1**
blood	4	5	2	3	**1**
breast	4	5	2	3	**1**
cryotherapy	2	5	3	4	**1**
customers	**1**	5	3	4	2
haberman	5	3	**1**	4	**1**
heart	2	5	3	4	**1**
liver	2	5	3	4	**1**
pima	3	5	2	4	**1**
planning	5	2	3	3	**1**
voting	**1**	4	2	5	3
wpbc	**1**	4	3	5	2
diabetis	4	5	2	3	**1**
Avg.	2.9	4.5	2.5	3.7	**1.3**

rior/inferior to the compared algorithms. Otherwise, no maker is given.

From Tables 3 and 4, it is obvious that due to the introduction of the regularized term for ITWSVM which can be viewed as an implementation of the structural risk minimization principle, the classification performances of ITWSVM are significantly superior to the original TWSVM. From Table 3, when using the sigmoid kernel, the performance of TWSVM is not favorable in many datasets, which indicates that directly using indefinite kernel for TWSVM may lose useful information for non-convex problems. In these SVM methods for indefinite kernels (Flip, Diffusion, Shift, Clip, IKSVM-DC), the performance of IKSVM-DC algorithm is better than that of the IKSVMs which employ the methods of spectrum transformation in many cases, which means that the introduction of DC programming plays a significant role in solving non-convex problems and improves the performance of the model. It is worth noting that, in sigmoid kernel settings, our ITWSVM-DC outperforms all the algorithms on all binary classification datasets and is statistically significantly superior to compared algorithms in most cases, which indicates that the proposed ITWSVM-DC algorithm is effective and can significantly improve the classification accuracy of the algorithm when using indefinite kernels. It means that ITWSVM-DC can not only make full use of the advantages of TWSVM and hold the structural risk minimization in SVM but also effectively apply DC algorithm to solve non-convex problems caused by indefinite kernels. Therefore, our algorithm can always achieve the best result and successfully employ indefinite kernels to TWSVM. From Table 4, in RBF kernel settings, our proposed ITWSVM still outperforms the original TWSVM. The performance of IKSVM-DC is not particularly favorable and stable while our ITWSVM-DC performs robustly and achieves the highest average accuracy for binary classification datasets. The results demonstrate that our method performs outstandingly in terms of PSD kernels and indefinite kernels.

In order to show the classification effect of each algorithm more clearly, Figs. 1 and 2 show the performances of compared algo-

Table 5
The rank of various algorithms on binary classification datasets when using the sigmoid kernel.

Datasets	Flip	Diffusion	Shift	Clip	TWSVM	ITWSVM	IKSVM-DC	ITWSVM-DC
australian	5	8	3	6	7	2	4	**1**
blood	8	5	5	7	4	2	3	**1**
breast	6	4	5	7	8	2	3	**1**
cryotherapy	7	8	6	5	4	2	3	**1**
customers	5	6	8	4	6	2	3	**1**
haberman	5	4	6	7	8	2	3	**1**
heart	3	8	6	2	7	3	5	**1**
liver	3	7	5	4	8	2	5	**1**
pima	5	7	6	4	8	2	3	**1**
planning	6	6	3	2	6	6	6	**1**
voting	3	8	7	6	4	2	5	**1**
wpbc	4	5	8	3	6	2	7	**1**
diabetis	5	7	6	4	8	2	3	**1**
Avg.	5.0	6.4	5.7	4.7	6.5	2.4	4.1	**1.0**

Table 7
The classification accuracy (mean ± standard deviation) and training time of multi-class classification of various algorithms when using the sigmoid kernel. •/◦ indicates whether the ITWSVM-DC is statistically superior/inferior to the compared models (pairwise *t*-test at 0.05 significance level).

Datasets		Flip	Diffusion	Shift	Clip	TWSVM	ITWSVM	IKSVM-DC	ITWSVM-DC
soybean	mean(%)	98.70•	91.30	99.57	98.70	99.57	99.57	99.57	**99.57**
	±std(%)	1.99	9.91	1.30	1.99	1.30	1.30	1.30	1.30
	time(s)	0.08	0.08	0.09	0.10	0.10	0.11	0.58	0.82
breast-tissue	mean(%)	40.75•	40.94•	46.79•	37.17•	60.00	61.70	60.57	**61.32**
	±std(%)	4.40	9.99	7.54	10.23	5.46	7.55	5.56	5.08
	time(s)	0.17	0.18	0.18	0.21	0.17	0.17	1.77	1.80
iris	mean(%)	65.60•	70.40•	86.53•	66.13•	94.27	96.40	95.20	**96.40**
	±std(%)	7.00	4.29	4.15	4.59	2.39	2.15	2.25	2.07
	time(s)	0.13	0.13	0.12	0.14	0.13	0.18	1.13	1.36
wine	mean(%)	91.35	68.99•	97.30	89.10•	94.61•	96.07	95.73	**96.74**
	±std(%)	16.65	10.47	1.25	6.95	2.06	2.20	2.35	1.37
	time(s)	0.12	0.13	0.12	0.14	0.12	0.11	0.94	1.26
seeds	mean(%)	70.95•	47.52•	83.62•	70.19•	90.29	90.76	90.76	**90.95**
	±std(%)	9.43	18.44	5.50	7.92	2.12	0.86	1.05	1.06
	time(s)	0.14	0.14	0.14	0.18	0.17	0.22	1.26	1.30
glass	mean(%)	49.81•	48.69•	52.90•	46.92•	61.40	60.84	58.69	**63.46**
	±std(%)	3.37	5.21	5.64	4.34	3.34	4.15	6.13	3.75
	time(s)	0.29	0.32	0.29	0.51	0.37	0.34	2.74	2.91
balance	mean(%)	86.71•	87.12•	86.87•	86.93•	87.00•	**93.67**	91.92	92.59
	±std(%)	1.21	1.16	1.34	1.15	0.72	1.43	1.18	1.18
	time(s)	0.64	0.77	0.63	0.70	0.76	0.57	6.49	7.23

