Neural Processing Letters (2019) 50:165–188
https://doi.org/10.1007/s11063-019-10019-7

A Primal Framework for Indefinite Kernel Learning

Hui Xue[1,2] · **Lin Wang**[1,2] · **Songcan Chen**[3] · **Yunyun Wang**[4]

Published online: 10 June 2019

Abstract
Kernel methods have been widely applied in machine learning to solve complex nonlinear problems. Kernel selection is one of the key issues in kernel methods, since it is vital for improving generalization performance. Traditionally, the selection of kernel is restricted to be positive definite which makes their applicability partially limited. Actually, in many real applications such as gene identification and object recognition, indefinite kernels frequently emerge and can achieve better performance. However, compared to positive definite ones, indefinite kernels are more complicated due to the non-convexity of the subsequent optimization problems, which leads to the incapability of most existing kernel algorithms. Some indefinite kernel methods have been proposed based on the dual of support vector machine (SVM), which mostly emphasize on how to transform the non-convex optimization to be convex by using positive definite kernels to approximate indefinite ones. In fact, the duality gap in SVM usually exists in the case of indefinite kernels and therefore these algorithms do not indeed solve the indefinite kernel problems themselves. In this paper, we present a novel framework for indefinite kernel learning derived directly from the primal of SVM, which establishes several new models not only for single indefinite kernel but also extends to multiple indefinite kernel scenarios. Several algorithms are developed to handle the non-convex optimization problems in these models. We further provide a constructive approach for kernel selection in the algorithms by using the theory of similarity functions. Experiments on real world datasets demonstrate the superiority of our models.

Keywords Indefinite kernel · Primal problem · Multiple kernel learning · Machine learning

1 Introduction

Kernel methods have been extensively used in a variety of learning tasks with the successful applications of the best known paradigm support vector machine (SVM) [50]. They work through mapping the input data into a high-dimensional (possibly infinite-dimensional) fea-

This work was supported by the National Natural Science Foundations of China (Grant Nos. 61375057, 61300165 and 61403193), the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20131298) and the National Key Research and Development Program of China (Grant Nos. 2016YFC1306700 and 2016YFC1306704). It was also supported by Collaborative Innovation Center of Wireless Communications Technology.

Extended author information available on the last page of the article

ture space [4], in order to transform nonlinear learning problems in the original feature space into tractable linear ones which can be easily solved by linear learning machines. Kernels are introduced by replacing the inner products on all pairs of the mapping data and actually act as the mapping [4]. By using the kernels, these methods can avoid explicitly representing the mapping and thus the possible curse of dimensionality caused by the numerical calculation in the high-dimensional mapping space.

Kernel selection is one of the key issues in kernel methods, which directly affects whether an appropriate mapping space can be found and thus influences the generalization performance. Following the classical statistical learning theory, so-selected kernels require to be (conditionally) positive definite (PD) and satisfy the Mercer's condition, in order to ensure the existence of a reproducing kernel Hilbert space (RKHS) and lead to convex formulations for the optimization problems [4]. As a result, the corresponding algorithms can converge to global optima.

In practice, however, such requirement turns out to be too strict [17]. Actually, standard PD kernels are inapplicable in many situations [41] such as suboptimal optimization procedures for measure derivation [42], partial projections or occlusions [26], and context-dependent alignments or object comparisons [46]. On the contrary, indefinite kernels have increasingly emerged and shown much better performance [3,25,31,33,49]. Liu utilized an indefinite fractional power polynomial kernel in kernel principal component analysis (KPCA) in face recognition, which achieves higher recognition accuracies than the KPCA using PD polynomial kernels [32]. Liwicki et al. applied an indefinite robust gradient-based kernel in an incremental KPCA algorithm for visual tracking, leading to more efficient and exact results [34].

Furthermore, there are some other situations in which, although PD kernels can be applied, indefinite kernels more often appear due to in which additional problem-specific prior knowledge is integrated and boost the learning-performance of problems at hand [54]. Xue et al. embedded discriminative and structural information of data to the traditional regularization framework and thus deduced an indefinite discriminative regularization term for classification [24,51,55]. Ackermann et al. introduced several problem-dependent non-metric distance measures into k-median clustering and proposed indefinite kernel clustering algorithms [1]. Haasdonk and Pkalska presented indefinite kernel discriminant analysis methods with aiming to incorporate the invariance into feature extraction problems [18,19].

In past few years, indefinite kernels have attracted more and more attention in machine learning community. However, compared to PD ones, indefinite kernels are more complex. Thanks to the loss of the PD-ness of the kernels, the corresponding optimization problems built on them are more likely non-convex which result in most of existing PD kernel methods inapplicable. Recently, some indefinite kernel algorithms have been developed for solving such problem. One simplest way is to convert into a corresponding positive semi-definite kernel matrix by transforming the spectrum of the indefinite kernel matrix [7], including "*Clip*" which sets the negative eigenvalues to zeros [43], "*Flip*" which flips the sign of the negative eigenvalues [15], and "*Shift*" which shifts the eigenvalues by a positive constant [47]. However, such methods actually change indefinite kernels themselves forcibly. A few other works use the indefinite kernel matrix directly and formulate as variant optimization problems from standard PD kernel methods. Haasdonk executed indefinite kernel SVM by minimizing the distances between convex hulls in pseudo-Euclidean space [17]. Ong et al. extended the common inner product in RKHS to a reproducing kernel Kreĭn space (RKKS), where the product associated with the indefinite kernel can be negative and a more general representer theorem is derived and met accordingly [38].

