Front. Comput. Sci., 2020, 14(2): 349–363
https://doi.org/10.1007/s11704-018-8148-z

RESEARCH ARTICLE

A primal perspective for indefinite kernel SVM problem

Hui XUE (✉)[1,2], Haiming XU[1,2], Xiaohong CHEN[3], Yunyun WANG[4]

1 School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
2 Key Laboratory of Computer Network and Information Integration (Southeast University), MOE, Nanjing 210096, China
3 College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
4 School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210046, China

Abstract Indefinite kernel support vector machine (IKSVM) has recently attracted increasing attentions in machine learning. Since IKSVM essentially is a non-convex problem, existing algorithms either change the spectrum of indefinite kernel directly but risking losing some valuable information or solve the dual form of IKSVM whereas suffering from a dual gap problem. In this paper, we propose a primal perspective for solving the problem. That is, we directly focus on the primal form of IKSVM and present a novel algorithm termed as IKSVM-DC for binary and multi-class classification. Concretely, according to the characteristics of the spectrum for the indefinite kernel matrix, IKSVM-DC decomposes the primal function into the subtraction of two convex functions as a difference of convex functions (DC) programming. To accelerate convergence rate, IKSVM-DC combines the classical DC algorithm with a line search step along the descent direction at each iteration. Furthermore, we construct a multi-class IKSVM model which can classify multiple classes in a unified form. A theoretical analysis is then presented to validate that IKSVM-DC can converge to a local minimum. Finally, we conduct experiments on both binary and multi-class datasets and the experimental results show that IKSVM-DC is superior to other state-of-the-art IKSVM algorithms.

Keywords indefinite kernel, support vector machine, multi-class classification, non-convex optimization

Received April 18, 2018; accepted November 21, 2018

E-mail: hxue@seu.edu.cn

1 Introduction

Support vector machines (SVM) [1] with kernels have been successfully used in many application areas. In traditional SVMs, the kernels embed samples into a high-dimensional (possibly infinite-dimensional) feature space for linear separation, where the corresponding kernel matrix is required to be symmetric and positive semi-definite (PSD) [2–4]. The PSD property guarantees that the problem can be formulated as a convex quadratic programming and yields a global optimum. However, in practice, many real-world applications directly utilize similarity measures as the kernels, most of which are indefinite rather than PSD. For example, Smith-Waterman and BLAST scores for evaluating pair-wise similarity between protein sequences usually generate indefinite kernel matrices [5]. The weighted meta-path based similarity matrices for text classification in natural language processing are frequently indefinite [6]. The sigmoid kernels in neural networks with various values of the hyper-parameters are also mostly indefinite [7]. As a result, indefinite kernels have become increasingly important in kernel methods [8–12] and indefinite kernel SVM (IKSVM) has attracted more and more attentions in machine learning [13–15]. However, different from the traditional SVMs, IKSVM boils down to a non-convex optimization which is an NP-hard problem.

In the past few years, many algorithms have been proposed to address the IKSVM problem. They generally fall into two categories: (1) "Kernel Transformation" which transforms the indefinite kernel matrix to be PSD and (2) "Non-convex

Optimization" which solves the non-convex problem directly. In the first category, some algorithms directly transform the eigenspectrum of the kernel matrix. For example, "Clip" neglects the negative eigenvalues [16], "Flip" flips the sign of the negative eigenvalues [17], and "Shift" shifts all the eigenvalues by a positive constant [18]. Other algorithms further consider the indefinite kernel as a noisy observation of some unknown PSD kernel. Luss and d'Aspremont presented a joint optimization on the dual model of SVM with an additional regularization term which measures the similarity between the proxy and the original indefinite kernel matrices [19]. Chen and Ye reformulated the formulation into a semi-infinite quadratically constrained linear programming and proposed a faster algorithm [20]. Chen et al. further introduced a primal model to avoid over-fitting [21]. Gu and Guo incorporated the kernel principal component analysis into the SVM classification and naturally generated a surrogate PSD kernel [22]. However, these methods actually change the indefinite kernels themselves and thus may lead to the loss of some important information involved in the kernels.

In the second category, most algorithms aim to solve the non-convex dual form of IKSVM. For example, Lin and Lin proposed an SMO-type method to solve the non-convex dual formulation of IKSVM which can converge to some stationary points for the non-PSD sigmoid kernel [23]. Akoa incorporated difference of convex functions programming into decomposition methods to tackle the dual problem and obtained a stationary point as a solution [24]. Ong et al. extended IKSVM into a reproducing kernel kreĭn space (RKKS), in which they stabilized the primal IKSVM model and reformulated it as a dual optimization problem by transforming the indefinite kernel into the summation of two PSD kernels [25, 26]. Alabdulmohsin et al. transferred the indefinite kernel matrix into an affine constraint and the non-convex problem was converted into a linear programming by imposing an additional non-negative constraint on kernel functions' coefficients [27]. However, these approaches either suffer from a dual gap between the primal and dual problems of IKSVM or sacrifice optimization performance and converge to a stationary point.

Furthermore, since multi-class classification problems are very common in reality, it is necessary to extend the binary classification model to the multi-class scenarios [28]. However, all the above algorithms are basically designed for binary classification. For complex multi-class scenarios, they have to resort to some classical multi-class strategies, e.g., One vs. One (OvO) [29, 30], One vs. Rest (OvR) [31] and Many vs. Many (MvM) [32–34] etc. Unfortunately, the algorithms using these strategies would suffer from some drawbacks caused by the strategies themselves, such as the high computational cost in the case of many classes.

In this paper, we propose a primal perspective to solve the non-convex IKSVM problem. That is, we directly focus on the primal form of IKSVM and present a novel algorithm named as IKSVM-DC for both binary and multi-class classification problems. IKSVM-DC firstly formulates the primal problem as a difference of convex functions (DC) programming equivalently, and iteratively optimizes it by the DC algorithm (DCA). For speeding convergence rate, IKSVM-DC then adopts a line search along the descent direction under the Armijo type rule at each iteration in classical DCA. We further extend IKSVM-DC to multi-class classification. The corresponding multi-class IKSVM-DC algorithm can learn a unified model to predict multiple classes all-together. A theoretical analysis is finally presented to validate that IKSVM-DC can converge to a local minimum. We conduct systematic experiments on binary and multi-class real-world datasets respectively. For binary classification problems, the experimental results demonstrate that our algorithm has not only much better classification accuracy compared to some related IKSVM algorithms, but also nearly three times higher convergence rate than the classical DCA. For multi-class classification problems, our algorithm is superior to both OvO-strategy-based and OvR-strategy-based related IKSVM algorithms.

This paper is organized as follows. In Section 2, we analyze that related dual-based IKSVM works would suffer from a dual gap problem. In Section 3, we present a brief introduction to DC programming and DCA. In Section 4, we formulate a primal IKSVM model for binary classification problem and further incorporate the classical DCA into the corresponding algorithm IKSVM-DC. A theoretical analysis is then given for the convergence of IKSVM-DC. The primal multi-class IKSVM model and the corresponding algorithm are derived in Section 5. In Section 6, experiments on both binary and multi-class classification are presented to compare the proposed algorithms with several state-of-the-art related IKSVM methods. Finally, we conclude with some remarks in Section 7.

2 Dual gap problem

Given a training set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathcal{X}$ and $y_i \in \{-1, +1\}$, the soft margin SVM classification is in the formulation:

$$\begin{aligned}\min_{\boldsymbol{w},b,\boldsymbol{\xi}}\ & f_p(\boldsymbol{w},b,\boldsymbol{\xi})=\frac{1}{2}\langle\boldsymbol{w},\boldsymbol{w}\rangle+C\sum_{i=1}^{n}\xi_i,\\ s.t.\ & y_i(\langle\boldsymbol{w},\boldsymbol{x}_i\rangle+b)\geqslant 1-\xi_i,\\ & \xi_i\geqslant 0,\quad i=1,\ldots,n,\end{aligned}\tag{1}$$

and the associated kernelized dual problem [35] is

$$\begin{aligned}\max_{\boldsymbol{\alpha}}\ & f_d(\boldsymbol{\alpha})=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j),\\ s.t.\ & \sum_{i=1}^{n}\alpha_i y_i=0,\quad 0\leqslant\alpha_i\leqslant C,\quad i=1,\ldots,n,\end{aligned}\tag{2}$$

where $K(\cdot,\cdot)$ is a kernel function. Then, the Lagrangian of Eq. (1) is

$$\begin{aligned}&L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\zeta})\\ &=f_p(\boldsymbol{w},b,\boldsymbol{\xi})-\sum_{i=1}^{n}\alpha_i[y_i(\langle\boldsymbol{w},\boldsymbol{x}_i\rangle+b)-1+\xi_i]-\sum_{i=1}^{n}\zeta_i\xi_i\,.\end{aligned}\tag{3}$$

In the view of the primal and dual problems respectively, Eq. (3) can be transformed into these two problems:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} f_p(\boldsymbol{w},b,\boldsymbol{\xi})=p^*=\min_{\boldsymbol{w},b,\boldsymbol{\xi}}\max_{\boldsymbol{\alpha},\boldsymbol{\zeta}} L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\zeta})\,,$$

and

$$\max_{\boldsymbol{\alpha}} f_d(\boldsymbol{\alpha})=d^*=\max_{\boldsymbol{\alpha},\boldsymbol{\zeta}}\min_{\boldsymbol{w},b,\boldsymbol{\xi}} L(\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\zeta})\,,$$

where p^* and d^* are the optimal solutions of the primal and dual problems respectively.

Obviously, the relationship between the two optimal solutions is

$$d^*\leqslant p^*\,.$$

The equality holds if and only if the kernel matrix generated from $K(\cdot,\cdot)$ is PSD [2]. When the kernels become indefinite, the equality would never hold and thus a dual gap exists between the primal and dual problems.

However, many existing IKSVM algorithms still emphasize on the dual problem. For example, proxy kernel algorithms obtain a surrogate PSD kernel matrix for the indefinite kernel directly based on the dual form of IKSVM [19–22]. SMO-type algorithm proposes an improved SMO method to solve the non-convex dual form of IKSVM [23]. Akoa utilized difference of convex functions programming to solve non-convex problems in decomposition methods, but the decomposition methods are still based on the dual form of IKSVM [24]. In order to avoid suffering from the dual gap, we will directly focus on the primal form of IKSVM in this paper.