Table 8
The classification accuracy (mean ± standard deviation) and training time of multi-class classification of various algorithms when using the RBF kernel. •/◦ indicates whether the ITWSVM-DC is statistically superior/inferior to the compared models (pairwise *t*-test at 0.05 significance level).

Datasets		SVM	TWSVM	ITWSVM	IKSVM-DC	ITWSVM-DC
soybean	mean(%)	99.57	99.57	99.57	99.57	**100.00**
	±std(%)	1.30	1.30	1.30	1.30	0.00
	time(s)	0.09	0.08	0.09	0.43	0.61
breast-tissue	mean(%)	61.13	59.81	**62.45**	60.94	62.08
	±std(%)	7.41	6.48	5.69	5.91	4.42
	time(s)	0.18	0.18	0.16	1.47	1.84
iris	mean(%)	96.13	95.73	96.27	96.13	**96.40**
	±std(%)	1.83	1.31	1.87	2.19	1.98
	time(s)	0.12	0.11	0.10	1.06	1.13
wine	mean(%)	97.98	97.19•	98.31	98.42	**98.54**
	±std(%)	0.98	1.26	0.91	1.03	0.88
	time(s)	0.14	0.12	0.13	1.01	1.54
seeds	mean(%)	**94.00**	93.05	92.00	92.38	92.95
	±std(%)	1.05	1.35	0.97	1.59	1.36
	time(s)	0.14	0.11	0.11	1.25	1.68
glass	mean(%)	**68.69**	66.17	67.48	68.41	68.31
	±std(%)	3.60	3.07	3.49	4.71	4.35
	time(s)	0.27	0.25	0.23	3.01	3.23
balance	mean(%)	91.21•	88.59•	**93.74**	92.84	93.19
	±std(%)	1.33	1.40	1.55	1.27	1.27
	time(s)	0.59	0.69	0.45	7.23	7.42

rithms on different datasets with sigmoid kernel and RBF kernel respectively.

For better illustrating the results of experiments, we use statistical comparisons of classifiers-Friedman test. The null-hypothesis is that all the algorithms perform the same and the observed differences are merely random. The test results of each algorithm on each dataset are obtained and can be sorted from good to bad. If the test performances of the algorithms are the same, the score order value is the same. Tables 5 and 6 show the ranks of the algorithms in this paper.

The Friedman statistic is as follow:

$$\chi_F^2 = \frac{12N}{k(k-1)}\left[\sum_j \left(\frac{1}{N}\sum_i r_i^j\right)^2 - \frac{k(k+1)^2}{4}\right]. \tag{62}$$

We compare these k algorithms on N datasets and r_i^j is the rank of the ith of N datasets and the jth of k algorithms. In this section, N is noted as 13 and k are 8 and 5 in sigmoid kernel and RBF kernel settings respectively. The Friedman statistic is distributed according to χ_F^2 with $k-1$ degrees of freedom. The original Friedman test is too conservative, and now we usually use

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}. \tag{63}$$

where χ_F^2 can be attained from Eq. (62). F_F is distributed according to F-distribution with $k-1$ and $(k-1)(N-1)$ degrees of freedom. When the significance level is 0.05, according to Eq. (63), the value of F_F of these classifiers is 20.5099 when using the sigmoid kernel, which is bigger than the critical values of the F-test 2.1206. For the RBF kernel, the value of F_F is 15.4239, which is also bigger than the

Table 9
The rank of various algorithms on multi-class classification datasets when using the sigmoid kernel.

Datasets	Flip	Diffusion	Shift	Clip	TWSVM	ITWSVM	IKSVM-DC	ITWSVM-DC
soybean	6	8	**1**	6	**1**	**1**	**1**	**1**
breast-tissue	7	6	5	8	4	**1**	3	2
iris	8	6	5	7	4	**1**	3	**1**
wine	6	8	**1**	7	5	3	4	2
seeds	6	8	5	7	4	2	2	**1**
glass	6	7	5	8	2	3	4	**1**
balance	8	4	7	6	5	**1**	3	2
Avg.	6.7	6.7	4.1	7.0	3.6	1.7	2.9	**1.4**