Another more sophisticated mode of handling indefinite kernels is to convert the associated non-convex optimization into a convex one. Specially, the indefinite kernel matrix is considered as the noisy observation of some unknown positive semi-definite one and then approximated by learning a proxy PD kernel. Luss and d'Aspremont proposed a dual model of SVM with an additional regularization term which measures the similarity between the proxy and the original indefinite kernel matrices and then quadratically smoothed the resulting non-differentiable objective for optimization, consequently yielding the two optimization algorithms, i.e., the project gradient method and the analytic center cutting plan method to simultaneously learn the support vectors as well as the proxy kernel [36]. Chen and Ye further reformulated such an objective as a semi-infinite quadratically constrained linear program which can ensure convergence to a global optimum [7]. Ying et al. verified that the objective involved is continuously differentiable and its gradient is Lipschitz continuous, and then used Nesterov's smooth optimization method with achievable optimal convergence rate [57]. Gu and Guo reformulated the common KPCA as a general kernel transformation framework and then incorporated it into the SVM classification to formulate a joint optimization model for solving indefinite kernel SVM problems, which can make consistent kernel transformations over training and testing samples [16]. Loosli et al. [35] extended indefinite kernel into a Reproducing Kernel Kreĭn Space (RKKS) and tried to obtain a stabilized solution for the indefinite kernel problem. However, these methods are basically designed based on the dual of SVM rather than the original non-convex primal problem. In fact, the duality gap between the two problems usually exists in the case of indefinite kernels, which leads to the difference between their solutions. As a result, such methods actually do not indeed solve the problem of indefinite kernel SVM itself.

In this paper, we propose a novel SVM framework for indefinite kernel learning. The main contributions of this paper include:

- We focus on the indefinite kernel SVM problem itself and thus derive the framework directly from the primal. Considering the particularity of indefinite kernels, the framework is constructed on the larger RKKS (than RKHS) and the corresponding solutions satisfy the generalized representer theorem in the RKKS.
- In the framework, we not only establish a primal model for single indefinite kernel, but also extend it to multiple indefinite kernel scenarios. Based on different properties of base kernels and combination coefficients, we further provide a generalized multiple kernel learning framework that not only covers common multiple PD kernel methods, but also deduces two novel non-convex multiple indefinite kernel models which emphasize on the convex combination of indefinite base kernels and the non-convex combination of PD base kernels respectively.
- According to the single indefinite kernel model, we present an algorithm using Polak-Ribiere-Polyak (PRP) conjugate gradient. In the multiple indefinite kernel models, a two-stage algorithm is further developed to optimize both the SVM parameters and combination coefficients alternately.
- A constructive approach for kernel selection in the algorithms is further provided by using more general theory of similarity functions. Systematic experiments demonstrate that our models can show much better performance than related methods in real world applications.

The rest of the paper is organized as follows. Section 2 briefly analyzes the duality gap in the case of indefinite kernels. The primal model for single indefinite kernel and the corresponding algorithm are discussed in Sect. 3. Section 4 extends the framework to multiple indefinite

kernels. Further discussion for the models is presented in Sect. 5. In Sect. 6, systematically experimental comparisons are conducted. Some conclusions are drawn in Sect. 7.

2 Dual Gap

Given a training set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n \in \boldsymbol{R}^m \times \{\pm 1\}$, the primal formulation of soft margin SVM classification is given by

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

and the associated dual problem is

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left\langle \boldsymbol{x}_i, \boldsymbol{x}_j \right\rangle \\ &s.t. \quad \boldsymbol{\alpha}^T \boldsymbol{y} = 0 \quad 0 \leq \alpha_i \leq 1/\lambda, \quad i = 1, \ldots, n \end{aligned} \tag{2}$$

where $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_n]^T$ and $\boldsymbol{y} = [y_1, \cdots, y_n]^T$ are the vectors of Lagrange dual variables and class labels respectively. λ is a pre-specified parameter.

When the data are linear inseparable, we usually embed them into a high-dimensional feature space by using a kernel function instead of the inner products on all pairs of the embeddings. As a result, the dual of kernel SVM can be further formulated as

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &s.t. \quad \boldsymbol{\alpha}^T \boldsymbol{y} = 0 \quad 0 \leq \alpha_i \leq 1/\lambda, \quad i = 1, \ldots, n \end{aligned} \tag{3}$$

where $K(\cdot, \cdot)$ a kernel function.