3 DC programming and DCA

DC programming and DCA [36, 37] are powerful tools for solving smooth/non-smooth non-convex problems which can be decomposed into the form of the subtraction of two convex functions. Concretely, the corresponding objective function f can be formulated as

$$\mathcal{P}=\inf\{f(\boldsymbol{\omega})=g(\boldsymbol{\omega})-h(\boldsymbol{\omega}):\boldsymbol{\omega}\in\mathbb{R}^n\}\,.\tag{4}$$

The two functions g,h are convex and lower semi-continuous on $\mathbb{R}^n$ and take values in $\mathbb{R}\cup\{+\infty\}$. Especially, for some objective functions with a closed convex constraint set $\mathcal{O}\subset\mathbb{R}^n$, DC programming can also extend the variable domain by incorporating an indicator function $\mathbb{I}_{\mathcal{O}}$ of $\mathcal{O}$, i.e., $\mathbb{I}_{\mathcal{O}}(\boldsymbol{\omega})=0$ if $\boldsymbol{\omega}\in\mathcal{O}$, and $+\infty$ otherwise. Thus we have

$$\mathcal{P}=\inf\{f(\boldsymbol{\omega})=(g(\boldsymbol{\omega})+\mathbb{I}_{\mathcal{O}}(\boldsymbol{\omega}))-h(\boldsymbol{\omega}):\boldsymbol{\omega}\in\mathbb{R}^n\}\,.$$

Let $h^*(\boldsymbol{\psi})=\sup\{\langle\boldsymbol{\omega},\boldsymbol{\psi}\rangle-h(\boldsymbol{\omega}),\boldsymbol{\omega}\in\mathbb{R}^n\}$ be the Fenchel conjugate function of h. The dual problem of Eq. (4) can be described as

$$\mathcal{D}=\inf\{f^*(\boldsymbol{\psi})=h^*(\boldsymbol{\psi})-g^*(\boldsymbol{\psi}):\boldsymbol{\psi}\in\mathbb{R}^n\}\,.$$

Due to the property of Fenchel conjugate dual, $\mathcal{P}=\mathcal{D}$ always holds. It means that the primal and dual problems are completely equivalent. Furthermore, two variables $\boldsymbol{\omega}$ and $\boldsymbol{\psi}$ satisfy

$$\boldsymbol{\psi}\in\partial h(\boldsymbol{\omega}),\quad \boldsymbol{\omega}\in\partial g^*(\boldsymbol{\psi})\,,\tag{5}$$

where ∂h and ∂g^* denote the sub-gradients of h and g^* respectively.[1)]

The algorithm DCA further utilizes Eq. (5) to linearize the concave parts $-h$ and $-g^*$ of the two problems and constructs two sequences $\{\boldsymbol{\omega}^k\}$ and $\{\boldsymbol{\psi}^k\}$ for solutions by solving the primal and dual problems alternately. The performance of DCA is affected by three important choices [38]: (1) the explicit choice of the decomposition on f, (2) the choice of the starting point $\boldsymbol{\omega}^0$, (3) the choice of the intermediate convex solver. We will discuss these choices detailedly in our algorithm in Section 6.

4 Primal binary IKSVM classification

In this section, we will firstly construct a primal IKSVM model for binary classification problem. Then we further

[1)] The sub-gradient $\partial h(\boldsymbol{\omega})$ of function $h(\boldsymbol{\omega})$ at $\boldsymbol{\omega}'$ can be defined as $\partial h(\boldsymbol{\omega}')=\{\boldsymbol{\psi}\in\mathbb{R}^n:h(\boldsymbol{\omega})\geqslant h(\boldsymbol{\omega}')+\langle\boldsymbol{\omega}-\boldsymbol{\omega}',\boldsymbol{\psi}\rangle,\forall\boldsymbol{\omega}\in\mathbb{R}^n\}$. Gradient $\nabla h(\boldsymbol{\omega})$ and sub-gradient $\partial h(\boldsymbol{\omega})$ have different requirements for functions. A function can find a gradient if and only if the function is continuous and differentiable while sub-gradient does not need these conditions

characterize the primal binary IKSVM into a DC problem and finally propose a novel algorithm to solve it [39].

4.1 Primal binary IKSVM model

The primal binary problem of IKSVM has the same form as Eq. (1), but the kernel becomes indefinite. So we firstly reformulate Eq. (1) as an unconstrained optimization problem:

$$\min_{\boldsymbol{w},b} \gamma \langle \boldsymbol{w}, \boldsymbol{w} \rangle + \sum_{i=1}^{n} V(y_i, \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b), \tag{6}$$

where the parameter $\gamma = 1/C$ and $V(\cdot)$ is a loss function.

When the kernel is indefinite, we can solve Eq. (6) in a wider RKKS $\mathcal{K}$ as

$$\min_{f \in \mathcal{K}, b} \gamma \langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\mathcal{K}} + \sum_{i=1}^{n} V(y_i, \boldsymbol{f}(\boldsymbol{x}_i) + b). \tag{7}$$

In RKKS, Ong et al. have verified that the Representer Theorem still holds [25] and the solution to the problem of minimizing a regularized risk function can be expanded as

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

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

Consequently, using the Representer Theorem in RKKS, we can further express the primal model in Eq. (7) as

$$\min_{\boldsymbol{\beta},b} \gamma \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\beta} + \sum_{i=1}^{n} V(y_i, \boldsymbol{K}^i \boldsymbol{\beta} + b), \tag{8}$$

where $\boldsymbol{K}$ is the indefinite kernel matrix derived from associated kernel function $\boldsymbol{K}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{K}^i$ represents the ith row of $\boldsymbol{K}$. It is worth noting that the coefficient $\boldsymbol{\beta}$ is not the same as the parameter $\boldsymbol{\alpha}$ in Eq. (2), and thus the coefficient $\boldsymbol{\beta}$ should not be interpreted as a Lagrange multiplier. In fact, the main difference between them is the value range: the parameter $\boldsymbol{\alpha}$ is required to be non-negative but such requirement is inapplicable to the coefficient $\boldsymbol{\beta}$. Furthermore, for the solution $\boldsymbol{\beta}^*$ of Eq. (8), the corresponding support vector set is

$$SVs = \{\boldsymbol{x}_i \in \mathcal{X} \quad s.t. \quad V(y_i, \boldsymbol{K}^i \boldsymbol{\beta}^* + b) \neq 0\},$$

that is, the samples which let the loss function not equal to zero.

In order to make the primal binary IKSVM model continuously differentiable in the variable $\boldsymbol{\beta}$, we select the smooth quadratic hinge loss function as $V(\cdot)$. So the optimization problem in Eq. (8) after adding the scaling constant $1/2$ becomes

$$\min_{\boldsymbol{\beta},b} \frac{1}{2} \left[\gamma \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\beta} + \sum_{i=1}^{n} \max\left(0, 1 - y_i(\boldsymbol{K}^i \boldsymbol{\beta} + b)\right)^2 \right]. \tag{9}$$

Although much similar to the traditional primal PSD kernel SVM, Eq. (9) is actually an unconstrained non-convex optimization which has become an NP-hard problem in terms of indefinite kernels.

4.2 Binary IKSVM converted into a DC problem

The primal IKSVM model can be converted into a DC problem due to the favorable property of the spectra for indefinite kernel matrices, which involve valuable information in kernels. Firstly, we denote the objective function of primal IKSVM as

$$f(\boldsymbol{\beta}) = \frac{1}{2} \left[\gamma \boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\beta} + \sum_{i=1}^{n} \max\left(0, 1 - y_i(\boldsymbol{K}^i \boldsymbol{\beta} + b)\right)^2 \right], \tag{10}$$

and the eigenspectrum of the indefinite kernel matrix can be depicted as $\boldsymbol{K} = U \Lambda U^{\mathrm{T}}$, where U and Λ represent the orthonormal column eigenvector matrix and the diagonal eigenvalue matrix respectively. Λ consists of both positive and negative eigenvalues. Then, we can easily get several equivalent decompositions on Eq. (10) through shifting the eigenspectrum of the indefinite kernels. In our algorithm, we utilize the following two kinds of decompositions, that is, the objective function can be decomposed as $f(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) - h(\boldsymbol{\beta})$ with

$$\textcircled{1} \begin{cases} g(\boldsymbol{\beta}) = & \frac{1}{2}[\gamma \boldsymbol{\beta}^{\mathrm{T}} U(\rho_1 I + \Lambda) U^{\mathrm{T}} \boldsymbol{\beta} \\ & + \sum_{i=1}^{n} \max(0, 1 - y_i(\boldsymbol{K}^i \boldsymbol{\beta} + b))^2], \\ h(\boldsymbol{\beta}) = & \frac{1}{2} \gamma \boldsymbol{\beta}^{\mathrm{T}} U(\rho_1 I) U^{\mathrm{T}} \boldsymbol{\beta}, \end{cases}$$

$$\textcircled{2} \begin{cases} g(\boldsymbol{\beta}) = & \frac{1}{2}[\gamma \boldsymbol{\beta}^{\mathrm{T}} U(\rho_2 I) U^{\mathrm{T}} \boldsymbol{\beta} \\ & + \sum_{i=1}^{n} \max(0, 1 - y_i(\boldsymbol{K}^i \boldsymbol{\beta} + b))^2], \\ h(\boldsymbol{\beta}) = & \frac{1}{2} \gamma \boldsymbol{\beta}^{\mathrm{T}} U(\rho_2 I - \Lambda) U^{\mathrm{T}} \boldsymbol{\beta}, \end{cases} \tag{11}$$

where the two positive numbers ρ_1 and ρ_2 are chosen to guarantee that the two functions $g(\boldsymbol{\beta})$ and $h(\boldsymbol{\beta})$ are convex functions, i.e., $\rho_1 \geqslant -\min(\{\lambda_i\}_{i=1}^n)$ and $\rho_2 \geqslant \max(\{\lambda_i\}_{i=1}^n)$, and the set $\{\lambda_i\}_{i=1}^n$ represents eigenvalues in the eigenvalue matrix Λ.

Given the decomposition of primal IKSVM model, we can obtain the conjugate dual problem of function $f(\boldsymbol{\beta})$, i.e., $\min_{\boldsymbol{\theta} \in \mathbb{R}^n} \{f^*(\boldsymbol{\theta}) = h^*(\boldsymbol{\theta}) - g^*(\boldsymbol{\theta})\}$. According to the property of DC programming in Eq. (5), we have

$$\boldsymbol{\theta} \in \partial h(\boldsymbol{\beta}), \quad \boldsymbol{\beta} \in \partial g^*(\boldsymbol{\theta}). \tag{12}$$

Utilizing Eq. (12), we can approximate the function h with its affine minorization at point $\boldsymbol{\beta}_t$

$$h(\boldsymbol{\beta}) \geqslant h(\boldsymbol{\beta}_t) + \langle \boldsymbol{\beta} - \boldsymbol{\beta}_t, \boldsymbol{\theta}_t \rangle , \tag{13}$$

where $\boldsymbol{\theta}_t \in \partial h(\boldsymbol{\beta}_t)$. At point $\boldsymbol{\theta}_t$, the function g^* of conjugate dual problem can be formulated as

$$g^*(\boldsymbol{\theta}) \geqslant g^*(\boldsymbol{\theta}_t) + \langle \boldsymbol{\theta} - \boldsymbol{\theta}_t, \boldsymbol{\beta}_{t+1} \rangle , \tag{14}$$

where $\boldsymbol{\beta}_{t+1} \in \partial g^*(\boldsymbol{\theta}_t)$. As a result, the primal IKSVM problem and its conjugate dual problem become convex after the transformation in Eqs. (13) and (14).

We further construct two sequences $\{\boldsymbol{\beta}_t\}$ and $\{\boldsymbol{\theta}_t\}$ for solutions by solving Eq. (15) alternately

$$\begin{cases} \boldsymbol{\beta}_{t+1} = \arg\min\limits_{\boldsymbol{\beta}\in\mathbb{R}^n} g(\boldsymbol{\beta}) - \langle \boldsymbol{\beta}, \boldsymbol{\theta}_t \rangle, \ \boldsymbol{\beta}_{t+1} \in \{\boldsymbol{\beta}_t\}, \\ \boldsymbol{\theta}_{t+1} = \arg\min\limits_{\boldsymbol{\theta}\in\mathbb{R}^n} h^*(\boldsymbol{\theta}) - \langle \boldsymbol{\theta}, \boldsymbol{\beta}_{t+1} \rangle, \ \boldsymbol{\theta}_{t+1} \in \{\boldsymbol{\theta}_t\} . \end{cases} \tag{15}$$

Following [37], we omit the conjugate dual problem with a simplified form $\boldsymbol{\theta}_t \in \partial h(\boldsymbol{\beta}_t)$ in practice, and obtain

$$\begin{cases} \boldsymbol{\theta}_t \in \partial h(\boldsymbol{\beta}_t), \\ \boldsymbol{\beta}_{t+1} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^n} g(\boldsymbol{\beta}) - \langle \boldsymbol{\beta}, \boldsymbol{\theta}_t \rangle . \end{cases} \tag{16}$$

The sequence $\{\boldsymbol{\beta}_t\}$ can generate a descent direction at each iteration. In order to accelerate the convergence rate, we further search the smallest non-negative integer l_t under the Armijo type rule along the direction to achieve a larger reduction in the value of f [40]

$$f(\boldsymbol{\beta}_{t+1} + \eta^{l_t} d(\boldsymbol{\beta})) \leqslant f(\boldsymbol{\beta}_{t+1}) - \mu\eta^{l_t} \|d(\boldsymbol{\beta})\|^2 .$$

Algorithm 1 summarizes the procedure of our algorithm binary IKSVM-DC [2)]. Given the training set, a DC decomposition is chosen to formulate the primal binary IKSVM into a DC problem (Step 2). After that, an iterative DC algorithm is performed to obtain the solutions for primal binary IKSVM problem and its conjugate dual problem (Steps 4–9). Meanwhile, a line search step is conducted to accelerate the convergence of binary IKSVM-DC (Steps 10–14). Finally, the unseen instance is classified based on the solutions (Step 16).