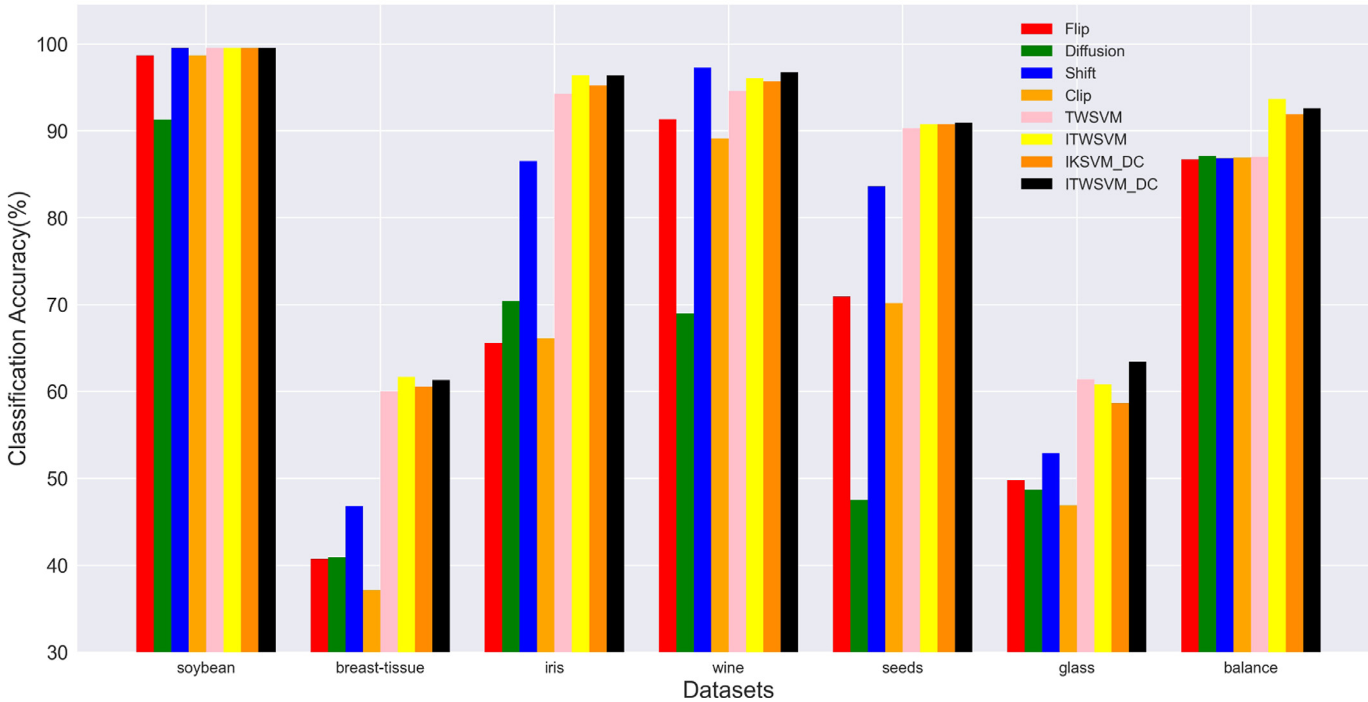


Fig. 3. The multi-class classification accuracy of various algorithms when using the sigmoid kernel.

