Solving Indefinite Kernel Support Vector Machine with Difference of Convex Functions Programming

Hai-Ming Xu[1,2] and **Hui Xue**[1,2,*] and **Xiao-Hong Chen**[3] and **Yun-Yun Wang**[4]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, 210096, China
[2]MOE Key Laboratory of Computer Network and Information Integration (Southeast University), 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
{heimingx, hxue}@seu.edu.cn, lyandcxh@nuaa.edu.cn, wangyunyun@njupt.edu.cn

Abstract

Indefinite kernel support vector machine (IKSVM) has recently attracted increasing attentions in machine learning. Different from traditional SVMs, IKSVM essentially is a non-convex optimization problem. Some algorithms directly change the spectrum of the indefinite kernel matrix at the cost of losing some valuable information involved in the kernels so as to transform the non-convex problem into a convex one. Other algorithms aim to solve the dual form of IKSVM, but suffer from the dual gap between the primal and dual problems in the case of indefinite kernels. In this paper, we directly focus on the non-convex primal form of IKSVM and propose a novel algorithm termed as IKSVM-DC. According to the characteristics of the spectrum for the indefinite kernel matrix, IKSVM-DC decomposes the objective function into the subtraction of two convex functions and thus reformulates the primal problem as a difference of convex functions (DC) programming which can be optimized by the DC algorithm (DCA). In order to accelerate convergence rate, IKSVM-DC further combines the classical DCA with a line search step along the descent direction at each iteration. A theoretical analysis is then presented to validate that IKSVM-DC can converge to a local minimum. Systematical experiments on real-world datasets demonstrate the superiority of IKSVM-DC compared to state-of-the-art IKSVM related algorithms.

1 Introduction

Support vector machines (SVM) 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) (Cristianini and Shawe-Taylor 2000). 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 for 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 (Saigo et al. 2004). The weighted meta-path based similarity matrices for text classification in natural language processing are frequently indefinite (Wang et al. 2016). The sigmoid kernels in neural networks with various values of the hyper-parameters are also mostly indefinite (Vapnik 2013). As a result, indefinite kernels have become increasingly important in kernel methods and indefinite kernel SVM (IKSVM) has attracted more and more attentions in machine learning. 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 (Pekalska, Paclik, and Duin 2001), "Flip" flips the sign of the negative eigenvalues (Graepel et al. 1999), and "Shift" shifts all the eigenvalues by a positive constant (Roth et al. 2003). 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 (Luss and d'Aspremont 2008). Chen and Ye reformulated the formulation into a semi-infinite quadratically constrained linear programming and proposed a faster algorithm (Chen and Ye 2008). Chen et al. further introduced a primal model to avoid over-fitting (Chen, Gupta, and Recht 2009). Gu and Guo incorporated the kernel principal component analysis into the SVM classification and naturally generated a surrogate PSD kernel (Gu and Guo 2012). 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 (Lin and Lin 2003). Akoa incorporated difference of convex functions programming into decomposition methods to tackle

*Corresponding author.

IKSVM problems and obtained a stationary point as a solution (Akoa 2008). 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 decomposing the indefinite kernel into the summation of two PSD kernels (Ong et al. 2004; Loosli, Canu, and Ong 2016). Alabdulmohsin et al. transferred the indefinite kernel matrix into an affine constraint so that the non-convex problem was converted into a linear programming (Alabdulmohsin, Gao, and Zhang 2014). 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.

In this paper, we directly focus on the non-convex primal form of IKSVM and propose a novel algorithm named IKSVM-DC. The algorithm firstly constructs the primal problem as a difference of convex functions (DC) programming equivalently, and then iteratively optimizes it by the DC algorithm (DCA). Furthermore, for speeding convergence rate, IKSVM-DC adopts a line search along the descent direction under the Armijo type rule at each iteration in classical DCA. A theoretical analysis is finally presented to validate that IKSVM-DC can converge to a local minimum. Experiments conducted on several real-world datasets demonstrate that IKSVM-DC has not only much better classification accuracy compared to some IKSVM related algorithms, but also nearly three times higher convergence rate than the classical DCA.