4.3 Convergence analysis

In this subsection, we will present a theoretical analysis for the convergence of binary IKSVM-DC.

Proposition 1 For the sequence $\{\boldsymbol{\beta}_t\}$, we have

$$(g-h)(\boldsymbol{\beta}_t) - (g-h)(\boldsymbol{\beta}_{t+1}) \geqslant \tau \|d(\boldsymbol{\beta})\|^2 ,$$

the equality holds if and only if $\tau\|d(\boldsymbol{\beta})\|^2 = 0$, where τ is a positive parameter to make functions g and h strongly convex.

[2)] Code package is available at the corresponding author's homepage

Algorithm 1 Binary IKSVM-DC

Inputs:

$\mathcal{D}$: the training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n \in \mathbb{R}^m \times \{\pm 1\}$

γ: the regularization parameter

$\bar{\upsilon}$: 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 unseen instance

Outputs:

y^*: the predicted class label for $\boldsymbol{x}^*$

Process:

1: Initialize the kernel coefficient $\boldsymbol{\beta}_0$ and set $t = 0$;
2: Choose a DC decomposition: $f(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) - h(\boldsymbol{\beta})$;
3: **while** $t < T$ **do**
4: Obtain a solution for conjugate dual problem: $\boldsymbol{\theta}_t = \nabla h(\boldsymbol{\beta}_t)$;
5: Solve the convex minimization problem in Eq. (16) to obtain a solution $\boldsymbol{\beta}_{t+1}$ for primal IKSVM problem;
6: Set $d(\boldsymbol{\beta}) = \boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t$;
7: **if** $\|d(\boldsymbol{\beta})\|^2 \leqslant \delta$ **then**
8: IKSVM-DC converges to a local minimum and break;
9: **end if**
10: Set $\upsilon_t = \bar{\upsilon}$;
11: **while** $f(\boldsymbol{\beta}_{t+1} + \upsilon_t d(\boldsymbol{\beta})) > f(\boldsymbol{\beta}_{t+1}) - \mu\upsilon_t \|d(\boldsymbol{\beta})\|^2$ **do**
12: $\upsilon_t = \eta\upsilon_t$;
13: **end while**
14: Update the solution of IKSVM: $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_{t+1} + \upsilon_t d(\boldsymbol{\beta})$ and set $t = t+1$;
15: **end while**
16: **return** $y^* = \text{sign}(K(\boldsymbol{x}^*, \boldsymbol{x})\boldsymbol{\beta} + b)$;

Proof Firstly, we can construct the the convex functions g, h as being strongly convex with an additional term $\frac{\tau}{2}\boldsymbol{\beta}^2$:

$$(g-h)(\boldsymbol{\beta}) = \underbrace{\left(g(\boldsymbol{\beta}) - \frac{\tau}{2}\boldsymbol{\beta}^2\right)}_{G(\boldsymbol{\beta})} - \underbrace{\left(h(\boldsymbol{\beta}) - \frac{\tau}{2}\boldsymbol{\beta}^2\right)}_{H(\boldsymbol{\beta})} .$$

Then given the convexity of function G, we have

$$G(\boldsymbol{\beta}_t) \geqslant G(\boldsymbol{\beta}_{t+1}) + \nabla G(\boldsymbol{\beta}_{t+1})(\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1})^{\mathrm{T}} .$$

After simplified, we get

$$g(\boldsymbol{\beta}_t) \geqslant g(\boldsymbol{\beta}_{t+1}) + \langle \nabla g(\boldsymbol{\beta}_{t+1}), \boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1} \rangle + \frac{\tau}{2}\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1}\|^2 . \tag{17}$$

Similarly, for the function H, we can get

$$H(\boldsymbol{\beta}_{t+1}) \geqslant H(\boldsymbol{\beta}_t) + \nabla H(\boldsymbol{\beta}_t)(\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t)^{\mathrm{T}} ,$$

$$h(\boldsymbol{\beta}_{t+1}) \geqslant h(\boldsymbol{\beta}_t) + \langle \nabla h(\boldsymbol{\beta}_t), \boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t \rangle + \frac{\tau}{2}\|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t\|^2 . \tag{18}$$

Since $\boldsymbol{\beta}_{t+1}$ is a unique solution of the convex problem in Eq. (16), we have

$$\nabla g(\boldsymbol{\beta}_{t+1}) = \boldsymbol{\theta}_t = \nabla h(\boldsymbol{\beta}_t) . \tag{19}$$

Combining Eqs. (17), (18) and (19), we have

$$(g(\boldsymbol{\beta}_t) - h(\boldsymbol{\beta}_t)) - (g(\boldsymbol{\beta}_{t+1}) - h(\boldsymbol{\beta}_{t+1})) \geqslant \tau\|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t\|^2 .$$

Proposition 1 presents that IKSVM-DC can decrease the value of objective function at each iteration and further provides a condition $\|d(\boldsymbol{\beta})\|^2 = 0$ for the convergence to IKSVM-DC. The following Proposition 2 verifies that $d(\boldsymbol{\beta}) = \boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t$ is a descent direction for f at $\boldsymbol{\beta}_{t+1}$ and thus we can conduct a line search along the direction in IKSVM-DC to further decrease the value of objective function.

Proposition 2 For the sequence $\{\boldsymbol{\beta}_t\}$, we have

$$\langle \nabla(g-h)(\boldsymbol{\beta}_{t+1}), \boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t \rangle \leqslant 0\,,$$

that is, $d(\boldsymbol{\beta}) = \boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t$ is a descent direction for $f = g - h$ at $\boldsymbol{\beta}_{t+1}$.

Proof Following Proposition 1, we have

$$h(\boldsymbol{\beta}_t) \geqslant h(\boldsymbol{\beta}_{t+1}) + \langle \nabla h(\boldsymbol{\beta}_{t+1}), \boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1} \rangle + \frac{\tau}{2}\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1}\|^2\,. \quad (20)$$

Given the Eq. (20), we can know that the function $h(\boldsymbol{\beta})$ is strongly convex on $\mathbb{R}^n$. Then according to the Theorem 2.1.9 in [41], we have

$$\langle \nabla h(\boldsymbol{\beta}_t) - \nabla h(\boldsymbol{\beta}_{t+1}), \boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1} \rangle \geqslant \tau\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1}\|^2\,.$$

Combining Eq. (19), we have

$$\langle \nabla g(\boldsymbol{\beta}_{t+1}) - \nabla h(\boldsymbol{\beta}_{t+1}), \boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t \rangle \leqslant -\tau\|d(\boldsymbol{\beta})\|^2 \leqslant 0\,,$$

the equality holds if and only if $\tau\|d(\boldsymbol{\beta})\|^2 = 0$.

Based on Propositions 1 and 2, we can further validate that IKSVM-DC can converge to a local optimum.

Theorem 1 If the sequence $\{\boldsymbol{\beta}_t\}$ satisfies $d(\boldsymbol{\beta}) = \boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t = 0$, let $\boldsymbol{\beta}^* = \boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t$ and $\mathcal{U}$ be a neighbourhood of $\boldsymbol{\beta}^*$. For $\forall \boldsymbol{\beta} \in \mathcal{U}$, we have

$$g(\boldsymbol{\beta}) - h(\boldsymbol{\beta}) \geqslant g(\boldsymbol{\beta}^*) - h(\boldsymbol{\beta}^*)\,.$$

Proof Following Eq. (19), the condition $d(\boldsymbol{\beta}) = \boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t = 0$ implies $\nabla g(\boldsymbol{\beta}^*) = \nabla g(\boldsymbol{\beta}_{t+1}) = \boldsymbol{\theta}_t$, that is, $\exists \boldsymbol{\theta} \in \partial g(\boldsymbol{\beta}^*)$. So the conjugate function of g at $\boldsymbol{\beta}^*$ is

$$g^*(\boldsymbol{\theta}) = \sup\{\langle \boldsymbol{\beta}^*, \boldsymbol{\theta} \rangle - g(\boldsymbol{\beta}^*)\} = \langle \boldsymbol{\beta}^*, \boldsymbol{\theta} \rangle - g(\boldsymbol{\beta}^*)\,, \quad (21)$$

and $\forall \boldsymbol{\theta} \in \mathbb{R}^n$, the conjugate function of h at $\boldsymbol{\beta}^*$ is

$$h^*(\boldsymbol{\theta}) = \sup\{\langle \boldsymbol{\beta}^*, \boldsymbol{\theta} \rangle - h(\boldsymbol{\beta}^*)\} \geqslant \langle \boldsymbol{\beta}^*, \boldsymbol{\theta} \rangle - h(\boldsymbol{\beta}^*)\,. \quad (22)$$

Combining Eqs. (21) and (22), we have

$$g(\boldsymbol{\beta}^*) + g^*(\boldsymbol{\theta}) = \langle \boldsymbol{\beta}^*, \boldsymbol{\theta} \rangle \leqslant h(\boldsymbol{\beta}^*) + h^*(\boldsymbol{\theta})\,. \quad (23)$$

On the other hand, since $\boldsymbol{\theta} = \nabla h(\boldsymbol{\beta})$, it means $\exists \boldsymbol{\theta} \in \partial h(\boldsymbol{\beta})$. Similar to the process in Eqs. (21), (22) and (23), we have

$$h(\boldsymbol{\beta}) + h^*(\boldsymbol{\theta}) = \langle \boldsymbol{\beta}, \boldsymbol{\theta} \rangle \leqslant g(\boldsymbol{\beta}) + g^*(\boldsymbol{\theta})\,. \quad (24)$$

Combining Eqs. (23) and (24), we can reach the conclusion.

5 Primal multi-class IKSVM classification

In this section, we extend the binary IKSVM model to a unified multi-class formulation for multi-class problems.

5.1 Primal multi-class IKSVM model

Given a multi-class training set $\{(\boldsymbol{x}_i, c_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathcal{X}$ and $c_i \in \{1, 2, \ldots, c\}$. Since the primal binary IKSVM model is not suitable for the multiple classes scenarios, here we construct another multi-class model with different loss term and regularization term. Firstly, based on the Representer Theorem in RKKS [25], the solution to the multi-class problem of minimizing a regularized risk function can be expanded as

$$\boldsymbol{f}^* = \sum_{j=1}^{c}\sum_{i=1}^{n} B_{ji} K(\boldsymbol{x}_i, \cdot)\,,$$

where c is the number of classes and $\boldsymbol{B} \in \mathbb{R}^{c\times n}$ is the coefficient of kernel K. Then, considering the classes are independent from each other in the multi-class problem, we can have the following formulation in a trace form for the regularization term

$$\langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\mathcal{K}} = tr(\boldsymbol{B}\boldsymbol{K}\boldsymbol{B}^{\mathrm{T}})\,,$$

where $tr(\cdot)$ represents the trace operation.