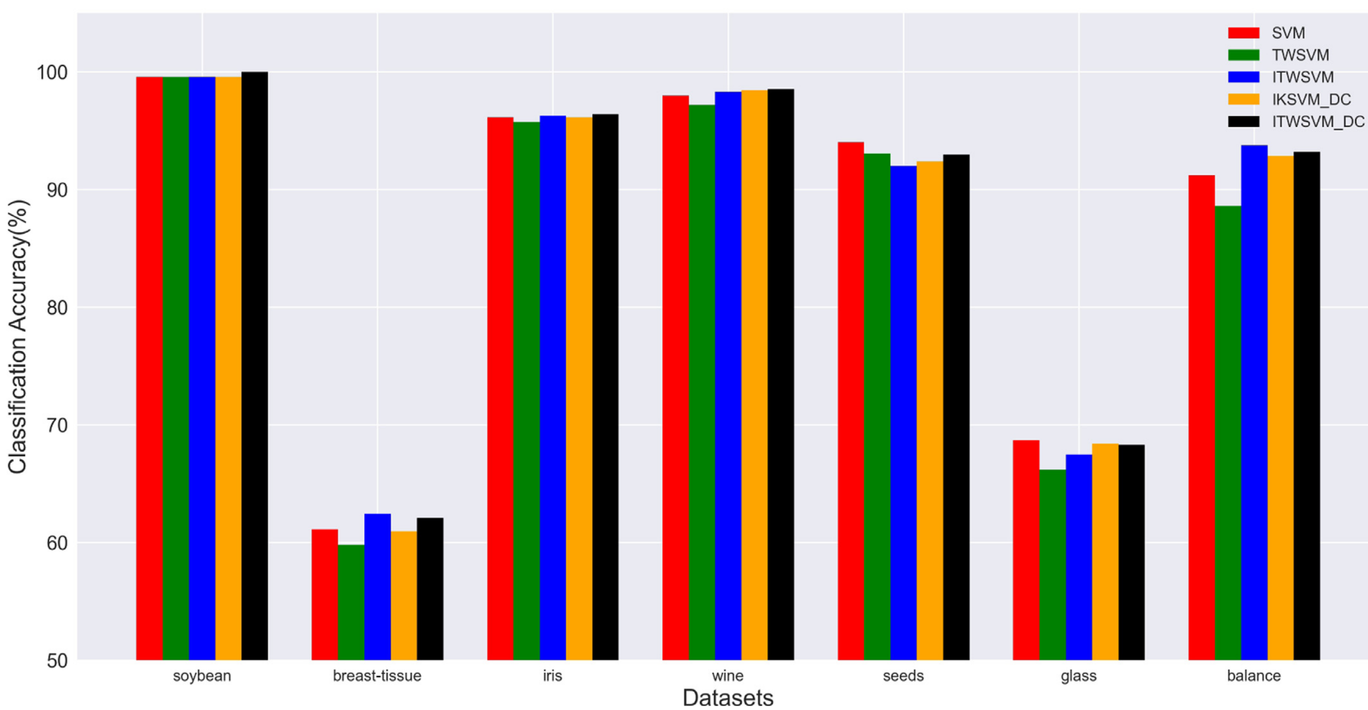
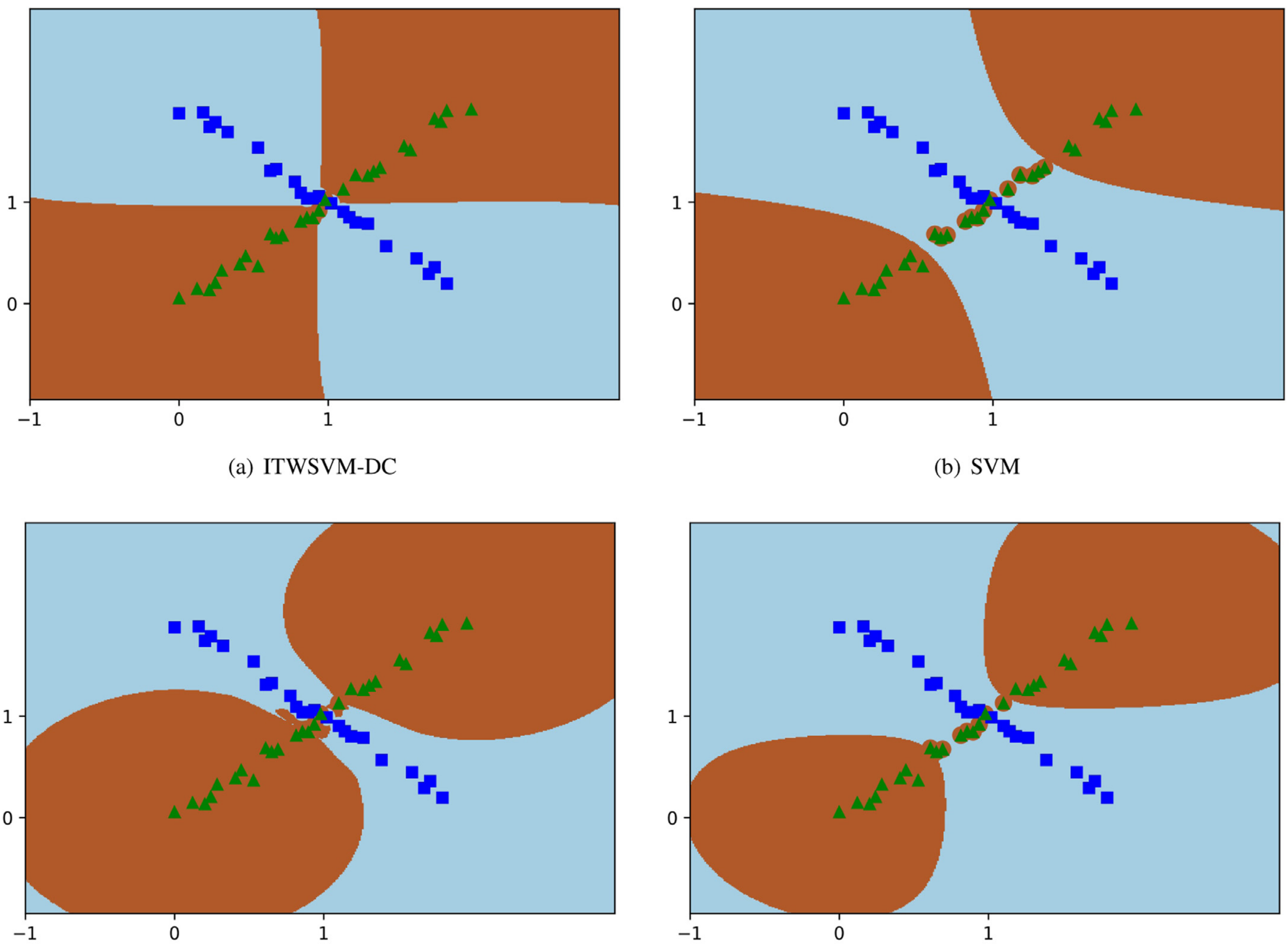


Fig. 4. The multi-class classification accuracy of various algorithms when using the RBF kernel.

critical values of the F-test 2.5652. Therefore, the null-hypothesis is rejected, which means that the performances of these algorithms are different.

4.3. Experimental results on multi-class classification datasets

In this section, we perform experiments on multi-class classification datasets. Tables 7 and 8 are the classification accuracies and training time of different algorithms when using the sigmoid kernel and RBF kernel respectively. From Tables 7 and 8, it is obviously that the classification performances of ITWSVM are significantly superior to the original TWSVM. From Table 7, in sigmoid kernel settings, our ITWSVM-DC almost outperforms all the algorithms on all datasets and is statistically significantly superior to compared algorithms in most cases, which indicates that the proposed ITWSVM-DC algorithm is effective and can significantly improve the classification accuracy of the algorithm when using the sigmoid kernel. Therefore, our algorithm can successfully employ indefinite kernels to TWSVM and always achieve the best result with indefinite kernels in multi-class classification setting. From



(a) ITWSVM-DC (b) SVM

(c) TWSVM (d) IKSVM-DC

Fig. 5. Comparisons of the decision boundary of different methods on the artificial dataset.