Let the feasible region of the primal problem (1) be

$$S = \{y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \ldots, n\}$$

Then the Lagrangian

$$L(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{w}, b, \boldsymbol{\xi}) = f(\boldsymbol{w}, b, \boldsymbol{\xi}) - \sum_{i=1}^{n} \alpha_i [y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) - 1 + \xi_i] - \sum_{i=1}^{n} \gamma_i \xi_i \tag{4}$$

can be viewed as a function with $(\boldsymbol{w}, b, \boldsymbol{\xi}) \in S$ and $(\boldsymbol{\alpha}, \boldsymbol{\gamma}) \in \Lambda = \boldsymbol{R}_+^n \times \boldsymbol{R}_+^n$.

Define the objective function associated with the primal problem (1) as

$$L_P(\boldsymbol{w}, b, \boldsymbol{\xi}) = \sup_{(\boldsymbol{\alpha}, \boldsymbol{\gamma}) \in \Lambda} L(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{w}, b, \boldsymbol{\xi}) \tag{5}$$

and the dual function as

$$L_D(\boldsymbol{\alpha}, \boldsymbol{\gamma}) = \inf_{(\boldsymbol{w}, b, \boldsymbol{\xi}) \in S} L(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{w}, b, \boldsymbol{\xi}) \tag{6}$$

As a result, the two optimization problems can boil down to

$$\min_{(\boldsymbol{w}, b, \boldsymbol{\xi}) \in S} L_P(\boldsymbol{w}, b, \boldsymbol{\xi})$$

and

$$\max_{(\boldsymbol{\alpha},\boldsymbol{\gamma})\in\Lambda} L_D(\boldsymbol{\alpha},\boldsymbol{\gamma})$$

Obviously, for every $(\boldsymbol{\alpha},\boldsymbol{\gamma}) \in \Lambda$ and $(\boldsymbol{w}, b, \boldsymbol{\xi}) \in S$, we have

$$L_D(\boldsymbol{\alpha},\boldsymbol{\gamma}) \leq L_P(\boldsymbol{w}, b, \boldsymbol{\xi})$$

So the goal of the dual problem is to find the best (maximum) lower bound of the primal [48]. And the corresponding difference between the two problems is called duality gap

$$\delta = \min_{(\boldsymbol{w},b,\boldsymbol{\xi})\in S} L_P(\boldsymbol{w}, b, \boldsymbol{\xi}) - \max_{(\boldsymbol{\alpha},\boldsymbol{\gamma})\in\Lambda} L_D(\boldsymbol{\alpha},\boldsymbol{\gamma}) \geq 0 \tag{7}$$

In terms of PD kernels, the primal and dual problems are typical convex quadratic programming optimizations. In such case, the optimal solutions of the two problems $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$ and $(\boldsymbol{\alpha}^*, \boldsymbol{\gamma}^*)$ will constitute a point $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}^*, \boldsymbol{\gamma}^*)$ of the Lagrangian satisfied

$$\begin{aligned}\max_{(\boldsymbol{\alpha},\boldsymbol{\gamma})\in\Lambda} L(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\gamma}) &= L(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}^*, \boldsymbol{\gamma}^*) \\ &= \min_{(\boldsymbol{w},b,\boldsymbol{\xi})\in S} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}^*, \boldsymbol{\gamma}^*)\end{aligned} \tag{8}$$

Consequently, the duality relationship holds true

$$\min_{(\boldsymbol{w},b,\boldsymbol{\xi})\in S} L_P(\boldsymbol{w}, b, \boldsymbol{\xi}) = \max_{(\boldsymbol{\alpha},\boldsymbol{\gamma})\in\Lambda} L_D(\boldsymbol{\alpha},\boldsymbol{\gamma}) \tag{9}$$

and the corresponding duality gap is zero [48]. In this situation, we can seek the solution to the primal by firstly solving the dual to get $(\boldsymbol{\alpha}^*, \boldsymbol{\gamma}^*)$ and then determining the solution $(\boldsymbol{w}^*, b^*, \boldsymbol{\xi}^*)$.

However, in terms of indefinite kernels, the primal and dual problems become non-convex, which leads to such point non-existent. As a result, the duality relationship (9) is not valid any longer [48]. The dual problem can only be used to bound the solution of the primal from $L_D(\boldsymbol{\alpha},\boldsymbol{\gamma}) \leq L_P(\boldsymbol{w}, b, \boldsymbol{\xi})$.

In other words, the optimizations of the primal and dual problems are actually not equivalent in indefinite kernel SVM. A more reasonable way is directly learning from the primal.

3 Primal Framework for Single Indefinite Kernel

In this section, we firstly derive a primal model for single indefinite kernel and then present the corresponding optimization algorithm. The whole model is founded in the RKKS [38].

3.1 Model Construction

We rewrite the primal SVM problem (1) as an unconstrained optimization problem [6,37]:

$$\min_{\boldsymbol{w},b} \lambda \langle \boldsymbol{w}, \boldsymbol{w} \rangle + \sum_{i=1}^{n} V(y_i, \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle + b) \tag{10}$$

where $V(\cdot)$ is a loss function.