2 Related Work

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:

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

and the associated kernelized dual problem is

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

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

$$d^* \leq p^*.$$

The equality holds if and only if the kernel matrix generated from $K(\cdot, \cdot)$ is PSD (Cristianini and Shawe-Taylor 2000). When the kernels become indefinite, the equality would never hold and thus a dual gap exists between the primal and dual problems.

However, many IKSVM algorithms still emphasize on the dual problem. For example, proxy kernel algorithms obtained a surrogate PSD kernel matrix for the indefinite kernel directly based on the dual form of IKSVM (Luss and d'Aspremont 2008; Chen and Ye 2008; Chen, Gupta, and Recht 2009; Gu and Guo 2012). SMO-type algorithm proposed an improved SMO method to solve the non-convex dual form of IKSVM (Lin and Lin 2003). Akoa utilized difference of convex functions programming to solve non-convex problems in decomposition methods, but the decomposition methods are based on the dual form of IKSVM (Akoa 2008). In order to avoid suffering from the dual gap, we will directly focus on the primal form of IKSVM in this paper.

3 Primal IKSVM Model

The primal 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:

$$\min_{\boldsymbol{w},b} \quad \gamma \langle \boldsymbol{w}, \boldsymbol{w} \rangle + \sum_{i=1}^{n} V(y_i, \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b), \tag{4}$$

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

When the kernel is indefinite, we can solve Eq. (4) in a wider RKKS $\mathcal{K}$ as

$$\min_{\boldsymbol{f} \in \mathcal{K}, b} \quad \gamma \langle \boldsymbol{f}, \boldsymbol{f} \rangle_{\mathcal{K}} + \sum_{i=1}^{n} V(y_i, \boldsymbol{f}(\boldsymbol{x}_i) + b). \tag{5}$$

In RKKS, Ong et al. have verified that the Representer Theorem still holds (Ong et al. 2004) 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, considering the Representer Theorem in RKKS, the primal model of IKSVM in Eq. (5) can be further expressed as

$$\min_{\boldsymbol{\beta},b} \quad \gamma \boldsymbol{\beta}^T \boldsymbol{K} \boldsymbol{\beta} + \sum_{i=1}^{n} V(y_i, \boldsymbol{K}^i \boldsymbol{\beta} + b), \tag{6}$$

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. (6), 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 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. (6) after adding the scaling constant $1/2$ becomes

$$\min_{\boldsymbol{\beta},b} \frac{1}{2}\left[\gamma\boldsymbol{\beta}^T\boldsymbol{K}\boldsymbol{\beta} + \sum_{i=1}^{n}\max\left(0, 1 - y_i(\boldsymbol{K}^i\boldsymbol{\beta} + b)\right)^2\right]. \tag{7}$$

Although much similar to the traditional primal PSD kernel SVM, Eq. (7) is actually an unconstrained non-convex optimization which has become an NP-hard problem in terms of indefinite kernels.

4 IKSVM with DC

In this section, we further characterize the primal IKSVM into a DC problem and then propose a novel algorithm to solve it.

4.1 DC Programming

DC programming (Tao and An 1997; Dinh and Le Thi 2014) is a powerful tool 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

$$f(\boldsymbol{\omega}) = g(\boldsymbol{\omega}) - h(\boldsymbol{\omega}), \tag{8}$$

where the variable $\boldsymbol{\omega} \in \mathbb{R}^n$. The two functions g, h are convex and lower semi-continuous on $\mathbb{R}^n$. Let $h^*(\boldsymbol{\psi}) = \sup\{\langle\boldsymbol{\omega}, \boldsymbol{\psi}\rangle - h(\boldsymbol{\omega}), \boldsymbol{\omega} \in \mathbb{R}^n\}$ be the conjugate function of h. The dual problem of Eq. (8) can be described as

$$f^*(\boldsymbol{\psi}) = h^*(\boldsymbol{\psi}) - g^*(\boldsymbol{\psi}), \tag{9}$$

where the conjugate variable $\boldsymbol{\psi} \in \mathbb{R}^n$. Due to the property of conjugate dual, Eqs. (8) and (9) are equal to each other. The variables $\boldsymbol{\omega}$ and $\boldsymbol{\psi}$ satisfy

$$\boldsymbol{\psi} \in \partial h(\boldsymbol{\omega}), \quad \boldsymbol{\omega} \in \partial g^*(\boldsymbol{\psi}), \tag{10}$$

where ∂h and ∂g^* denote the sub-gradients of h and g^* respectively. DC algorithm (DCA) further utilizes Eq. (10) 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 (Piot, Geist, and Pietquin 2014) : (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 5.1.