Tables 8, in RBF kernel settings, our ITWSVM-DC still achieves the highest average accuracy for multi-class classification datasets. The results demonstrate that our method performs outstandingly in terms of PSD kernels and indefinite kernels no matter in binary classification settings or multi-class classification settings. Therefore, ITWSVM-DC is a robust and prominent algorithm and can excellently deal with problems in different situations.

In order to show the classification effect of each algorithm more clearly, Figs. 3 and 4 show the performances of compared algorithms on different datasets with sigmoid kernel and RBF kernel respectively.

To statistically measure the significance of performance difference, Friedman test at 0.05 significance level is conducted on all datasets. The null-hypothesis is that all the algorithms perform the same and the observed differences are merely random. The test results of each algorithm on each dataset are obtained and can be sorted from good to bad. Tables 9 and 10 show the ranks of the algorithms with sigmoid kernel and RBF kernel in the multi-class classification settings respectively. When using the sigmoid kernel, the value of F_F of these classifiers is 18.5487, which is bigger than the critical values of the F-test 2.2371. For the RBF kernel, the value of F_F is 2.8688, which is also bigger than the critical values of the F-test 2.7763. Therefore, the null-hypothesis is rejected, which means that the performances of these algorithms are different.

Table 10
The rank of various algorithms on multi-class classification datasets when using the RBF kernel.

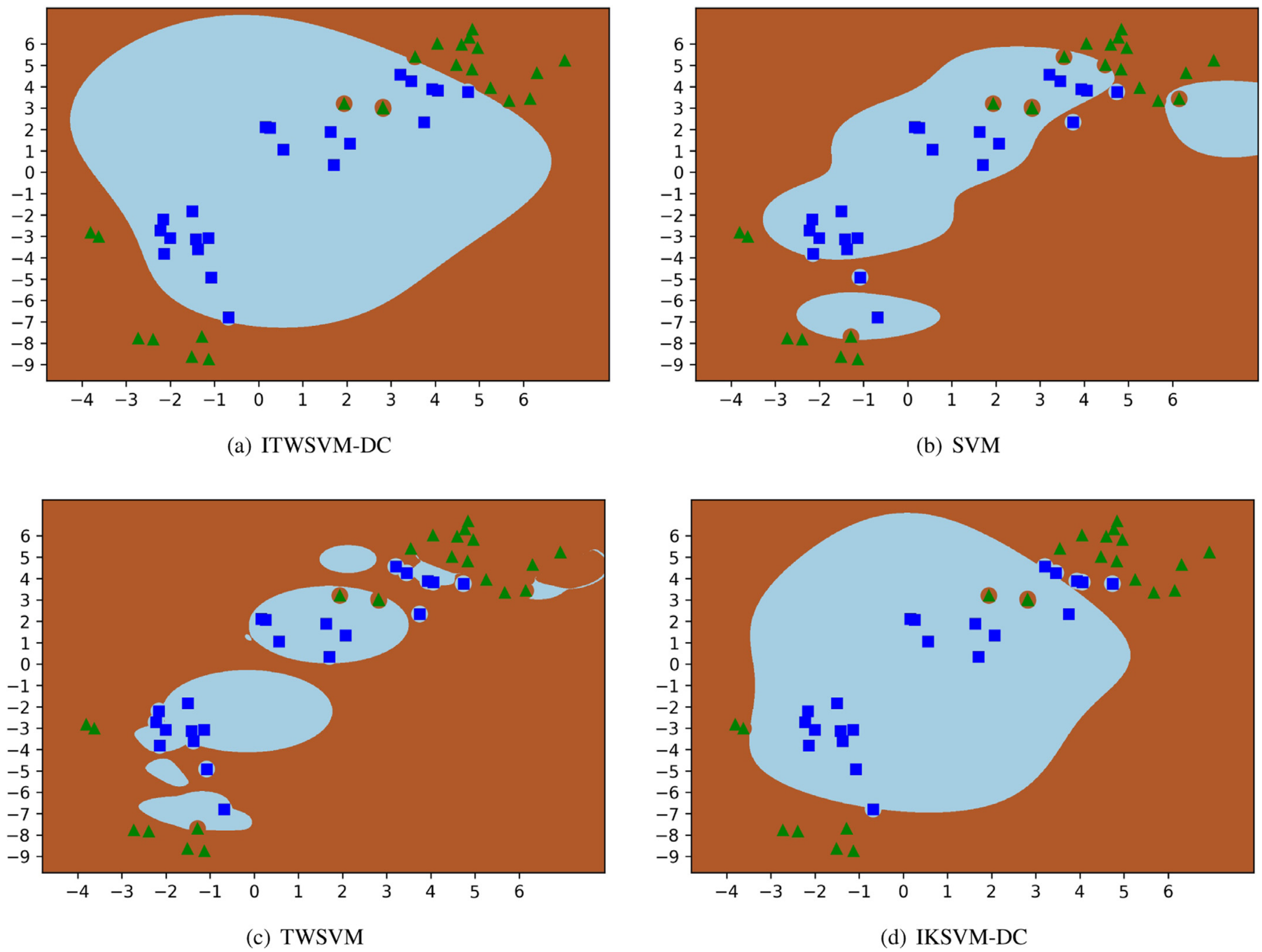
Datasets	SVM	TWSVM	ITWSVM	IKSVM-DC	ITWSVM-DC
soybean	2	2	2	2	**1**
breast-tissue	3	5	**1**	4	2
iris	3	5	2	3	**1**
wine	4	5	3	2	**1**
seeds	1	2	5	4	3
glass	1	5	4	2	3
balance	4	5	1	3	2
Avg.	2.6	4.1	2.6	2.9	**1.9**

4.4. Experimental results with different indefinite kernels

Finally, we compare the performance of ITWSVM-DC with different indefinite kernels. Three indefinite kernels are selected for comparison [37].