Now let us consider the nonlinear SVM with an indefinite kernel which induces a RKKS $\tilde{\boldsymbol{K}}$. The optimization problem (10) becomes

$$\min_{f\in\tilde{\boldsymbol{K}},b} \lambda\langle \boldsymbol{f}, \boldsymbol{f}\rangle_{\tilde{\boldsymbol{K}}} + \sum_{i=1}^{n} V\left(y_i, f(\boldsymbol{x}_i) + b\right) \tag{11}$$

In the RKKS, the solution to the problem of minimizing a regularized risk functional still admits a similar representation in terms of an expansion over the training samples to the RKHS.

Theorem 1 (Representer Theorem) [38] *Let $\tilde{\boldsymbol{K}}$ be an RKKS with kernel K. Denote by $V(\boldsymbol{f}, \mathcal{X})$ a continuous convex loss functional depending on $\boldsymbol{f} \in \tilde{\boldsymbol{K}}$ only via its evaluations $f(\boldsymbol{x}_i)$ with $\boldsymbol{x}_i \in \mathcal{X}$, let $\Omega(\langle \boldsymbol{f}, \boldsymbol{f}\rangle)$ be a continuous stabilizer with strictly monotonic $\Omega : \boldsymbol{R} \to \boldsymbol{R}$ and let $C\{\boldsymbol{f}, \mathcal{X}\}$ be a continuous functional imposing a set of constraints on $\boldsymbol{f}$, that is $C : \tilde{\boldsymbol{K}} \times \mathcal{X} \to \boldsymbol{R}$. Then if the optimization problem*[1]

$$\begin{aligned} &stabilize_{f\in\tilde{\boldsymbol{K}}} \; V(\boldsymbol{f}, \mathcal{X}) + \Omega\left(\langle \boldsymbol{f}, \boldsymbol{f}\rangle_{\tilde{\boldsymbol{K}}}\right) \\ &\quad s.t. \;\; C\{\boldsymbol{f}, \mathcal{X}\} \le \zeta \end{aligned}$$

has a saddle point $\boldsymbol{f}^*$, it admits the expansion $\boldsymbol{f}^* = \sum_{i=1}^{n} \beta_i K(x_i, \cdot)$ where $\boldsymbol{x}_i \in \mathcal{X}$ and $\beta_i \in \boldsymbol{R}$.

Consequently, the primal model of single indefinite kernel can be further expressed as

$$\min_{\boldsymbol{\beta},b} \lambda\boldsymbol{\beta}^T \boldsymbol{K}\boldsymbol{\beta} + \sum_{i=1}^{n} V\left(y_i, \boldsymbol{K}^i\boldsymbol{\beta} + b\right) \tag{12}$$

where $\boldsymbol{K}$ is the indefinite kernel matrix with $\boldsymbol{K}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\boldsymbol{K}^i$ is the ith row of $\boldsymbol{K}$. It is worth noting that the coefficients β_i are not α_i in the optimization (2), and thus should not be interpreted as Lagrange multipliers. In fact, the main difference between them is the value range: α_i are required to be non-negative but such requirement is inapplicable to β_i. Furthermore, for the solution $\boldsymbol{\beta}^*$ of (12), the corresponding support vector set is

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

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

3.2 Optimization Algorithm

We select the smooth quadratic hinge loss function as $V(\cdot)$, which can make the primal continuous and differentiable in $\mathbf{f}$ and so in $\boldsymbol{\beta}$ [6,37]. Note that if $\boldsymbol{K}$ is not symmetric, let $\boldsymbol{K} = (\boldsymbol{K}+\boldsymbol{K}^T)/2$. So the optimization problem after adding the scaling constant 1/2 becomes

$$\min_{\boldsymbol{\beta},b} \frac{1}{2}\left[\lambda\boldsymbol{\beta}^T \boldsymbol{K}\boldsymbol{\beta} + \sum_{i=1}^{n} \max\left(0, 1 - y_i\left(\boldsymbol{K}^i\boldsymbol{\beta} + b\right)\right)^2\right] \tag{13}$$

Although (13) is much similar to the traditional primal PD kernel SVM problem, it is actually an unconstrained non-convex optimization in terms of indefinite kernels which is an NP-hard problem. Fortunately, some state-of-the-art techniques can still be applied, but

[1] Here "stabilize" means finding a stationary point in a RKKS.

Algorithm 1 PRIMAL SINGLE INDEFINITE KERNEL SVM (PRIMAL IKSVM)