4.2 IKSVM Converted into a DC Problem

IKSVM 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

$$f(\boldsymbol{\beta}) = \frac{1}{2}\left[\gamma\boldsymbol{\beta}^T\boldsymbol{K}\boldsymbol{\beta} + \sum_{i=1}^{n}\max\left(0, 1 - y_i(\boldsymbol{K}^i\boldsymbol{\beta} + b)\right)^2\right], \tag{11}$$

and the eigenspectrum of the indefinite kernel matrix can be depicted as $\boldsymbol{K} = U^T\Lambda U$, where U and Λ represent the orthonormal column eigenvector matrix and the diagonal eigenvalue matrix respectively, and Λ consists of both positive and negative eigenvalues. Then, we can easily get several equivalent decompositions on Eq. (11) 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

$$\textcircled{1}\begin{cases} g(\boldsymbol{\beta}) = \frac{1}{2}\left[\gamma\boldsymbol{\beta}^T U^T(\rho_1 I + \Lambda)U\boldsymbol{\beta} + V\right] \\ h(\boldsymbol{\beta}) = \frac{1}{2}\gamma\boldsymbol{\beta}^T U^T(\rho_1 I)U\boldsymbol{\beta}, \end{cases} \tag{12}$$

$$\textcircled{2}\begin{cases} g(\boldsymbol{\beta}) = \frac{1}{2}\left[\gamma\boldsymbol{\beta}^T U^T(\rho_2 I)U\boldsymbol{\beta} + V\right] \\ h(\boldsymbol{\beta}) = \frac{1}{2}\gamma\boldsymbol{\beta}^T U^T(\rho_2 I - \Lambda)U\boldsymbol{\beta}, \end{cases}$$

where $V = \sum_{i=1}^{n}\max(0, 1 - y_i(\boldsymbol{K}^i\boldsymbol{\beta} + b))^2$. 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. $\rho_1 \geq -\min(\{\lambda_i\}_{i=1}^n)$ and $\rho_2 \geq \max(\{\lambda_i\}_{i=1}^n)$, 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. (10), we have

$$\boldsymbol{\theta} \in \partial h(\boldsymbol{\beta}), \quad \boldsymbol{\beta} \in \partial g^*(\boldsymbol{\theta}). \tag{13}$$

Utilizing Eq. (13), we can approximate the function h with its affine minorization at point $\boldsymbol{\beta}_t$

$$h(\boldsymbol{\beta}) = h(\boldsymbol{\beta}_t) + \langle\boldsymbol{\beta} - \boldsymbol{\beta}_t, \boldsymbol{\theta}_t\rangle, \tag{14}$$

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

$$g^*(\boldsymbol{\theta}) = g^*(\boldsymbol{\theta}_t) + \langle\boldsymbol{\theta} - \boldsymbol{\theta}_t, \boldsymbol{\beta}_{t+1}\rangle, \tag{15}$$

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. (14) and (15).

We further construct two sequences $\{\boldsymbol{\beta}_t\}$ and $\{\boldsymbol{\theta}_t\}$ for solutions by solving Eq. (16) alternately

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

Following (Dinh and Le Thi 2014), we omit the conjugate dual problem with a simplified form $\boldsymbol{\theta}_t \in \partial h(\boldsymbol{\beta}_t)$ in practice,

Algorithm 1 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. (17) 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** $\|d(\boldsymbol{\beta})\|^2 \leq \delta$ **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** $y^* = \text{sign}(K(\boldsymbol{x}^*, \boldsymbol{x})\boldsymbol{\beta} + b)$;

and obtain

$$\begin{cases} \boldsymbol{\theta}_t & \in \partial h(\boldsymbol{\beta}_t) \\ \boldsymbol{\beta}_{t+1} & \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^n} g(\boldsymbol{\beta}) - \langle \boldsymbol{\beta}, \boldsymbol{\theta}_t \rangle. \end{cases} \tag{17}$$