- Gaussian combination kernel:

$$K(x,z) = \exp\left(-\gamma_1\|x-z\|^2\right) + \exp\left(-\gamma_2\|x-z\|^2\right)$$



(a) ITWSVM-DC (b) SVM

(c) TWSVM (d) IKSVM-DC

Fig. 6. Comparisons of the decision boundary of different methods on the cryotherapy dataset.

$$+\exp\left(-\gamma_3\|x-z\|^2\right), \tag{64}$$

- Multiquadric kernel:

$$K(x,z)=\sqrt{\gamma\|x-z\|^2+c^2}, \tag{65}$$

- Thin plate spline kernel:

$$K(x,z)=\gamma\|x-z\|^{2p}\ln\left(\gamma\|x-z\|^2\right), \tag{66}$$

The kernel parameters in these indefinite kernels are selected by grid search from the set $\{2^{-6},2^{-5},\cdots,2^{6}\}$. Table 11 illustrates the classification accuracy of ITWSVM-DC with different indefinite kernels on thirteen binary classification datasets. From Table 11, we can demonstrate that there is no certain indefinite kernel function which is superior to others in all cases. Experiments show that it is necessary for us to select the appropriate kernel function for ITWSVM-DC to achieve optimal performance according to specific problems.

4.5. Decision boundary and convergence

We conduct the comparisons of the decision boundaries of SVM, TWSVM, IKSVM-DC and ITWSVM-DC on artificial and real-world datasets. The artificial dataset is produced by two cross lines with Gaussian noise, which has zero-mean and the variance of 0.05. The cryotherapy dataset is a real-world dataset. The t-SNE [38] method is used for visualizing the decision boundaries. Figs. 5 and 6 illustrate the decision boundaries of different methods on artificial and real-world datasets respectively with RBF kernel. From Figs. 5 and 6, we can find that compared with other algo-

Table 11
The binary classification accuracy (mean ± standard deviation) and training time of ITWSVM-DC with various kernels.

Datasets		Gaussian combination	Multiquadric	Thin plate spline
australian	mean(%)	87.01	**87.16**	86.32
	±std(%)	0.74	0.67	0.71
	time(s)	4.94	5.31	5.86
blood	mean(%)	77.70	**78.64**	77.73
	±std(%)	1.26	1.41	1.61
	time(s)	5.11	5.40	14.37
breast	mean(%)	76.33	**76.62**	75.97
	±std(%)	1.72	2.64	1.88
	time(s)	1.01	1.11	1.23
cryotherapy	mean(%)	88.67	86.00	**90.89**
	±std(%)	4.15	4.45	3.64
	time(s)	0.48	0.50	1.10
customers	mean(%)	89.14	86.68	**90.23**
	±std(%)	2.61	1.61	1.73
	time(s)	2.03	2.26	3.49

(continued on next page)

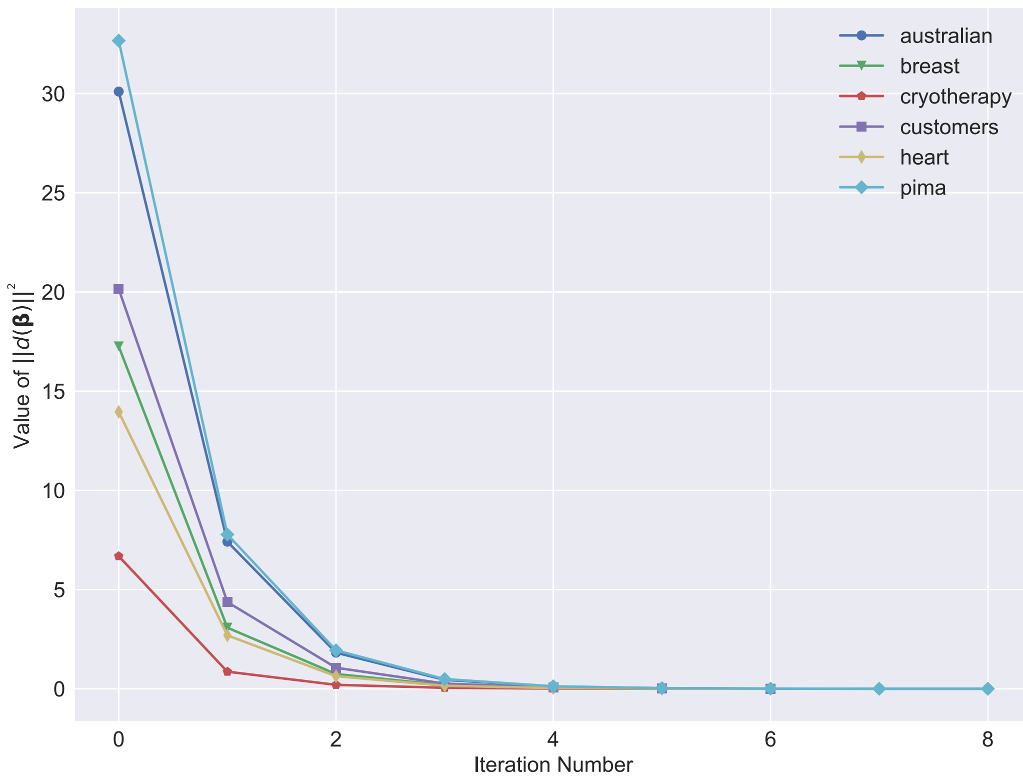


Fig. 7. The convergence of ITWSVM-DC on 6 datasets.