Initial : $\mathbf{z}^{(0)} = \mathbf{0} \in \mathbf{R}^{(n+1)}$, $\mathbf{d}^{(0)} = -\nabla^{(0)} = -[\mathbf{y}; sum(\mathbf{y})]$, $t = 0$
while stopping criterion not met **do**
  $t = t + 1$;
  Find the optimal step s^* by exact Newton line search:
    $s^* = \min_s F(\mathbf{z}^{(t-1)} + s \times \mathbf{d}^{(t-1)})$;
  Update $\mathbf{z}^{(t)} = \mathbf{z}^{(t-1)} + s^* \times \mathbf{d}^{(t-1)}$;
  Compute the descent direction $\mathbf{d}^{(t)}$:
    $SV = \{\mathbf{x}_i \in \mathcal{X} \mathbf{s.t.} y_i f(\mathbf{x}_i) < 1\}$;
    $\nabla^{(t)} = \begin{bmatrix} \lambda \mathbf{fi} + \mathbf{I}_{SV}(\mathbf{Kfi} + \mathbf{1}b) - \mathbf{I}_{SV}\mathbf{y} \\ \mathbf{1}^T \mathbf{I}_{SV}(\mathbf{Kfi} + \mathbf{1}b) - \mathbf{1}^T \mathbf{I}_{SV}\mathbf{y} \end{bmatrix}$;
    $\hat{\nabla}^{(t)} = \mathbf{P} \times \nabla^{(t)}$;
    $\rho = \max(0, \frac{\hat{\nabla}^{(t)T}(\nabla^{(t)} - \nabla^{(t-1)})}{\hat{\nabla}^{(t-1)T}\nabla^{(t-1)}})$;
    $\mathbf{d}^{(t)} = -\nabla^{(t)} + \rho \times \mathbf{d}^{(t-1)}$;
end while

they will reach local minima [52]. Here we adopt the conjugate gradient with PRP iteration formula that has shown empirically better performance in solving primal SVM problems [37].

More specifically, we firstly compute the gradients of the variables $\boldsymbol{\beta}$ and b by (13)

$$\begin{aligned} \nabla = \begin{bmatrix} \nabla_{\boldsymbol{\beta}} \\ \nabla_b \end{bmatrix} &= \begin{bmatrix} \lambda \boldsymbol{K}\boldsymbol{\beta} + \sum_{i \in SV} (1 - y_i(\boldsymbol{K}^i\boldsymbol{\beta} + b))(-y_i \boldsymbol{K}^i) \\ \sum_{i \in SV} (1 - y_i(\boldsymbol{K}^i\boldsymbol{\beta} + b))(-y_i) \end{bmatrix} \\ &= \begin{bmatrix} \lambda \boldsymbol{K}\boldsymbol{\beta} + \boldsymbol{K}\boldsymbol{I}_{SV}(\boldsymbol{K}\boldsymbol{\beta} + \mathbf{1}b) - \boldsymbol{K}\boldsymbol{I}_{SV}\boldsymbol{y} \\ \mathbf{1}^T \boldsymbol{I}_{SV}(\boldsymbol{K}\boldsymbol{\beta} + \mathbf{1}b) - \mathbf{1}^T \boldsymbol{I}_{SV}\boldsymbol{y} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{K} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \times \begin{bmatrix} \lambda \boldsymbol{\beta} + \boldsymbol{I}_{SV}(\boldsymbol{K}\boldsymbol{\beta} + \mathbf{1}b) - \boldsymbol{I}_{SV}\boldsymbol{y} \\ \mathbf{1}^T \boldsymbol{I}_{SV}(\boldsymbol{K}\boldsymbol{\beta} + \mathbf{1}b) - \mathbf{1}^T \boldsymbol{I}_{SV}\boldsymbol{y} \end{bmatrix} \end{aligned} \tag{14}$$

where $\mathbf{1} \in \boldsymbol{R}^n$ and $\mathbf{0} \in \boldsymbol{R}^n$ are the vectors whose elements are all equal to one and zero respectively. $\boldsymbol{I}_{SV} \in \boldsymbol{R}^{n \times n}$ denotes a diagonal matrix whose elements along the primal diagonal non-zero (equal to one) only correspond to the position of support vectors at the current iteration. In particular, let the matrix $\boldsymbol{P} = \begin{bmatrix} \boldsymbol{K} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$ as a preconditioner which can be computed in advance to avoid calculating repeatedly in the iterations [21,37].

Then we apply conjugate gradient to compute descent direction

$$\boldsymbol{d}^{(t)} = -\nabla^{(t)} + \rho \times \boldsymbol{d}^{(t-1)}$$

where ρ is updated by PRP formula

$$\rho = \frac{\hat{\nabla}^{(t)T}(\nabla^{(t)} - \nabla^{(t-1)})}{\hat{\nabla}^{(t-1)T}\nabla^{(t-1)}}$$

Once the direction is found, we use exact Newton line search to find the optimal step and further update the variables $\boldsymbol{\beta}$ and b.

Let $\boldsymbol{z} = [\boldsymbol{\beta}^T b]^T$ and $F(\cdot)$ denotes the objective function. The stopping criterion used in the algorithm is the difference between two adjacent iteration values of the quadratic hinge loss function less than 10^{-6}. The whole algorithm can be summarized in Algorithm 1.

Complexity Analysis We investigate each step of the algorithm. Detailedly, the exact Newton line search for the optimal step leads to a complexity of $O(n^2)$. The search process for

support vectors also has the complexity $O(n^2)$. The computation of the descent direction by conjugate gradient is with a complexity of $O(nn_{SV})$, where n_{SV} denotes the number of support vectors. Consequently, the algorithm at most has the complexity of $O(n^2)$.