Traditionally, in order to tackle multi-class classification problems, many methods would resort to different classification strategies, e.g., OvO, OvR and MvM, which focus on the combination of several binary classification classifiers. However, all these strategies have to train a large number of classifiers. Especially, OvO strategy must construct $c(c-1)/2$ classifiers which results in high computational cost [42]. Thus we will construct a multi-class loss function to consider all classes in an unified optimization formulation. Motivated by many "all-together" methods [43–45], we construct a multi-class loss function in the following formulation

$$\sum_{i=1}^{n} V(\cdot) = \sum_{i=1}^{n}\sum_{j\neq c_i} \max(0, \boldsymbol{B}_j K^i - \boldsymbol{B}_{c_i} K^i + 1)\,,$$

where $\boldsymbol{B}_j$ and $\boldsymbol{B}_{c_i}$ represent the jth row and the c_ith row of the coefficient matrix $\boldsymbol{B}$ respectively. K^i represents the ith row

of the kernel matrix K. This multi-class loss function aims to maximize the margin between the correct class and other wrong classes.

Consequently, we can obtain the multi-class IKSVM model by combining the regularization term and the loss term together

$$\min_{\boldsymbol{B}} \frac{1}{2}\gamma tr(\boldsymbol{B}K\boldsymbol{B}^{\mathrm{T}}) + \sum_{i=1}^{n}\sum_{j\neq c_i}\max(0, \boldsymbol{B}_j K^i - \boldsymbol{B}_{c_i}K^i + 1)\,,$$

where the additional term $1/2$ is for the convenience of derivation which would be used in the following section. γ is the regularization parameter.

When the optimal solution $\boldsymbol{B}^*$ is obtained, for an unseen sample, the predict scores of every class can be computed as

$$C_j = \sum_{i=1}^{n} B^*_{ji} K(\boldsymbol{x}_i, \boldsymbol{x}')\,, \tag{25}$$

where $\boldsymbol{x}'$ is an unseen sample and $\boldsymbol{C} = \{C_1, \ldots, C_j, \ldots, C_c\} \in \mathbb{R}^{1\times c}$ is the scores of every class. Then we choose the class corresponding to the maximal score to be the predicted class for the sample, that is

$$\text{predicted\_class} = \max_{1,\ldots,j,\ldots,c} C_j\,. \tag{26}$$

5.2 DC form of primal multi-class IKSVM

We denote the primal multi-class IKSVM model as the following function

$$f(\boldsymbol{B}) = \frac{1}{2}\gamma tr(\boldsymbol{B}K\boldsymbol{B}^{\mathrm{T}}) + \sum_{i=1}^{n}\sum_{j\neq c_i}\max(0, \boldsymbol{B}_j K^i - \boldsymbol{B}_{c_i}K^i + 1)\,. \tag{27}$$

The corresponding second derivative of function $f(\boldsymbol{B})$ is $\nabla^2 f(\boldsymbol{B}) = \gamma \boldsymbol{K}$.[3] Since the function $f(\boldsymbol{B})$ is convex if and only if $\nabla^2 f(\boldsymbol{B}) \succeq 0$, the objective function $f(\boldsymbol{B})$ will be a non-convex problem when the kernel matrix K becomes indefinite.

Observing the primal multi-class IKSVM Model in Eq. (27), we can see that this model is actually a function of matrix variable $\boldsymbol{B}$. Thanks to the property of trace operation, the primal multi-class IKSVM model still satisfies the two decomposition formulations in Section 2 and here we take one of the decompositions as an example. That is, Eq. (27) can be written as $f(\boldsymbol{B}) = g(\boldsymbol{B}) - h(\boldsymbol{B})$ where

$$\begin{cases} g(\boldsymbol{B}) = \dfrac{1}{2}\gamma tr(\boldsymbol{B}U(\rho_1 I + \Lambda)U^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}) \\ \qquad\qquad + \displaystyle\sum_{i=1}^{n}\sum_{j\neq c_i}\max(0, \boldsymbol{B}_j K^i - \boldsymbol{B}_{c_i}K^i + 1) \\ h(\boldsymbol{B}) = \dfrac{1}{2}\gamma tr(\boldsymbol{B}U(\rho_1 I)U^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}})\,, \end{cases} \tag{28}$$

where U and Λ represent the orthonormal column eigenvector matrix and the diagonal eigenvalue matrix of kernel matrix $\boldsymbol{K}$ respectively. The positive numbers ρ_1 is chosen to guarantee that the two functions $g(\boldsymbol{B})$ and $h(\boldsymbol{B})$ are convex functions.

Furthermore, given the DC form of multi-class IKSVM model in Eq. (28), we can also obtain the following two solution sequences

$$\begin{cases} \{\boldsymbol{B}_t\} = \arg\min_{\boldsymbol{B}\in\mathbb{R}^{c\times n}}\{\boldsymbol{B}_{t+1} : g(\boldsymbol{B}) - tr(\langle \boldsymbol{B}, \boldsymbol{\Theta}_t\rangle)\}, \\ \{\boldsymbol{\Theta}_t\} = \arg\min_{\boldsymbol{\Theta}\in\mathbb{R}^{c\times n}}\{\boldsymbol{\Theta}_{t+1} : h^*(\boldsymbol{\Theta}) - tr(\langle \boldsymbol{\Theta}, \boldsymbol{B}_{t+1}\rangle)\}, \end{cases}$$

where $\boldsymbol{\Theta}$ is the conjugate dual variable of objective function $f(\boldsymbol{B})$ and $h^*(\boldsymbol{\Theta})$ is the conjugate dual function. Then, in practice, we omit the conjugate dual problem with a simplified form $\boldsymbol{\Theta}_t \in \nabla h(\boldsymbol{B}_t)$ and obtain Eq. (29)[4]

$$\begin{cases} \boldsymbol{\Theta}_t \in \nabla h(\boldsymbol{B}_t), \\ \boldsymbol{B}_{t+1} \in \arg\min_{\boldsymbol{B}\in\mathbb{R}^{c\times n}} g(\boldsymbol{B}) - tr(\langle \boldsymbol{B}, \boldsymbol{\Theta}_t\rangle)\,. \end{cases} \tag{29}$$

For the matrix sequence $\{\boldsymbol{B}_t\}$, we can also prove that the directions between each iteration are all descent directions (See Proposition 4). Combining these descent directions together, we can use the following formulation to further reduce the objective function value and accelerate the convergence rate

$$f(\boldsymbol{B}_{t+1} + \eta^{l_t} d(\boldsymbol{B})) \leqslant f(\boldsymbol{B}_{t+1}) - \mu\eta^{l_t}\|d(\boldsymbol{B})\|_F^2\,,$$

where $d(\boldsymbol{B}) = \boldsymbol{B}_{t+1} - \boldsymbol{B}_t$ and $\|d(\boldsymbol{B})\|_F$ represents the Frobenius norm of $d(\boldsymbol{B})$.

The overall algorithm procedure for multi-class IKSVM-DC algorithm[5] is similar to binary IKSVM-DC in Algorithm 1 except for several steps, i.e., the condition for **if** in step 7 is $\|d(\boldsymbol{B})\|_F^2 \leqslant \delta$, the condition for **while** in step 11 is $f(\boldsymbol{B}_{t+1} + \upsilon_t d(\boldsymbol{B})) > f(\boldsymbol{B}_{t+1}) - \mu\upsilon_t\|d(\boldsymbol{B})\|_F^2$, and the class predict formulation in step 16 is the combination of Eqs. (25) and (26).

[3] The derivative of trace in Eq. (27) is $d/d\boldsymbol{B}(tr(\boldsymbol{B}K\boldsymbol{B}^{\mathrm{T}})) = \boldsymbol{B}(\boldsymbol{K} + \boldsymbol{K}^{\mathrm{T}}) = 2\boldsymbol{B}\boldsymbol{K}$, and the second equation holds because the kernel matrix $\boldsymbol{K}$ is symmetry

[4] $\nabla h(\boldsymbol{B}_t)$ represents the gradient of function $h(\boldsymbol{B})$ at variable matrix $\boldsymbol{B}_t$. Since the classes are independent from each other in the multi-class problem, we have $\nabla h(\boldsymbol{B}_t) = [\nabla h(\boldsymbol{B}_t^1); \ldots; \nabla h(\boldsymbol{B}_t^i); \ldots; \nabla h(\boldsymbol{B}_t^c)]$ where $\boldsymbol{B}_t^i$ represents the ith row of the matrix $\boldsymbol{B}_t$

[5] Code package is available at the corresponding author's homepage

5.3 Convergence analysis

In this subsection, we will present a theoretical analysis for the convergence of multi-class IKSVM model. Since the deduction process of the corresponding theoret- ical results are similar to the ones of binary IKSVM, we only present the results here and the detailed proofs can be found in the appendix A.

Firstly, we have the following proposition to show that the multi-class IKSVM-DC algorithm would decrease the multi-class IKSVM objective function value.

Proposition 3 For the sequence $\{\boldsymbol{B}_t\}$, we have

$$(g-h)(\boldsymbol{B}_t) - (g-h)(\boldsymbol{B}_{t+1}) \geqslant \tau \|d(\boldsymbol{B})\|_F^2 ,$$

the equality holds if and only if $\tau\|d(\boldsymbol{B})\|_F^2 = 0$, where τ is a positive parameter to make functions g and h strongly convex.

According to Proposition 3, the equation $\|d(\boldsymbol{B})\|_F^2 = 0$ is also a condition for the convergence to multi-class IKSVM-DC algorithm.

Then Proposition 4 verifies that every direction $d(\boldsymbol{B}^i) = \boldsymbol{B}_{t+1}^i - \boldsymbol{B}_t^i$ is a descent direction for f at $\boldsymbol{B}_{t+1}^i$ and thus we can conduct a line search along these directions in multi-class IKSVM-DC algorithm to further decrease the value of objective function.

Proposition 4 For the sequence $\{\boldsymbol{B}_t\} = \{[\boldsymbol{B}_t^1; \ldots; \boldsymbol{B}_t^i; \ldots; \boldsymbol{B}_t^c]\}$ where $\boldsymbol{B}_t^i$ is one row of the variable matrix $\boldsymbol{B}_t$, we have

$$\langle \nabla(g-h)(\boldsymbol{B}_{t+1}^i), d(\boldsymbol{B}^i)\rangle \leqslant 0, \forall i \in [1, \ldots, c] ,$$

that is, $d(\boldsymbol{B}^i) = \boldsymbol{B}_{t+1}^i - \boldsymbol{B}_t^i$ is a descent direction for $f = g - h$ at $\boldsymbol{B}_{t+1}^i$.

Based on Propositions 3 and 4, we can further validate that multi-class IKSVM-DC algorithm can also converge to a local optimum.

Theorem 2 If the sequence $\{\boldsymbol{B}_t\}$ satisfies $d(\boldsymbol{B}) = \boldsymbol{B}_{t+1} - \boldsymbol{B}_t = 0$, let $\boldsymbol{B}^* = \boldsymbol{B}_{t+1} = \boldsymbol{B}_t$ and $\mathcal{U}$ be a neighbourhood of $\boldsymbol{B}^*$. For $\forall \boldsymbol{B} \in \mathcal{U}$, we have

$$g(\boldsymbol{B}) - h(\boldsymbol{B}) \geqslant g(\boldsymbol{B}^*) - h(\boldsymbol{B}^*) .$$

6 Experiments

We experimentally evaluate the performance of the proposed algorithm IKSVM-DC for both binary and multi-class classification through comparing with several related IKSVM algorithms using a collection of datasets on the benchmark.