Table 11 (*continued*)

Datasets		Gaussian combination	Multiquadric	Thin plate spline
haberman	mean(%)	73.59	**74.12**	**74.12**
	±std(%)	3.57	3.79	4.53
	time(s)	1.17	1.20	1.56
heart	mean(%)	**84.52**	83.70	84.30
	±std(%)	1.38	1.55	1.93
	time(s)	1.02	1.02	1.30
liver	mean(%)	64.86	65.55	**68.96**
	±std(%)	5.71	2.57	3.54
	time(s)	1.32	1.49	3.17
pima	mean(%)	76.64	76.93	**77.73**
	±std(%)	1.43	2.00	1.28
	time(s)	5.41	6.17	7.97
planning	mean(%)	71.98	**72.97**	71.43
	±std(%)	3.08	3.04	2.99
	time(s)	0.75	0.76	0.88
voting	mean(%)	94.93	94.70	**95.62**
	±std(%)	3.96	3.42	4.45
	time(s)	5.84	6.45	10.51
wpbc	mean(%)	76.97	76.57	**77.98**
	±std(%)	2.63	2.63	2.38
	time(s)	0.76	0.70	1.03
diabetis	mean(%)	76.61	**78.41**	77.47
	±std(%)	1.89	1.10	1.79
	time(s)	5.07	5.90	6.64

rithms, ITWSVM-DC can generate more reasonable decision boundaries and distinguish instances of different classes better.

In order to better illustrate the convergence of our algorithm, we design experiments to verify it. The experimental results on 6 datasets (australian, breast, cryotherapy, customers, heart, pima) is shown in Fig. 7. In Fig. 7, $\|d(\boldsymbol{\beta})\|^2 = \|d(\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t)\|^2$ is the value of the solution sequence $\boldsymbol{\beta}_t$ during the iterations. From Fig. 7, it is obviously that the value $\left\|d(\boldsymbol{\beta})\right\|^2$ gradually converges in few iterations on the 6 datasets.

5. Conclusions

In this paper, we propose a new algorithm named indefinite twin support vector machine with difference of convex functions programming (ITWSVM-DC) which is the first time to employ indefinite kernel to TWSVM. We directly focus on the primal problem of TWSVM instead of the dual form of TWSVM to avoid the existence of dual gap and the loss caused by dual form. By modifying the objective function, a new regularized TWSVM (ITWSVM) comes into being which can improve the generalization of TWSVM. By using the Representer Theorem in RKKS, we reconstruct the ITWSVM and provide theoretical support for the indefinite TWSVM. After analyzing the convexity of the proposed ITWSVM, DC programming is introduced to solve the non-convex problem. A line search along the descent direction at each iteration is adopted to find the solution. Furthermore, experiments with sigmoid kernel have been performed to prove the superiority of our algorithm with indefinite kernels. Radial Basis Function kernel is also applied to demonstrate the robustness of our algorithm. Extensive experiments demonstrate that ITWSVM-DC is a robust and prominent algorithm and can perform excellently in different situations.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Grant No.62076062) and National Key R&D Program of China (Grant No.2017YFB1002801). Furthermore, the work was also supported by Collaborative Innovation Center of Wireless Communications Technology.