4 Primal Framework for Multiple Indefinite Kernel

Many previous studies have shown it is not reasonable that using a single kernel to map all samples, especially when the samples contain heterogeneous information [40] or distribute non-flatly in high-dimensional feature space [58]. Multiple kernel learning (MKL) [20,22,23, 53] utilizes a combination of multiple base kernels instead of a single kernel to mix multiple source information, and thus has been regarded as a promising technique to improve the performance of kernel methods effectively [8,9,14,27,28,56]. Particularly, some recent state-of-the-art works have presented many different MKL algorithms from different perspectives. Aiolli and Donini utilized the kernel optimization of the margin distribution technique and proposed a scalable MKL algorithm named easyMKL which combines large sets of base kernels by solving a simple quadratic problem [2]. Furthermore, they combined a hierarchy of base kernels with easyMKL to generate overall deeper kernels [10]. Fan et al. incorporated the locality preserving projection into multiple empirical kernel learning framework and proposed a lower generalization error bound algorithm [12]. Then they further proposed a multiple random empirical kernel learning machine to deal with large-scale problems which has good efficient performance in both computation and memory [13].

Consequently, we further extend the proposed primal framework to multiple indefinite kernel scenarios. To the best of our knowledge, this is more likely the first attempt to indeed fuse indefinite kernels with MKL. Ong et al. constructed a hyperkernel in a RKHS on the space of kernels itself which can be expressed as a non-convex linear combination of PD kernels [39]. However, in order to make the kernel matrix positive semi-definite out of easily being solved, they actually imposed extra non-negative constraints on the combination coefficients. Hinrichs et al. presented a Q-MKL method in which though base kernels can be allowed to be indefinite, their combined kernel is still confined to be PD by imposing necessary constraints [22]. Kowalski et al. proposed a multiple kernel algorithm involving indefinite kernels [29]. However, the algorithm more emphasizes on using the mixed norm regularization to reach better sparsity rather than multiple indefinite kernel learning itself.

4.1 Model Construction

In this subsection, we firstly present a generalized MKL framework and then derive two primal multiple indefinite kernel models. Based on the single kernel model (11), the multiple kernel model can be uniformly formulated as

$$\min_{f\in\tilde{\boldsymbol{K}}_{\boldsymbol{\mu}},\boldsymbol{\mu}\in\Delta,b} \lambda\langle \boldsymbol{f},\boldsymbol{f}\rangle_{\tilde{\boldsymbol{K}}_{\boldsymbol{\mu}}} + \sum_{i=1}^{n} V(y_i, f(\boldsymbol{x}_i)+b) \tag{15}$$

where $\tilde{\boldsymbol{K}}_{\boldsymbol{\mu}}$ is a RKKS parameterized by $\boldsymbol{\mu}$, which is endowed with kernel function $K(\cdot,\cdot,\boldsymbol{\mu}) = \sum_{j=1}^{M}\mu_j K_j(\cdot,\cdot)$. $\{K_j(\cdot,\cdot)\}_{j=1}^{M}$ is a group of base kernels allowed to be indefinite. In addition, we also extend the coefficients $\boldsymbol{\mu}$ from the convex combination commonly required in traditional MKL methods

Table 1 Generalized MKL framework

	$K_j(\cdot,\cdot)$PD	$K_j(\cdot,\cdot)$Indefinite
Δ_1	I	II
Δ_2	III	IV

$$\Delta_1 = \{\boldsymbol{\mu} \in \boldsymbol{R}_+^M : \sum_{j=1}^{M} \mu_j = 1, \mu_j \geq 0\}$$

to the non-convex combination

$$\Delta_2 = \{\boldsymbol{\mu} \in \boldsymbol{R}^M : |\mu_j| \leq c, c \in \boldsymbol{R}_+\}$$

In order to avoid degradation, an absolute value constraint is imposed to μ_j. c is a positive constant.

Consequently, based on different properties of base kernels and combination coefficients, we can naturally generate a generalized MKL framework which classifies multiple kernel models into four categories, as shown in Table 1.

Most existing multiple PD kernel methods fall in Category I. Category IV actually is involved in Category II, since the minus sign of μ_j can be absorbed into the base kernel functions, doing so can not change the indefiniteness of the kernels, but can convert μ_j to be positive.

Categories II and III deduce two new primal multiple indefinite kernel models, termed as Primal MIKSVM-1 and Primal MIKSVM-2, which expand base kernels and combination coefficients respectively.