6.1 Experimental setup

In the experiments, nineteen real-world datasets are used for learning IKSVMs, including six datasets {*Ionosphere*, *Sonar*, *Dermatology*, *Ecoli*, *NewThyriod*, *Glass*} from UCI Machine Learning Repository [46], four datasets {*Titanic*, *Breast − cancer*, *Thyroid*, *Flare − solar*} from IDA database [47], and the rest nine dissimilarity datasets are {*Balls3D*, *Protein*, *CoilYork*, *Zongker*, *Chickenpieces* – 5 – 45, *Chickenpieces* − 10 − 45, *Chickenpieces* − 20 − 45, *Chickenpieces* − 30 − 45, *Chickenpieces* − 40 − 45}[6] provided by similarity-based pattern analysis and pecognition project [48]. Table 1 lists a brief description of these datasets and the corresponding similarity measures.

For the UCI and IDA datasets, we randomly divide the samples into two non-overlapping training and testing sets which contain almost half of the samples in each class. For all the dissimilarity datasets, we extract half of the points from the dissimilarity matrix for training set and the rest for testing set. The processes are repeated ten times to generate ten independent epoches for each dataset, and then the average results are reported.

For all the datasets, we choose the regularization parameter γ and the parameters in sigmoid kernels by ten-fold cross-validation on the training set from the set $\{2^{-6}, 2^{-5}, \ldots, 2^5, 2^6\}$.

As the algorithms IKSVM-DC for both binary and multi-class classification are both quadratic programming without constraints, we utilize the interior-point optimizer to solve it by Mosek optimization software [49]. Moreover, since the values in variables $\boldsymbol{\beta} \in \mathbb{R}^n$ and $\boldsymbol{B} \in \mathbb{R}^{c\times n}$ can be negative, we randomly initialize $\{\boldsymbol{\beta}_0, \boldsymbol{B}_0\} \in [-1, +1]$. As a result, considering the three factors of DCA described above, we only need to take the decomposition of f into consideration in the experiments, which is depicted in Eq. (11).

Since many datasets used in our experiments are naturally incorporated with indefinite kernels and traditional SVM methods fail to train on these datasets, we compare IKSVM-DC with several state-of-the-art IKSVM algorithms including:

- "Clip", "Flip" and "Shift" [50]: three methods directly change the eigenspectrum to obtain a PSD kernel matrix, and take the modified PSD kernel into a dual form of SVM.

[6] The numbers in chicken pieces datasets are two parameters to calculate the weight edit distance between two chicken pieces images where the first number represents straight line segment of a fixed length and the second number represents the angles between the neighbouring segments and editing cost. Here we try to observe the influence of the second parameter on the distance metric by fixing the first parameter

Table 1 Datasets description

UCI & IDA Datasets	Abbreviation	# Examples	# Features	# Classes	ϕ[7)]	Measure
Ionosphere	Ion.	351	33	2	0.340	Sigmoid kernel
Sonar	Son.	208	60	2	0.290	Sigmoid kernel
Titanic	Tit.	2,201	3	2	0.261	Sigmoid kernel
Breast-cancer	Bre.	277	9	2	0.718	Sigmoid kernel
Thyroid	Thy.	215	5	2	0.470	Sigmoid kernel
Flare-solar	Fla.	1,066	9	2	0.211	Sigmoid kernel
Dermatology	Der.	366	33	6	0.093	Sigmoid kernel
Ecoli	Eco.	332	6	6	0.143	Sigmoid kernel
NewThyriod	New.	215	5	3	0.085	Sigmoid kernel
Glass	Gla.	214	9	6	0.172	Sigmoid kernel
Dissimilarity Datasets	**Abbreviation**	**# Examples**	**# Classes**	ϕ[7)]	**Measure**	
Balls3D	Bal.	200	2	0.500	Distance on 3-D balls	
Protein	Pro.	213	2	0.500	Protein sequences matching	
CoilYork	Coi.	288	4	0.500	Graph matching	
Zongker	Zon.	2,000	10	0.497	Handwritten digits matching	
Chickenpieces-5-45	C5.	446	5	0.500	Chicken pieces images' distance	
Chickenpieces-10-45	C10.	446	5	0.500	Chicken pieces images' distance	
Chickenpieces-20-45	C20.	446	5	0.500	Chicken pieces images' distance	
Chickenpieces-30-45	C30.	446	5	0.500	Chicken pieces images' distance	
Chickenpieces-40-45	C40.	446	5	0.500	Chicken pieces images' distance	

- SMO-IKSVM [23]: a method utilizes the SMO-type algorithm to solve the dual form of IKSVM.
- TDCASVM [24]: a method uses DC algorithm to solve non-convex dual problems in decomposition methods.
- IKSVM-CA [22]: a method iteratively achieves a low dimensional representation PSD kernel matrix for the indefinite kernel, and solves the dual form of SVM with the PSD kernel matrix.
- ESVM [26]: a method transforms the indefinite kernel from Kreĭn spaces into Hilbert spaces, and trains the convex dual form of SVM.
- 1-norm IKSVM [27]: a method imposes the coefficients of kernel functions to be non-negative in 1-norm IKSVM, and tackles the convex problem by Mosek optimization software [49].

The dual problem of SVM/IKSVM in the above algorithms is all solved by the LIBSVM library [51]. For multi-class classification problems, all the algorithms above are not easy to develop an unified model to solve the multi-class problems. Thus we extend all these algorithms with OvO and OvR strategies to train multiple binary classifiers.

6.2 Binary classification results

Table 2 reports the performance of each compared algorithm on the two-class real-world datasets, where the mean classification accuracies as well as the standard deviations of each algorithm are recorded and the best results are highlighted in bold. Furthermore, to statistically measure the significance of performance difference, pairwise t-test at 0.05 significance level is conducted between the algorithms. Specifically, when IKSVM-DC is significantly superior/inferior to the compared algorithm on any dataset, a marker $\bullet/\circ$ is shown. Otherwise, no marker is given [52].

We conduct experiments on the two kinds of decompositions, and the classification accuracies of these two decompositions are comparable, which means that IKSVM-DC is robust for the decomposition factor. Thus we choose the higher classification accuracy as the final result to show in Table 2. It is impressive that IKSVM-DC outperforms all the other algorithms on the eight datasets. Among the eight algorithms, three spectrum transformation methods obtain the lowest classification accuracies on six of the eight datasets. SMO-IKSVM and TDCASVM achieve similar results to three spectrum transformation methods. IKSVM-CA slightly excels the spectrum transformation methods on seven datasets. But it has too much parameters to tune and would fail when the number of positive eigenvalues is very small (e.g., the *Breast − cancer* dataset). ESVM exceeds IKSVM-CA on half of the datasets yet is worse than 1-norm IKSVM on most of these datasets. Our algorithm IKSVM-DC is

7) $\phi = \frac{\sum_{i=1}^{n} |\lambda_i| \cdot \mathbb{I}\{\lambda_i < 0\}}{\sum_{i=1}^{n} |\lambda_i|}$ represents the measure of indefiniteness for the datasets

Table 2 Classification accuracy (mean±std. deviation) of each compared algorithm on several real-world binary datasets. In addition, •/◦ indicates whether IKSVM-DC is statistically superior/inferior to the compared algorithm on each dataset (pairwise t-test at 0.05 significance level)

	Clip	Flip	Shift	SMO-IKSVM	TDCASVM	IKSVM-CA	ESVM	1-norm IKSVM	IKSVM-DC
Ion.	73.72%±0.104•	75.92%±0.086•	67.72%±0.055•	73.11%±0.108•	74.93%±0.047•	86.56%±0.057•	88.68%±0.020•	91.93%±0.016•	**93.64%±0.011**
Son.	67.63%±0.062•	68.93%±0.017•	65.83%±0.047•	64.94%±0.068•	63.86%±0.072•	75.86%±0.030•	73.44%±0.027•	79.23%±0.030•	**84.83%±0.023**
Tit.	73.62%±0.068•	77.40%±0.009•	71.78%±0.071•	74.3%4±0.051•	73.60%±0.043•	78.84%±0.005	78.82%±0.005	78.76%±0.005•	**79.18%±0.004**
Bre.	73.11%±0.022•	73.69%±0.023•	71.34%±0.007•	72.79%±0.020•	74.16%±0.022•	37.50%±0.395•	73.47%±0.027•	73.85%±0.026•	**78.33%±0.015**
Thy.	89.94%±0.039•	92.16%±0.036•	75.78%±0.074•	87.22%±0.041•	87.78%±0.057•	94.05%±0.025•	92.76%±0.051•	94.17%±0.034•	**97.73%±0.019**
Fla.	60.46%±0.052•	58.91%±0.050•	55.37%±0.000•	58.81%±0.049•	56.97%±0.026•	66.42%±0.039	63.27%±0.055•	62.35%±0.059•	**68.16%±0.013**
Bal.	47.87%±0.055•	47.16%±0.031•	48.28%±0.053•	49.98%±0.035•	55.84%±0.016•	51.31%±0.040•	53.68%±0.029•	54.69%±0.044•	**57.08%±0.031**
Pro.	67.90%±0.000•	67.97%±0.000•	67.91%±0.000•	67.93%±0.000•	68.98%±0.029•	95.81%±0.063•	99.07%±0.010	85.98%±0.070•	**99.91%±0.003**

superior to 1-norm IKSVM on all the datasets. Especially, the classification results corresponding to two dissimilarity datasets *Balls*3*D* and *Protein* differ hugely in all compared algorithms. The reason can obtain by observing the intensity images[8] of dissimilarity matrices: For the *Balls*3*D* dataset, the dissimilarity between two classes is very small and it would be hard for classifiers to discriminate these kinds of samples correctly. While for the *Protein* dataset, it is very clear that the dissimilarity of two classes is so large that most of the non-convex optimization methods obtain good classification accuracies.

The experiments about the convergence of IKSVM-DC are conducted on four datasets *Ionosphere*, *Sonar*, *Flare*−*solar* and *Balls*3*D*. We plot the value $\|d(\boldsymbol{\beta})\|^2 = \|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t\|^2$ of the solution sequence $\{\boldsymbol{\beta}_t\}$ during the iterations, as shown in Fig. 1. We can see that the value $\|d(\boldsymbol{\beta})\|^2$ gradually converges in a few iterations on the four datasets.

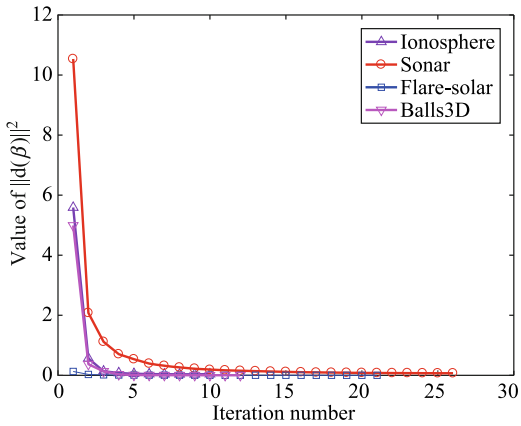


Fig. 1 Convergence of IKSVM-DC on four datasets

Figure 2 demonstrates the different performance between IKSVM-DC with and without a line search step on four datasets. We can see that the algorithm IKSVM-DC with a line search step would gain a smaller value of objective function during the iterations and more than two times faster than the algorithm without a line search step to obtain the same value of objective function. It illustrates that doing a line search along the descent direction at each iteration is very efficient. Moreover, the curve of value of objective function can only reveal the downward trend of the value of objective function and does not represent the convergence of the algorithm. The algorithm IKSVM-DC determines whether it converges to a local optimum by judging the value $\|d(\boldsymbol{\beta})\|^2 = \|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t\|^2$. The detailed procedure can be seen in the table Algorithm 1.