The sequence $\{\boldsymbol{\beta}_t\}$ can generate a descent direction at each iteration. In order to accelerate the convergence rate, we can 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 (Artacho, Fleming, and Vuong 2015)

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

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

4.3 Convergence Analysis

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

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

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

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 $\frac{\tau}{2}\boldsymbol{\beta}^2$:

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

Then given the convexity of function G, we have

$$G(\boldsymbol{\beta}_t) \geq G(\boldsymbol{\beta}_{t+1}) + \nabla G(\boldsymbol{\beta}_{t+1})(\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1}).$$

After simplified, we get

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

Similarly, for the function H, we can get

$$H(\boldsymbol{\beta}_{t+1}) \geq H(\boldsymbol{\beta}_t) + \nabla H(\boldsymbol{\beta}_t)(\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t),$$

$$h(\boldsymbol{\beta}_{t+1}) \geq h(\boldsymbol{\beta}_t) + \langle \nabla h(\boldsymbol{\beta}_t), \boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t \rangle + \frac{\tau}{2}\|\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t\|^2. \tag{19}$$

Since $\boldsymbol{\beta}_{t+1}$ is a unique solution of the convex problem in Eq. (17), we have

$$\nabla g(\boldsymbol{\beta}_{t+1}) = \boldsymbol{\theta}_t = \nabla h(\boldsymbol{\beta}_t). \tag{20}$$

Combining Eqs. (18), (19) and (20), we have

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

□

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. 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*

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

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

$$h(\boldsymbol{\beta}_t) \geq h(\boldsymbol{\beta}_{t+1}) + \langle \nabla h(\boldsymbol{\beta}_{t+1}), \boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1} \rangle + \frac{\tau}{2}\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1}\|^2. \tag{21}$$

Then the derivation of Eq. (21) at $\boldsymbol{\beta}_t$ yields

$$\nabla h(\boldsymbol{\beta}_t) - \nabla h(\boldsymbol{\beta}_{t+1}) \geq \tau\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t+1}\|.$$

Further, we get

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

Combining Eq. (20), we have

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

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 $\{\beta_t\}$ 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*

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

Proof. Following Eq. (20), 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

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

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

$$h^*(\boldsymbol{\theta}) = \sup\{\langle \boldsymbol{\beta}^*, \boldsymbol{\theta}\rangle - h(\boldsymbol{\beta}^*)\} \geq \langle \boldsymbol{\beta}^*, \boldsymbol{\theta}\rangle - h(\boldsymbol{\beta}^*). \quad (23)$$

Combining Eqs. (22) and (23), we have

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

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. (22), (23) and (24), we have

$$h(\boldsymbol{\beta}) + h^*(\boldsymbol{\theta}) = \langle \boldsymbol{\beta}, \boldsymbol{\theta}\rangle \leq g(\boldsymbol{\beta}) + g^*(\boldsymbol{\theta}). \quad (25)$$

Combining Eqs. (24) and (25), we can reach the conclusion. □

5 Experiments

In this section, we experimentally evaluate the performance of the proposed algorithm IKSVM-DC compared with several related algorithms using a collection of datasets on the benchmark.

5.1 Experimental Setup

In the experiments, ten real-world datasets are used for learning IKSVMs, including two datasets $Ionosphere$ and $Sonar$ from UCI Machine Learning Repository (Blake and Merz 1998), four datasets $Titanic$, $Breast-cancer$, $Thyroid$ and $Flare-solar$ from IDA database (Rätsch, Onoda, and Müller 2001), and the rest four dissimilarity datasets are $Balls3D$, $WoodyPlants50$, $CoilYork$ and $Zongker$ provided by the Pattern Recognition Lab of Delft University of Technology (Duin 2000). Table 1 lists a brief description of these ten 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 the four 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 $\{2^{-6}, 2^{-5}, \cdots, 2^5, 2^6\}$.

As IKSVM-DC is a quadratic programming without constraints, we utilize the interior-point optimizer to solve it by Mosek optimization software (Mosek 2010). Moreover, since the variable $\boldsymbol{\beta} \in \mathbb{R}^n$ can be negative, we randomly initialize $\boldsymbol{\beta}_0 \in [-1, +1]$. 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. (12).