Concretely, Primal MIKSVM-1 focuses on the convex combination of indefinite base kernels

$$\begin{aligned} \min_{f \in \tilde{\boldsymbol{K}}_{\boldsymbol{\mu}}, \boldsymbol{\mu}, b} \; & \lambda \sum_{j=1}^{M} \mu_j \left\langle \boldsymbol{f}_j, \boldsymbol{f}_j \right\rangle_{\tilde{\boldsymbol{K}}_j} + \sum_{i=1}^{n} V\left(y_i, \sum_{j=1}^{M} \mu_j f_j(\boldsymbol{x}_i) + b\right) \\ s.t. \; & \sum_{j=1}^{M} \mu_j = 1, \quad \mu_j \geq 0 \end{aligned} \tag{16}$$

By the Representer Theorem, we have $f_j^*(\boldsymbol{x}) = \sum_{i=1}^{n} \beta_i K_j(\boldsymbol{x}_i, \boldsymbol{x})$. So the model (16) can be further formulated as

$$\begin{aligned} \min_{\boldsymbol{\beta}, \boldsymbol{\mu}, b} \; & \lambda \boldsymbol{\beta}^T \boldsymbol{K} \boldsymbol{\beta} + \sum_{i=1}^{n} V(y_i, \boldsymbol{K}^i \boldsymbol{\beta} + b) \\ s.t. \; & \boldsymbol{K} = \sum_{j=1}^{M} \mu_j \boldsymbol{K}_j, \; \boldsymbol{K}^i = \sum_{j=1}^{M} \mu_j \boldsymbol{K}_j^i \\ & \sum_{j=1}^{M} \mu_j = 1, \; \mu_j \geq 0 \end{aligned} \tag{17}$$

where $\boldsymbol{K}_j$ is the indefinite base kernel matrix and $\boldsymbol{K}_j^i$ is the ith row of $\boldsymbol{K}_j$.

Primal MIKSVM-2 emphasizes on the non-convex combination of PD base kernels:

$$\begin{aligned} \min_{f\in\tilde{\boldsymbol{K}}_{\boldsymbol{\mu}},\boldsymbol{\mu},b} \ & \lambda\sum_{j=1}^{M}\mu_j\left\langle \boldsymbol{f}_j,\boldsymbol{f}_j\right\rangle_{\tilde{\boldsymbol{K}}_j}+\sum_{i=1}^{n}V(y_i,\sum_{j=1}^{M}\mu_j f_j(\boldsymbol{x}_i)+b) \\ s.t.\ & |\mu_j|\le c,\quad j=1,\ldots,M \end{aligned} \tag{18}$$

Following the same deduction as (17), the model of Primal MIKSVM-2 can finally boil down to

$$\begin{aligned} \min_{\boldsymbol{\beta},\boldsymbol{\mu},b}\ & \lambda\boldsymbol{\beta}^T\boldsymbol{K}\boldsymbol{\beta}+\sum_{i=1}^{n}V(y_i,\boldsymbol{K}^i\boldsymbol{\beta}+b) \\ s.t.\ & \boldsymbol{K}=\sum_{j=1}^{M}\mu_j\boldsymbol{K}_j,\ \boldsymbol{K}^i=\sum_{j=1}^{M}\mu_j\boldsymbol{K}_j^i \\ & |\mu_j|\le c,\quad j=1,\ldots,M \end{aligned} \tag{19}$$

It is worth to point out that single indefinite kernel model Primal IKSVM and most multiple PD kernel methods are in fact special cases of Primal MIKSVM-1 and Primal MIKSVM-2 respectively, while the constraints degenerate to accord with their specific learning scenarios. In this sense, Primal MIKSVM-1 and Primal MIKSVM-2 are more general and thus more likely possess better adaptability to complex applications.

4.2 Optimization Algorithm

We also choose the quadratic hinge loss function as $V(\cdot)$. Adding the scaling constant $1/2$, the two models become

$$\begin{aligned} \min_{\boldsymbol{\beta},\boldsymbol{\mu},b}\ & \frac{1}{2}\left[\lambda\boldsymbol{\beta}^T\boldsymbol{K}\boldsymbol{\beta}+\sum_{i=1}^{n}\max(0,1-y_i(\boldsymbol{K}^i\boldsymbol{\beta}+b))^2\right] \\ s.t.\ & \boldsymbol{K}=\sum_{j=1}^{M}\mu_j\boldsymbol{K}_j,\ \boldsymbol{K}^i=\sum_{j=1}^{M}\mu_j\boldsymbol{K}_j^i \\ & \sum_{j=1}^{M}\mu_j=1,\quad \mu_j\ge 0 \end{aligned} \tag{20}$$

and

$$\begin{aligned} \min_{\boldsymbol{\beta},\boldsymbol{\mu},b}\ & \frac{1}{2}\left[\lambda\boldsymbol{\beta}^T\boldsymbol{K}\boldsymbol{\beta}+\sum_{i=1}^{n}\max(0,1-y_i(\boldsymbol{K}^i\boldsymbol{\beta}+b))^2\right] \\ s.t.\ & \boldsymbol{K}=\sum_{j=1}^{M}\mu_j\boldsymbol{K}_j,\ \boldsymbol{K}^i=\sum_{j=1}^{M}\mu_j\boldsymbol{K}_j^i \\ & |\mu_j|\le c,\quad j=1,\ldots,M \end{aligned} \tag{21}$$

Let $\boldsymbol{z}=\left[\boldsymbol{\beta}^T\ b\right]^T$. Different from Primal IKSVM, here we have to optimize two variables the combination coefficients $\boldsymbol{\mu}$ and SVM parameters $\boldsymbol{z}$. Notice that one difference between the models (20) and (21) is the constraints on μ_j. When μ_j is fixed, the two problems will

degenerate to the same single kernel model. Therefore, here we utilize a two-stage algorithm to find the two variables alternately. The global Primal MIKSVM algorithm can be summarized in Algorithm 2, where the stopping criterion is the same as the one in Algorithm 1.