Furthermore, the computational cost of the five methods Shift, SMO-IKSVM, TDCASVM, 1-norm IKSVM and IKSVM-DC is $O(n^2)$, while the cost in other four methods is $O(n^3)$ which is caused by spectral decomposition or inversion of the kernel matrix $\boldsymbol{K} \in \mathbb{R}^{n\times n}$. Fortunately, although our method IKSVM-DC also involves spectral decomposition, only the minimum eigenvalue of the kernel matrix is necessary, and we adopt a low cost method [47] to estimate such a ρ that satisfies $\rho \geqslant -\min(\{\lambda_i\}_{i=1}^n)$ in actual implementation. Thus, IKSVM-DC is comparable to other algorithms on computational cost.

6.3 Multi-class classification results

Table 3 reports the performance of each compared algorithm on the real-world multi-class datasets, where the mean classification accuracies as well as the standard deviations of each algorithm are recorded and the best results are highlighted in bold. Furthermore, to statistically measure the significance of performance difference, pairwise t-test at 0.05 significance level is conducted between the algorithms. Specifically, when multi-class IKSVM-DC is significantly superior/inferior to the compared algorithm on any dataset, a marker •/◦ is shown. Otherwise, no marker is given [52].

According to the conclusion from Section 2, we choose the first kind of decomposition for multi-class IKSVM-DC

[8] An intensity image is a data matrix, I, whose values represent intensities within some range

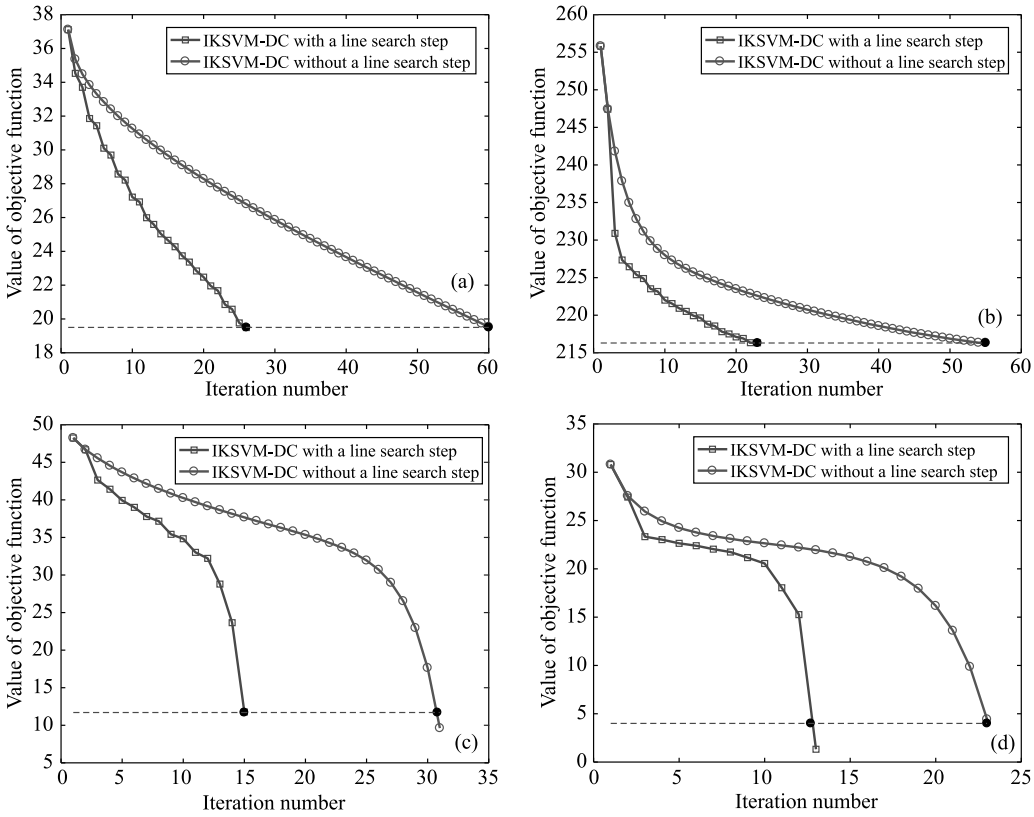


Fig. 2 Different performance between IKSVM-DC with and without a line search step on four datasets. (a) Sonar; (b) flare-solar; (c) breast-cancer; (d) thyroid

Table 3 Classification accuracy (mean±std. deviation) of each compared algorithm on several real-world multi-class datasets. In addition, •/◦ indicates whether multi-class IKSVM-DC is statistically superior/inferior to the compared algorithm on each dataset (pairwise t-test at 0.05 significance level)

	Strategy: One vs. One								Unified Form
	Clip	Flip	Shift	SMO-IKSVM	TDCASVM	IKSVM-CA	ESVM	1-norm IKSVM	IKSVM-DC
Der.	96.34%±0.066	94.12%±0.136	93.36%±0.148	62.08%±0.025•	95.32%±0.017•	98.63%±0.006	95.61%±0.087	97.15%±0.009•	**98.46%±0.006**
Eco.	72.34%±0.187•	71.11%±0.169•	67.18%±0.195•	63.63%±0.103•	66.67%±0.108•	89.13%±0.029	83.96%±0.096	88.01%±0.021•	**90.14%±0.021**
New.	90.65%±0.043•	92.24%±0.027•	83.86%±0.064•	70.14%±0.000•	86.46%±0.043•	96.67%±0.015•	97.13%±0.015•	97.85%±0.012•	**99.37%±0.011**
Gla.	53.05%±0.059•	55.33%±0.068•	53.26%±0.067•	47.84%±0.038•	50.01%±0.021•	61.35%±0.089•	68.73%±0.064	67.95%±0.052•	**72.42%±0.032**
Coi.	13.56%±0.034•	13.53%±0.033•	16.96%±0.028•	21.72%±0.078•	25.85%±0.026•	31.03%±0.017•	62.25%±0.032•	40.45%±0.035•	**67.37%±0.037**
Zon.	00.00%±0.000•	00.10%±0.001•	00.00%±0.000•	00.00%±0.000•	00.40%±0.002•	37.33%±0.021•	91.35%±0.007•	28.14%±0.029•	**92.54%±0.009**
C5.	26.55%±0.000•	26.56%±0.000•	26.52%±0.000•	26.56%±0.000•	24.93%±0.023•	28.45%±0.010•	76.17%±0.040•	64.85%±0.026•	**81.06%±0.033**
C10.	26.56%±0.000•	26.58%±0.000•	26.53%±0.000•	26.56%±0.000•	24.47%±0.044•	27.81%±0.009•	84.25%±0.024•	63.03%±0.023•	**89.46%±0.021**
C20.	26.55%±0.000•	26.56%±0.000•	26.54%±0.000•	26.54%±0.000•	20.67%±0.026•	27.13%±0.006•	87.17%±0.023•	59.56%±0.035•	**92.15%±0.028**
C30.	26.55%±0.000•	26.56%±0.000•	26.52%±0.000•	26.57%±0.000•	17.96%±0.031•	27.26%±0.007•	89.33%±0.019•	52.46%±0.028•	**92.48%±0.018**
C40.	26.57%±0.000•	26.56%±0.000•	26.57%±0.000•	26.53%±0.000•	18.47%±0.048•	26.84%±0.004•	86.57%±0.027•	50.64%±0.036•	**90.26%±0.022**
	Strategy: One vs. Rest								Unified Form
	Clip	Flip	Shift	SMO-IKSVM	TDCASVM	IKSVM-CA	ESVM	1-norm IKSVM	IKSVM-DC
Der.	94.85%±0.052•	94.26%±0.065	77.01%±0.195•	96.86%±0.007•	86.24%±0.023•	97.11%±0.007•	95.54%±0.060	95.36%±0.009•	**98.46%±0.006**
Eco.	68.65%±0.126•	67.87%±0.109•	50.11%±0.084•	69.84%±0.131•	66.16%±0.089•	79.93%±0.031•	77.47%±0.107•	83.46%±0.031•	**90.14%±0.021**
New.	76.57%±0.050•	79.49%±0.038•	73.63%±0.035•	74.98%±0.049•	79.38%±0.047•	80.83%±0.044•	93.57%±0.023•	87.93%±0.024•	**99.37%±0.011**
Gla.	44.24%±0.038•	47.22%±0.034•	41.45%±0.035•	43.18%±0.030•	47.84%±0.025•	47.42%±0.035•	65.06%±0.056•	60.97%±0.045•	**72.42%±0.032**
Coi.	25.15%±0.004•	25.02%±0.000•	26.27%±0.015•	26.33%±0.018•	25.92%±0.019•	25.05%±0.000•	55.41%±0.039•	27.07%±0.016•	**67.37%±0.037**
Zon.	10.01%±0.000•	10.00%±0.000•	10.02%±0.000•	10.01%±0.000•	10.01%±0.000•	10.00%±0.000•	61.74%±0.060•	44.67%±0.019•	**92.54%±0.009**
C5.	17.00%±0.000•	17.01%±0.000•	17.00%±0.000•	17.53%±0.010•	18.75%±0.021•	17.00%±0.000•	52.22%±0.048•	48.34%±0.042•	**81.06%±0.033**
C10.	17.04%±0.000•	17.05%±0.000•	17.02%±0.000•	17.03%±0.000•	20.86%±0.020•	17.01%±0.000•	55.75%±0.092•	50.65%±0.033•	**89.46%±0.021**
C20.	17.01%±0.000•	17.04%±0.000•	17.02%±0.000•	17.03%±0.000•	21.51%±0.000•	17.03%±0.000•	72.21%±0.035•	46.64%±0.022•	**92.15%±0.028**
C30.	17.04%±0.000•	17.03%±0.000•	17.00%±0.000•	17.52%±0.000•	21.52%±0.000•	17.25%±0.004•	80.46%±0.069•	46.92%±0.031•	**92.48%±0.018**
C40.	17.04%±0.000•	17.00%±0.000•	17.02%±0.000•	17.01%±0.000•	20.66%±0.018•	18.31%±0.011•	82.25%±0.056•	47.11%±0.024•	**90.26%±0.022**

and the classification results of all the compared algorithms are shown in Table 3. As our model is a unified form, we can directly solve multi-class problems without using strategies. For the convenience of comparison and analysis, we copied the results of multi-class IKSVM-DC twice, and compared them with the two strategies respectively. According to the results in Table 3, it is remarkable that multi-class IKSVM-DC shows great advantage to other compared algorithms with both One vs. One strategy and One vs. Rest strategy. Similar to the results in two-class classification, three spectrum transformation methods obtain the lowest classification accuracies on most of the datasets. Influenced by different multi-class strategies, SMO-IKSVM and TDCASVM algorithms can achieve slightly higher or lower classification accuracy than the three spectrum transformation methods. IKSVM-CA performs better than SMO-IKSVM and TDCASVM algorithms on most of the datasets. ESVM and 1-norm IKSVM algorithms are superior to all previous methods on most of the datasets. And ESVM is more robust than 1-norm IKSVM with both two multi-class strategies on most of the datasets. However, our unified multi-class IKSVM-DC excels ESVM in classification accuracies on all the datasets. Furthermore, we can also find that most of the compared methods achieve better results on OvO than OvR. One possible reason is that for OvR strategy it may assume one class as positive and the rest as negative. This will make the dataset imbalanced which would more likely lead to the descent of classification performance. However, OvO strategy do not suffer from this problem.