References

[1] C. Cores, V.N. Vapnik, Support vector networks, Mach. Learn. 20 (2) (1995) 273–297.
[2] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1996.
[3] A. Torres-Barrán, C. Alaíz, J. Dorronsoro, Faster SVM training via conjugate SMO, Pattern Recognit. 111 (2021) 107644.
[4] C. da Silva Santos, R. Sampaio, L. dos Santos Coelho, G. Bestard, C. Llanos, Multi-objective adaptive differential evolution for SVM/SVR hyperparameters selection, Pattern Recognit. 110 (2021) 107649.
[5] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, Cambrige, 2000.
[6] J. Xu, W. Zeng, Y.Y. Lan, J.F. Guo, X.Q. Cheng, Modeling the parameter interactions in ranking SVM with low-rank approximation, IEEE Trans. Knowl. Data Eng. 31 (6) (2019) 1181–1193.
[7] S.L. Peng, Q.H. Hu, Y.L. Chen, J.W. Dang, Improved support vector machine algorithm for heterogeneous data., Pattern Recognit. 48 (6) (2015) 2072–2083.
[8] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, B.W. Xi, Adversarial support vector machine learning, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 1059–1067.
[9] X. Miao, Y. Liu, H. Zhao, C. Li, Distributed online one-class support vector machine for anomaly detection over networks., IEEE Trans. Cybern. 49 (4) (2019) 1475–1488.
[10] U. Sharif, Z. Mehmood, T. Mahmood, M.A. Javid, A. Rehman, T. Saba, Scene analysis and search using local features and support vector machine for effective content-based image retrieval, Artif. Intell. Rev. 52 (2) (2019) 901–925.
[11] G. Taherzadeh, Y.D. Yang, T. Zhang, A.W.C. Liew, Y.Q. Zhou, Sequence-based prediction of protein-peptide binding sites using support vector machine, J. Comput. Chem. 37 (13) (2016) 1223–1229.
[12] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach Intell. 29(5) (2007) 905–910.
[13] Y. Shao, C.H. Zhang, X.B. Wang, N.Y. Deng, Improvements on twin support vector machines, IEEE Trans. Neural Netw. 22 (6) (2011) 962–968.
[14] Y.J. Tian, Z.Q. Qi, X.C. Ju, Y. Shi, X.H. Liu, Nonparallel support vector machines for pattern classification, IEEE Trans. Cybern. 44 (7) (2014) 1067–1079.
[15] G. Loosli, S. Canu, C.S. Ong, Learning SVM in Kreïn spaces, IEEE Trans. Pattern Anal. Mach Intell. 38 (6) (2015) 1204–1216.
[16] A.J. Smola, Z.L. Ovari, R.C. Williamson, Regularization with dot-product kernels, in: Proceedings of Advances in Neural Information Processing Systems, 2000, pp. 308–314.
[17] Y.H. Chen, M.R. Gupta, B. Recht, Learning kernels from indefinite similarities, in: Proceedings of 26th International Conference On Machine Learning, 2009, pp. 145–152.
[18] E. Pekalska, P. Paclik, R.P.W. Duin, A generalized kernel approach to dissimilarity-based classification, J. Mach. Learn. Res. 2 (2002) 175–211.
[19] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, K. Obermayer, Classification on pairwise proximity data, in: Proceedings of Advances in Neural Information Processing Systems, 1999, pp. 438–444.
[20] V. Roth, J. Laub, M. Kawanabe, J. Buhmann, Optimal cluster preserving embedding of nonmetric proximity data, IEEE Trans. Pattern Anal. Mach Intell. 25 (12) (2003) 1540–1551.
[21] D. Oglic, T. Gärtner, Learning in reproducing kernel Kreïn spaces, in: Proceedings of 35th International Conference on Machine Learning, 2018, pp. 6188–6200.
[22] J. Laub, K.R. Müller, Feature discovery in non-metric pairwise data, J. Mach. Learn. Res. 5 (2004) 801–818.
[23] H.M. Xu, H. Xue, X.H. Chen, Y.Y. Wang, Solving indefinite kernel support vector machine with difference of convex functions programming, in: Proceedings of 31st AAAI Conference on Artificial Intelligence, 2017, pp. 2782–2788.
[24] H.A. Le Thi, P.D. Tao, The DC (Difference of convex functions) programming and DCA revisited with dc models of real world nonconvex optimization problems, Ann. Oper. Res. 133 (1) (2005) 23–46.
[25] P.D. Tao, H.A. Le Thi, Convex analysis approach to dc programming: theory, algorithms and applications, Acta Math. Vietnam. 22 (1) (1997) 289–355.
[26] S. Boyd, L. Xiao, A. Mutapcic, Subgradient Methods, Lecture Notes of EE392o, Stanford University, Autumn Quarter, 2003.
[27] Y.H. Shao, N.Y. Deng, A coordinate descent margin based-twin support vector machine for classification, Neural Netw. 25 (2012) 114–121.
[28] S. Bernhard, S. Alexander, Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond, MIT Press, Cambrige, 2002.
[29] C.S. Ong, X. Mary, S. Canu, A.J. Smola, Learning with non-positive kernels, in: Proceedings of 25th International Conference on Machine Learning, 2004, pp. 639–646.
[30] B. Stephen, V. Lieven, Convex Optimization, Cambridge University Press, Cambrige, 2004.
[31] R. Bot, Conjugate Duality in Convex Optimization, Springer Science & Business Media, 2009.
[32] H. Bauschke, P. Combettes, Fenchel–rockafellar duality, in: Convex Analysis and Monotone Operator Theory in Hilbert Spaces, Springer, 2017, pp. 247–262.
[33] P.D. Tao, H.A. Le Thi, Recent advances in DC programming and DCA, Trans. Comput. Collect. Intel. XIII. 8342 (2014) 1–37.
[34] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course, Springer Science and Business Media, Dordrecht, 2013.
[35] J. Xie, K. Hone, W. Xie, X. Gao, Y. Shi, X. Liu, Extending twin support vector machine classifier for multi-category classification problems, Intell. Data Anal. 17 (4) (2013) 649–664.
[36] R.I. Kondor, J.D. Lafferty, Diffusion kernels on graphs and other discrete input spaces, in: Proceedings of the 9th International Conference on Machine Learning, 2002, pp. 315–322.
[37] C. Ong, X. Mary, S. Canu, A. Smola, Learning with non-positive kernels, vol. 69, 2004.
[38] L. van der Maaten, G. Hinton, Visualizing high-dimensional data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.

Yuexuan An, received her B.Sc in computer science and technology from Jiangsu Normal University in 2015 and M.Sc. degree in computer application technology in China University of Mining and Technology in 2019. She is currently pursuing the Ph.D. degree in the School of computer science and engineering, Southeast University. Her research interest includes machine learning, pattern recognition, SVM, kernel function and various applications.

Hui Xue, received her B.Sc in mathematics from Nanjing Normal University in 2002, and M.Sc. in mathematics from Nanjing University of Aeronautics & Astronautics (NUAA) in 2005. In 2008, she received her Ph.D. degree in computer science from NUAA. She is a professor at the PALM Group, School of Computer Science and Engineering, Southeast University.