Table 1: Datasets description.

dataset(abbreviation)	#num(#class)	ϕ[1]	measure
Ionosphere(Ion.)	351(2)	0.340	sigmoid kernel
Sonar(Son.)	208(2)	0.290	sigmoid kernel
Titanic(Tit.)	2201(2)	0.261	sigmoid kernel
Breast-cancer(Bre.)	277(2)	0.718	sigmoid kernel
Thyroid(Thy.)	215(2)	0.470	sigmoid kernel
Flare-solar(Fla.)	1066(2)	0.211	sigmoid kernel
Balls3D(Bal.)	200(2)	0.500	distance on 3-D balls
WoodyPlants50(Woo.)	791(14)	0.500	leaves shape matching
CoilYork(Coi.)	288(4)	0.500	graph matching
Zongker(Zon.)	2000(10)	0.120	handwritten digits matching

We compare IKSVM-DC with several state-of-the-art IKSVM algorithms including:

- "Clip", "Flip" and "Shift" (Wu, Chang, and Zhang 2005): three methods directly change the eigenspectrum to obtain a PSD kernel matrix, and take the modified PSD kernel into a dual form of SVM.
- SMO-IKSVM (Lin and Lin 2003): a method utilizes the SMO-type algorithm to solve the dual form of IKSVM.
- TDCASVM (Akoa 2008): a method uses DC algorithm to solve non-convex dual problems in decomposition methods.
- IKSVM-CA (Gu and Guo 2012): 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 (Loosli, Canu, and Ong 2016): a method transforms the indefinite kernel from Kreĭn spaces into Hilbert spaces, and trains the convex dual form of SVM.
- 1-norm IKSVM (Alabdulmohsin, Gao, and Zhang 2014): 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 (Mosek 2010).

The dual problem of SVM/IKSVM in the algorithms above is all solved by the LIBSVM library (Chang and Lin 2011).

5.2 Experimental Results

Table 2 reports the performance of each compared algorithm on the 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

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

Table 2: Classification accuracy (mean±std. deviation) of each compared algorithm on several real-world 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.	0.737±0.104•	0.759±0.086•	0.677±0.055•	0.731±0.108•	0.749±0.047•	0.865±0.057•	0.886±0.020•	0.919±0.016•	**0.936±0.011**
Son.	0.676±0.062•	0.689±0.017•	0.658±0.047•	0.649±0.068•	0.638±0.072•	0.758±0.030•	0.734±0.027•	0.792±0.030•	**0.848±0.023**
Tit.	0.736±0.068•	0.774±0.009•	0.717±0.071•	0.744±0.051•	0.736±0.043•	0.788±0.005	0.788±0.005	0.787±0.005•	**0.791±0.004**
Bre.	0.731±0.022•	0.736±0.023•	0.713±0.007•	0.727±0.020•	0.741±0.022•	0.375±0.395•	0.734±0.027•	0.738±0.026•	**0.783±0.015**
Thy.	0.899±0.039•	0.921±0.036•	0.757±0.074•	0.872±0.041•	0.877±0.057•	0.940±0.025•	0.927±0.051•	0.941±0.034•	**0.977±0.019**
Fla.	0.604±0.052•	0.589±0.050•	0.553±0.000•	0.588±0.049•	0.569±0.026•	0.664±0.039	0.632±0.055•	0.623±0.059•	**0.681±0.013**
Bal.	0.478±0.055•	0.471±0.031•	0.482±0.053•	0.499±0.035•	0.558±0.016•	0.513±0.040•	0.536±0.029•	0.546±0.044•	**0.570±0.031**
Woo.	0.263±0.043•	0.183±0.022•	0.356±0.074•	0.331±0.035•	0.499±0.001•	0.574±0.021•	0.923±0.012	0.721±0.019•	**0.924±0.010**
Coi.	0.293±0.028•	0.258±0.018•	0.319±0.021•	0.485±0.048•	0.480±0.047•	0.584±0.037•	0.638±0.054•	0.669±0.021•	**0.731±0.044**
Zon.	0.641±0.029•	0.645±0.023•	0.641±0.018•	0.645±0.048•	0.582±0.058•	0.558±0.023•	0.662±0.059•	0.622±0.019•	**0.818±0.033**