Especially, most algorithms fail on *Zongker* dataset and the classification accuracies of some algorithms (e.g., Clip, Flip, Shift, SMO-IKSVM and TDCASVM) are even worse than the results of random guess. The possible reason may be acquired that the dissimilarity between different classes are so small and most of the dissimilarity values fall in the interval $[0, 0.1]$. Therefor, it would be very hard for kernel transformation methods and some dual-based algorithms with multi-class strategies to obtain a good classification accuracies on this dataset. However, our multi-class IKSVM-DC algorithm can learn a unified classifier from these dissimilarity data as a whole and predict unseen samples with a very high accuracy. Furthermore, similar results can also be seen on several *Chickenpieces* datasets. Some algorithms (e.g., Clip, Flip, Shift, SMO-IKSVM) obtain the same classification accuracy. Actually, this means that these algorithms may not work on these dissimilarity datasets properly and the accuracies they obtained are more likely due to the error correction ability of the multi-class strategy they used. However, our multi-class IKSVM-DC algorithm is very robust for these kind of datasets and can achieve very good classification accuracies.

7 Conclusion

In this paper, we firstly present an introduction for recent researches on IKSVM problem and also analyze the dual gap problem that most existing IKSVM algorithms suffer from. Then, instead of employing the dual form of IKSVM, we propose a primal perspective for the IKSVM problem. Considering the characteristics of the spectrum for the indefinite kernel matrix, we transform the non-convex primal IKSVM model into a formulation of DC equivalently, and propose an algorithm IKSVM-DC to solve the non-convex problem efficiently. Furthermore, in order to accelerate the convergence rate of IKSVM-DC, we conduct a line search along the descent direction at each iteration. Meanwhile, we construct a unified multi-class IKSVM model for multi-class classification problems, and propose a multi-class IKSVM-DC algorithm to predict multi-class all together. Moreover, a theoretical analysis is presented to validate that our proposed IKSVM-DC algorithms can converge to a local minimum. Finally, extensive comparative experiments validate the superiority of the proposed IKSVM-DC algorithms for both binary and multi-class classification problems.

There are several directions for future study:

- **Optimization technique** In the paper, we apply DC programming to tackle the non-convex optimizations in our proposed primal IKSVM models. However, these algorithms can only arrive at local minima. How to develop better non-convex optimization techniques for our models needs more systematic research.
- **Multiple indefinite kernel scenario** In the paper, we mainly focus on the single indefinite kernel SVM problems. However, limited to the representation of single indefinite kernel, multiple indefinite kernels would provide more intelligent options and achieve better results. Thus, how to effectively utilize multiple indefinite kernels is a topic worthy of study.
- **Large-scale problem** In the experiments, we utilize the proposed primal IKSVM models in the middle-scale classification problems. However, due to the requirements of the practical applications, the large-scale learning problem has become a hot issue in machine learning. As the size of the dataset becomes larger, on the one hand, the scale of kernel matrix will increase

exponentially, then the access and calculation of data is a difficult problem. On the other hand, much more variables need to be solved, and the speed of the algorithm needs to be improved in order to satisfy the needs of practical applications. Therefore, we will explore some indefinite kernel approximation methods to tackle these problems.

Acknowledgements This work was supported by the National Key R&D Program of China (2017YFB1002801), the National Natural Science Foundations of China (Grant Nos. 61375057, 61876091 and 61403193). It was also supported by Collaborative Innovation Center of Wireless Communications Technology.

Appendixes A

Convergence Analysis of Multi-class IKSVM-DC

Proposition 3 For the sequence $\{\boldsymbol{\beta}_t\}$, we have

$$(g-h)(\boldsymbol{\beta}_t)-(g-h)(\boldsymbol{\beta}_{t+1})\geqslant\tau\|d(\boldsymbol{\beta})\|^2,$$

the equality holds if and only if $\tau\|d(\boldsymbol{\beta})\|^2=0$, where τ is a positive parameter to make functions g and h strongly convex.

Proof Firstly, we can construct the the convex functions g,h as being strongly convex with an additional term $\frac{\tau}{2}tr(\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}})$:

$$(g-h)(\boldsymbol{B})=\underbrace{\left(g(\boldsymbol{B})-\frac{\tau}{2}tr(\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}})\right)}_{G(\boldsymbol{B})}-\underbrace{\left(h(\boldsymbol{B})-\frac{\tau}{2}tr(\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}})\right)}_{H(\boldsymbol{B})}.$$

Then given the convexity of function G, we have

$$G(\boldsymbol{B}_t)\geqslant G(\boldsymbol{B}_{t+1})+tr\left(\left\langle\nabla G(\boldsymbol{B}_{t+1}),(\boldsymbol{B}_t-\boldsymbol{B}_{t+1})^{\mathrm{T}}\right\rangle\right).$$

After simplified, we get Eq. (30)[9)]

$$\begin{aligned}g(\boldsymbol{B}_t)&\geqslant g(\boldsymbol{B}_{t+1})+tr\left(\left\langle\nabla g(\boldsymbol{B}_{t+1}),(\boldsymbol{B}_t-\boldsymbol{B}_{t+1})^{\mathrm{T}}\right\rangle\right)\\&\quad+\frac{\tau}{2}(tr(\boldsymbol{B}_t\boldsymbol{B}_t^{\mathrm{T}})-2tr(\boldsymbol{B}_{t+1}\boldsymbol{B}_t^{\mathrm{T}})+tr(\boldsymbol{B}_{t+1}\boldsymbol{B}_{t+1}^{\mathrm{T}}))\\&=g(\boldsymbol{B}_{t+1})+tr\left(\left\langle\nabla g(\boldsymbol{B}_{t+1}),(\boldsymbol{B}_t-\boldsymbol{B}_{t+1})^{\mathrm{T}}\right\rangle\right)\\&\quad+\frac{\tau}{2}\|\boldsymbol{B}_t-\boldsymbol{B}_{t+1}\|_F^2.\end{aligned}\tag{30}$$

Similarly, for the function H, we can get

$$H(\boldsymbol{B}_{t+1})\geqslant H(\boldsymbol{B}_t)+tr\left(\left\langle\nabla H(\boldsymbol{B}_t),(\boldsymbol{B}_{t+1}-\boldsymbol{B}_t)^{\mathrm{T}}\right\rangle\right),$$

$$h(\boldsymbol{B}_{t+1})\geqslant h(\boldsymbol{B}_t)+tr\left(\left\langle\nabla h(\boldsymbol{B}_t),(\boldsymbol{B}_{t+1}-\boldsymbol{B}_t)^{\mathrm{T}}\right\rangle\right)+\frac{\tau}{2}\|\boldsymbol{B}_{t+1}-\boldsymbol{B}_t\|_F^2.\tag{31}$$

Since $\boldsymbol{B}_{t+1}$ is a unique solution of the convex problem in Eq. (21) of the main paper, we have

$$\nabla g(\boldsymbol{B}_{t+1})=\boldsymbol{\Theta}_t=\nabla h(\boldsymbol{B}_t).\tag{32}$$

Combining Eqs. (30), (31) and (32), we have

$$(g(\boldsymbol{B}_t)-h(\boldsymbol{B}_t))-(g(\boldsymbol{B}_{t+1})-h(\boldsymbol{B}_{t+1}))\geqslant\tau\|\boldsymbol{B}_{t+1}-\boldsymbol{B}_t\|_F^2.$$

Proposition 4 For the sequence $\{\boldsymbol{\beta}_t\}$, we have

$$\langle\nabla(g-h)(\boldsymbol{\beta}_{t+1}),\boldsymbol{\beta}_{t+1}-\boldsymbol{\beta}_t\rangle\leqslant 0,$$

that is, $d(\boldsymbol{\beta})=\boldsymbol{\beta}_{t+1}-\boldsymbol{\beta}_t$ is a descent direction for $f=g-h$ at $\boldsymbol{\beta}_{t+1}$.

Proof Following Proposition 3, we have

$$h(\boldsymbol{B}_t)\geqslant h(\boldsymbol{B}_{t+1})+tr\left(\left\langle\nabla h(\boldsymbol{B}_{t+1}),(\boldsymbol{B}_t-\boldsymbol{B}_{t+1})^{\mathrm{T}}\right\rangle\right)+\frac{\tau}{2}\|\boldsymbol{B}_t-\boldsymbol{B}_{t+1}\|_F^2.\tag{33}$$

Since each class is independent from each other for the multi-class problem, we can simplify Eq. (33) into the following formulation

$$h(\boldsymbol{B}_t^i)\geqslant h(\boldsymbol{B}_{t+1}^i)+\left\langle\nabla h(\boldsymbol{B}_{t+1}^i),(\boldsymbol{B}_t^i-\boldsymbol{B}_{t+1}^i)^{\mathrm{T}}\right\rangle+\frac{\tau}{2}\|\boldsymbol{B}_t^i-\boldsymbol{B}_{t+1}^i\|^2.\tag{34}$$

Given the Eq. (34), we can know that the function $h(\boldsymbol{B}_t^i)$ is strongly convex on $\mathbb{R}^n$. Then according to the Theorem 2.1.9 in [41], we have

$$\langle\nabla h(\boldsymbol{B}_t^i)-\nabla h(\boldsymbol{B}_{t+1}^i),(\boldsymbol{B}_t^i-\boldsymbol{B}_{t+1}^i)\rangle\geqslant\tau\|(\boldsymbol{B}_t^i-\boldsymbol{B}_{t+1}^i)\|^2.$$

Combining Eq. (32), we have

$$\langle\nabla g(\boldsymbol{B}_{t+1}^i)-\nabla h(\boldsymbol{B}_{t+1}^i),(\boldsymbol{B}_{t+1}^i-\boldsymbol{B}_t^i)\rangle\leqslant-\tau d(\boldsymbol{B}^i)^2\leqslant 0,$$

the equality holds if and only if $\tau d(\boldsymbol{B}^i)^2=0$.