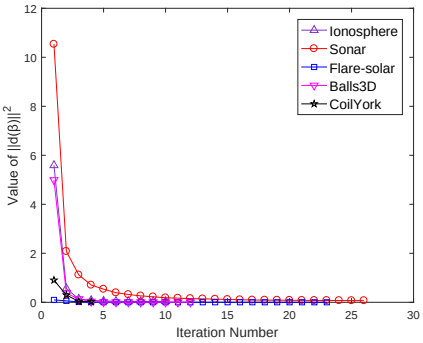

Figure 1: Convergence of IKSVM-DC on five datasets.

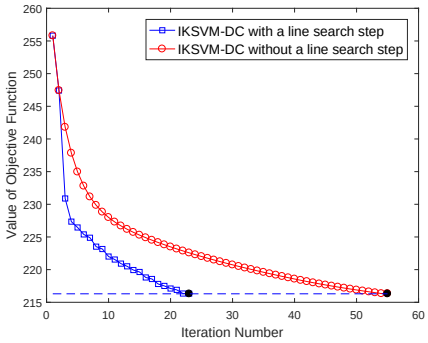

Figure 2: Different performance between IKSVM-DC with and without a line search step on the dataset $Flare-solar$.

compared algorithm on any dataset, a marker •/◦ is shown. Otherwise, no marker is given (Zhang and Zhou 2013).

We conduct experiments on the two kinds of decompositions, and the classification accuracies of these two decompositions are comparable which means 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 algorithms on the ten datasets. Among the eight algorithms, three spectrum transformation methods obtain the lowest classification accuracies on seven of the ten datasets. SMO-IKSVM and TDCASVM achieve similar results to three spectrum transformation methods. IKSVM-CA slightly excels the spectrum transformation methods on eight datasets. But it has too much parameters to tune and would fail when the number of positive eigenvalues is very small (i.e. the $Breast-cancer$ dataset). ESVM exceeds IKSVM-CA on six of the ten datasets yet is worse than 1-norm IKSVM on most of these datasets. Our algorithm IKSVM-DC is superior to 1-norm IKSVM on all the datasets.

The experiments about the convergence of IKSVM-DC are conducted on five datasets $Ionosphere$, $Sonar$, $Flare-solar$, $Balls3D$ and $CoilYork$. 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 Figure 1. We can see that the value $\|d(\boldsymbol{\beta})\|^2$ gradually converges in a few iterations on the five datasets.

Figure 2 demonstrates the different performance between IKSVM-DC with and without a line search step on the dataset $Flare-solar$. 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 nearly three 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.

Furthermore, the computational cost of the five methods Shift, SMO-IKSVM, TDCASVM, 1-norm IKSVM and IKSVM-DC is $O(n^2)$, while 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 (Wu, Chang, and Zhang 2005) to estimate such a ρ that satisfies $\rho \geq -\min(\{\lambda_i\}_{i=1}^n)$ in actual implementation. Thus, IKSVM-DC is comparable to other algorithms on computational cost.

6 Conclusion

Instead of employing the dual form of IKSVM, we directly focus on the primal form in this paper. Considering the

characteristics of the spectrum for the indefinite kernels, we transform the non-convex primal IKSVM problem into a formulation of DC equivalently, and propose an algorithm IKSVM-DC to obtain a local minimum for it. Furthermore, in order to accelerate the convergence rate of IKSVM-DC, we conduct a line search along the descent direction at each iteration. Extensive comparative experiments validate the effectiveness of our algorithm IKSVM-DC.

7 Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 61375057, 61300165 and 61403193) and Natural Science Foundation of Jiangsu Province of China (Grant No. BK20131298). Furthermore, the work was also supported by Collaborative Innovation Center of Wireless Communications Technology.

References

Akoa, F. B. 2008. Combining dc algorithms (dcas) and decomposition techniques for the training of nonpositive–semidefinite kernels. *IEEE Transactions on Neural Networks* 19(11):1854–1872.

Alabdulmohsin, I. M.; Gao, X.; and Zhang, X. 2014. Support vector machines with indefinite kernels. In *Proceedings of the Sixth Asian Conference on Machine Learning*, volume 39, 32–47. Nha Trang, Vietnam: JMLR.org.

Artacho, F. J. A.; Fleming, R. M.; and Vuong, P. T. 2015. Accelerating the dc algorithm for smooth functions. *arXiv preprint arXiv:1507.07375*.

Blake, C., and Merz, C. J. 1998. Uci repository of machine learning databases. *Online at http://archive.ics.uci.edu/ml/*.

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):27.

Chen, J., and Ye, J. 2008. Training svm with indefinite kernels. In *Proceedings of the Twenty-fifth International Conference on Machine Learning*, 136–143. Helsinki, Finland: ACM.

Chen, Y.; Gupta, M. R.; and Recht, B. 2009. Learning kernels from indefinite similarities. In *Proceedings of the Twenty-sixth International Conference on Machine Learning*, 145–152. Montreal, Quebec, Canada: ACM.

Cristianini, N., and Shawe-Taylor, J. 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press.

Dinh, T. P., and Le Thi, H. A. 2014. Recent advances in dc programming and dca. *Transactions on Computational Intelligence XIII* 1–37.

Duin, R. 2000. Prtools version 3.0: A matlab toolbox for pattern recognition. In *Proceedings of the International Society for Optical Engineering*. Citeseer.

Graepel, T.; Herbrich, R.; Bollmann-Sdorra, P.; and Obermayer, K. 1999. Classification on pairwise proximity data. In *The Thirteenth Conference on Neural Information Processing Systems*, 438–444. Denver, Colorado, USA: MIT Press.

Gu, S., and Guo, Y. 2012. Learning svm classifiers with indefinite kernels. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 942–948. Toronto, Ontario, Canada: AAAI Press.

Lin, H.-T., and Lin, C.-J. 2003. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. *Neural Computation* 1–32.

Loosli, G.; Canu, S.; and Ong, C. S. 2016. Learning svm in kreĭn spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(6):1204–1216.

Luss, R., and d'Aspremont, A. 2008. Support vector machine classification with indefinite kernels. In *The Twenty-First Conference on Neural Information Processing Systems*, 953–960. Vancouver, British Columbia, Canada: Curran Associates, Inc.

Mosek, A. 2010. The mosek optimization software. *Online at http://www.mosek.com*.

Ong, C. S.; Mary, X.; Canu, S.; and Smola, A. J. 2004. Learning with non-positive kernels. In *Proceedings of the Twenty-first International Conference on Machine Learning*, 81. Banff, Alberta, Canada: ACM.

Pekalska, E.; Paclik, P.; and Duin, R. P. 2001. A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research* 2(Dec):175–211.

Piot, B.; Geist, M.; and Pietquin, O. 2014. Difference of convex functions programming for reinforcement learning. In *The Twenty-seventh Conference on Neural Information Processing Systems*, 2519–2527. Montreal, Quebec, Canada: Curran Associates, Inc.

Rätsch, G.; Onoda, T.; and Müller, K.-R. 2001. Soft margins for adaboost. *Machine learning* 42(3):287–320.

Roth, V.; Laub, J.; Kawanabe, M.; and Buhmann, J. M. 2003. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(12):1540–1551.

Saigo, H.; Vert, J.-P.; Ueda, N.; and Akutsu, T. 2004. Protein homology detection using string alignment kernels. *Bioinformatics* 20(11):1682–1689.

Tao, P. D., and An, L. T. H. 1997. Convex analysis approach to dc programming: theory, algorithms and applications. *Acta Mathematica Vietnamica* 22(1):289–355.

Vapnik, V. 2013. *The nature of statistical learning theory*. Springer Science & Business Media.

Wang, C.; Song, Y.; Li, H.; Zhang, M.; and Han, J. 2016. Text classification with heterogeneous information network kernels. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2130–2136. Phoenix, Arizona, USA: AAAI Press.

Wu, G.; Chang, E. Y.; and Zhang, Z. 2005. An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines. In *Proceedings of the Twenty-second International Conference on Machine Learning*, volume 8. Citeseer.

Zhang, M.-L., and Zhou, Z.-H. 2013. Exploiting unlabeled data to enhance ensemble diversity. *Data Mining and Knowledge Discovery* 26(1):98–129.