Theorem 2 If the sequence $\{\boldsymbol{\beta}_t\}$ satisfies $d(\boldsymbol{\beta})=\boldsymbol{\beta}_{t+1}-\boldsymbol{\beta}_t=0$, let $\boldsymbol{\beta}^*=\boldsymbol{\beta}_{t+1}=\boldsymbol{\beta}_t$ and $\mathcal{U}$ be a neighbourhood of $\boldsymbol{\beta}^*$. For $\forall\boldsymbol{\beta}\in\mathcal{U}$, we have

$$g(\boldsymbol{\beta})-h(\boldsymbol{\beta})\geqslant g(\boldsymbol{\beta}^*)-h(\boldsymbol{\beta}^*).$$

Proof Following Eq. (32), the condition $d(\boldsymbol{B})=\boldsymbol{B}_{t+1}-\boldsymbol{B}_t=0$ implies $\nabla g(\boldsymbol{B}^*)=\nabla g(\boldsymbol{B}_{t+1})=\boldsymbol{\Theta}_t$, that is, $\exists\boldsymbol{\Theta}\in\partial g(\boldsymbol{B}^*)$. So the conjugate function of g at $\boldsymbol{B}^*$ is

$$g^*(\boldsymbol{\Theta})=\sup\{tr(\langle\boldsymbol{B}^*,\boldsymbol{\Theta}\rangle)-g(\boldsymbol{B}^*)\}=tr(\langle\boldsymbol{B}^*,\boldsymbol{\Theta}\rangle)-g(\boldsymbol{B}^*),\tag{35}$$

[9)] The equation always holds and we can derive from a simplified form. Let $A=\begin{bmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{bmatrix}$ and $B=\begin{bmatrix}b_{11}&b_{12}\\b_{21}&b_{22}\end{bmatrix}$. Then we have $tr(AA^{\mathrm{T}})=a_{11}^2+a_{12}^2+a_{21}^2+a_{22}^2$, $tr(BB^{\mathrm{T}})=b_{11}^2+b_{12}^2+b_{21}^2+b_{22}^2$ and $tr(AB^{\mathrm{T}})=a_{11}b_{11}+a_{12}b_{12}+a_{21}b_{21}+a_{22}b_{22}$. Therefore, we can derive the following equation that $tr(AA^{\mathrm{T}})-2tr(AB^{\mathrm{T}})+tr(BB^{\mathrm{T}})=\sum_{i,j=1}^2(a_{ij}-b_{ij})^2=\|A-B\|_F^2$

and $\forall \boldsymbol{\Theta} \in \mathbb{R}^{c\times n}$, the conjugate function of h at $\boldsymbol{B}^*$ is

$$h^*(\boldsymbol{\Theta}) = \sup\{tr(\langle \boldsymbol{B}^*, \boldsymbol{\Theta}\rangle) - h(\boldsymbol{B}^*)\} \geqslant tr(\langle \boldsymbol{B}^*, \boldsymbol{\Theta}\rangle) - h(\boldsymbol{B}^*)\,. \tag{36}$$

Combining Eqs. (35) and (36), we have

$$g(\boldsymbol{B}^*) + g^*(\boldsymbol{\Theta}) = tr(\langle \boldsymbol{B}^*, \boldsymbol{\Theta}\rangle) \leqslant h(\boldsymbol{B}^*) + h^*(\boldsymbol{\Theta})\,. \tag{37}$$

On the other hand, since $\boldsymbol{\Theta} = \nabla h(\boldsymbol{B})$, it means $\exists \boldsymbol{\Theta} \in \partial h(\boldsymbol{B})$. Similar to the process in Eqs. (35), (36) and (37), we have

$$h(\boldsymbol{B}) + h^*(\boldsymbol{\Theta}) = tr(\langle \boldsymbol{B}, \boldsymbol{\Theta}\rangle) \leqslant g(\boldsymbol{B}) + g^*(\boldsymbol{\Theta})\,. \tag{38}$$

Combining Eqs. (37) and (36), we can reach the conclusion.

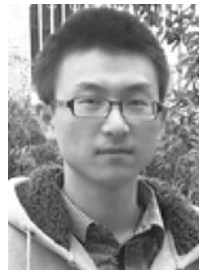
References

1. Cortes C. Support-vector network. Machine Learning Journal, 1995, 20(3): 273–297
2. Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press, 2000
3. Li Y Y, Lu R Q. Locality preserving projection on SPD matrix Lie group: algorithm and analysis. Science China Information Sciences, 2018, 61(9): 092104
4. Ma L R, Song D D, Liao L J, Wang J. PSVM: a preference-enhanced SVM model using preference data for classification. Science China Information Sciences, 2017, 60(12): 122103
5. Saigo H, Vert J P, Ueda N, Akutsu T. Protein homology detection using string alignment kernels. Bioinformatics, 2004, 20(11): 1682–1689
6. Wang C, Song Y, Li H, Zhang M, Han J. Text classification with heterogeneous information network kernels. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence. 2016, 2130–2136
7. Vapnik V. The Nature of Statistical Learning Theory. Germany: Springer Science & Business Media, 2013
8. Suard F, Rakotomamonjy A, Bensrhair A. Kernel on bag of paths for measuring similarity of shapes. In: Proceedings of European Symposium on Artificial Neural Networks. 2007, 355–360
9. Chen Y, Garcia E, Gupta M, Rahimi A, Cazzanti L. Similarity-based classification: concepts and algorithms. Journal of Machine Learning Research, 2009, 10(3): 747–776
10. Chen Y, Gupta M. Fusing similarities and kernels for classification, In: Proceedings of the 12th International Conference on IEEE Information Fusion. 2009, 474–481
11. Pkalska E, Haasdonk B. Kernel discriminant analysis for positive definite and indefinite kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(6): 1017–1032
12. Sun H, Wu Q. Least square regression with indefinite kernels and coefficient regularization. Applied and Computational Harmonic Analysis, 2011, 30(1): 96–109
13. Ying Y, Campbell C, Girolami M. Analysis of SVM with indefinite kernels. In: Proceedings of the 22nd International Conference on Neural Information Processing Systems. 2009, 2205–2213
14. Loosli G, Canu S. Non positive SVM. Technical Report, 2010
15. Haasdonk B. Feature space interpretation of SVMs with indefinite kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(4): 482–492
16. Pekalska E, Paclik P, Duin R P. A generalized kernel approach to dissimilarity-based classification. Journal of Machine Learning Research, 2001, 2(12): 175–211
17. Graepel T, Herbrich R, Bollmann-Sdorra P, Obermayer K. Classification on pairwise proximity data. In: Proceedings of the 13th Conference on Neural Information Processing Systems. 1999, 438–444
18. Roth V, Laub J, Kawanabe M, Buhmann J M. Optimal cluster preserving embedding of nonmetric proximity data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(12): 1540–1551
19. Luss R, d'Aspremont A. Support vector machine classification with indefinite kernels. In: Proceedings of the 21st Conference on Neural Information Processing Systems. 2008, 953–960
20. Chen J, Ye J. Training SVM with indefinite kernels. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 136–143
21. Chen Y, Gupta M R, Recht B. Learning kernels from indefinite similarities. In: Proceedings of the 26th International Conference on Machine Learning. 2009, 145–152
22. Gu S, Guo Y. Learning SVM classifiers with indefinite kernels. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence. 2012, 942–948
23. Lin H T, Lin C J. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Neural Computation, 2003, 3: 1–32
24. Akoa F B. Combining DC algorithms (DCAs) and decomposition techniques for the training of nonpositive-semidefinite kernels. IEEE Transactions on Neural Networks, 2008, 19(11): 1854–1872
25. Ong C S, Mary X, Canu S, Smola A J. Learning with non-positive kernels. In: Proceedings of the 21st International Conference on Machine Learning. 2004, 81
26. Loosli G, Canu S, Ong C S. Learning SVM in Kreĭn spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(6): 1204–1216
27. Alabdulmohsin I M, Gao X, Zhang X. Support vector machines with indefinite kernels. In: Proceedings of the 6th Asian Conference on Machine Learning. 2014, 32–47
28. Kotsiantis S B, Zaharakis I, Pintelas P. Supervised machine learning: a review of classification techniques. Emerging Artificial Intelligence Applications in Computer Engineering, 2007, 160: 3–24
29. Friedman J. Another approach to polychotomous classification. Technical Report, Department of Statistics, Stanford University, 1996
30. Krebel U. Pairwise classification and support vector machines. Advances in Kernel Methods: Support Vector Learning, 1999, 255–268
31. Bottou L, Cortes C, Denker J S, Drucker H, Guyon Z, Jackel L D, Le Cun Y, Muller U A, Sackinger E, Simard P, Vapnik U. Comparison of classifier methods: a case study in handwritten digit recognition. In: Proceedings of the 12th IAPR International Conference on Pattern Recognition. 1994, 77–82
32. Dietterich T G, Bakiri G. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Re-

search, 1995, 2: 263–286

33. Allwein E L, Schapire R E, Singer Y. Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research, 2000, 1(12): 113–141
34. Platt J C, Cristianini N, Shawe-Taylor J. Large margin dags for multiclass classification. In: Proceedings of the 12th International Conference on Neural Information Processing Systems. 1999, 547–553
35. Scholkopf B, Smola A J. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Massackusetts: MIT Press, 2001
36. Tao P D, An L T H. Convex analysis approach to DC programming: theory, algorithms and applications. Acta Mathematica Vietnamica, 1997, 22(1): 289–355
37. Dinh T P, Le Thi H A. Recent advances in DC programming and DCA. In: Nguyen N T, Le-Thi H A, eds. Transactions on Computational Intelligence XIII. Springer, Berlin, Heidelbery, 2014, 1–37
38. Piot B, Geist M, Pietquin O. Difference of convex functions programming for reinforcement learning. In: Proceedings of the 27th Conference on Neural Information Processing Systems. 2014, 2519–2527
39. Xu H M, Xue H, Chen X H, Wang Y Y. Solving indefinite kernel support vector machine with difference of convex functions programming. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence. 2017, 2782–2788
40. Artacho F J A, Fleming R M, Vuong P T. Accelerating the DC algorithm for smooth functions. Mathematical Programming, 2018, 169(1): 95–118
41. Nesterov Y. Introductory Lectures on Convex Optimization: A Basic Course. Germany: Springer Science & Business Media, 2013
42. Wang Z, Xue X. Multi-class support vector machine. In: Ma Y, Guo G, eds. Support Vector Machines Applications. Springer, Cham, 2014, 23–48
43. Vapnik V N. Statistical Learning Theory. Wiley New York: John Wiley & Sons, Inc., 1998
44. Weston J, Watkins C. Multi-class support vector machines. Technical Report, Department of Computer Science, Royal Holloway, University of London, 1998
45. Crammer K, Singer Y. On the learnability and design of output codes for multi-class problems. Machine Learning, 2002, 47(2): 201–233
46. Blake C, Merz C J. UCI repository of machine learning databases, 1998
47. Ratsch G, Onoda T, Muller K R. Soft margins for AdaBoost. Machine Learning, 2001, 42(3): 287–320
48. Duin R P, Pekalska E. The dissimilarity representation for pattern recognition: a tutorial. Technical Report, 2009
49. Mosek A. The MOSEK optimization software. Google Website, 2010
50. Wu G, Chang E Y, Zhang Z. An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines. In: Proceedings of the 22nd International Conference on Machine Learning. 2005, 1245–1256
51. Chang C C, Lin C J. LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3): 27
52. Zhang M L, Zhou Z H. Exploiting unlabeled data to enhance ensemble diversity. Data Mining and Knowledge Discovery, 2013, 26(1): 98–129

Hui Xue received the BSc degree in Mathematics from Nanjing Normal University, China in 2002. In 2005, she received the MSc degree in Mathematics from Nanjing University of Aeronautics & Astronautics (NUAA), China. And she also received the PhD degree in Computer Application Technology at NUAA, China in 2008. Since 2009, as an Associate Professor, she has been with the School of Computer Science and Engineering at Southeast University, China. Her research interests include pattern recognition and machine learning.



Haiming Xu received the BS degree in School of Computer Science and Technology from China University of Mining and Techology, China in 2015. In 2018, he received the MSc degree in the School of Computer Science and Engineering at Southeast University, China. Now, he is a PhD candidate at the University of Adelaide, Australia. His research interests include pattern recognition, machine learning, and data mining.

Xiaohong Chen received the BSc degree in Mathematics from Qufu Normal University, China in 1998. In 2001, she received the MSc degree in Mathematics from Nanjing University of Aeronautics & Astronautics (NUAA), China. And she also received the PhD degree in Computer Application Technology at NUAA, China in 2011. She is an Associate Professor at the College of Science at NUAA, China. Her research interests include pattern recognition and machine learning.

Yunyun Wang received the BS degree in computer science and technology from Anhui Normal University, China in 2006, and the PhD degree in computer science and engineering from the Nanjing University of Aeronautics and Astronautics, China in 2012. She is currently an associate professor with the Department of Computer Science and Engineering, Nanjing University of Posts & Telecommunications, China. Her current research interests include pattern recognition and machine learning.