In each iteration, when fixing the coefficients $\boldsymbol{\mu}$, we use Algorithm 1 to solve the SVM parameters $\boldsymbol{z}$. When fixing $\boldsymbol{z}$, we employ the projected gradient descent method to solve the optimal $\boldsymbol{\mu}$.

Algorithm 2 Primal multiple indefinite kernel SVM (Primal MIKSVM)

Initial: $z^{(0)} = \mathbf{0} \in \boldsymbol{R}^{(n+1)}$, $\boldsymbol{d}^{(0)} = -\nabla^{(0)} = -[\boldsymbol{y}; \text{sum}(\boldsymbol{y})]$, $\mu_j^{(0)} = 1/M$, $t = 0$
while stopping criterion not met **do**
  $t = t + 1$;
  Find the optimal step s^* by exact Newton line search:

$$s^* = \min_s \ F(\boldsymbol{z}^{(t-1)} + s \times \boldsymbol{d}^{(t-1)});$$

  Update $\boldsymbol{z}^{(t)} = \boldsymbol{z}^{(t-1)} + s^* \times \boldsymbol{d}^{(t-1)}$;
  Optimize the combination coefficients $\boldsymbol{\mu}$;
  Compute $\boldsymbol{K} = \sum_{j=1}^{M} \mu_j K_j$;
  Compute the descent direction $\boldsymbol{d}^{(t)}$;
end while

While the gradient

$$\nabla_{\mu_j} = \frac{\lambda}{2}\boldsymbol{\beta}^T \boldsymbol{K}\boldsymbol{\beta} + \sum_{i \in \text{SV}} (1 - y_i(\boldsymbol{K}^i\boldsymbol{\beta} + b))(-y_i \boldsymbol{K}_j^i \boldsymbol{\beta})$$

has been computed, $\boldsymbol{\mu}$ is updated by the gradient descent. In order to achieve faster optimization, we further utilize Nesterov's optimal gradient algorithm [21] to accelerate the gradient descent.

Since the feasible sets of $\boldsymbol{\mu}$ are different in Primal MIKSVM-1 and Primal MIKSVM-2, here we use different strategies to ensure the descent direction projecting in the appropriate feasible set in each iteration. More specifically, Primal MIKSVM-1 executes the simplex constraint on $\boldsymbol{\mu}$, and thus l_1-ball projection method [11] is used as shown in Algorithm 3.

Algorithm 3 Simplex projection

Input: A vector $\boldsymbol{\mu}^{(t)} \in \boldsymbol{R}^M$
  Sort $\boldsymbol{\mu}^{(t)}$ into $\boldsymbol{v}$ such that $v_1 \geq v_2 \cdots \geq v_M$;
  Find $\rho = \max\{i \in [1:M] : v_i - 1/i \cdot (\sum_{r=1}^{i} v_r - 1) \geq 0\}$;
  Compute $\theta = 1/\rho \cdot (\sum_{r=1}^{i} v_r - 1)$;
Output $\boldsymbol{\mu}_{\text{new}_i}^{(t)}$: $\boldsymbol{\mu}_{\text{new}_i}^{(t)} = \max(0, \boldsymbol{\mu}_i^{(t)} - \theta), i \in [1:M]$

Correspondingly, Primal MIKSVM-2 implements the box constraint on $\boldsymbol{\mu}$ which is simpler than that in Primal MIKSVM-1. We check the elements one by one to guarantee them satisfying the absolute value constraint, as depicted in Algorithm 4.

Complexity Analysis As in Algorithm 1, the exact Newton line search for the optimal step leads to a complexity of $O(n^2)$. The optimization of the combination coefficients $\boldsymbol{\mu}$ by projected gradient descent has the complexity $O(Mn^2)$. The computation of the SVM

Algorithm 4 Box projection

Input: A vector $\boldsymbol{\mu}^{(t)} \in \boldsymbol{R}^M$
  Check every element of $\boldsymbol{\mu}^{(t)}$;
    if $\boldsymbol{\mu}_i^{(t)} \geq 0$
      let $\boldsymbol{\mu}_{\text{new}_i}^{(t)} = \min(c, \boldsymbol{\mu}_i^{(t)})$;
    else
      let $\boldsymbol{\mu}_{\text{new}_i}^{(t)} = \max(-c, \boldsymbol{\mu}_i^{(t)})$;
Output: $\boldsymbol{\mu}_{\text{new}_i}^{(t)}$

parameters $\boldsymbol{z}$ by conjugate gradient is with a complexity of $O(n^2)$. As a result, the complexity of the whole algorithm is at most $O(Mn^2)$.

5 Algorithm Analysis on Kernel

In common kernel algorithms, the kernel is usually selected by cross-validation from a series of candidate kernels. In this section, we will provide a constructive approach about how to select an appropriate kernel in our algorithms. Due to the particularity of indefinite kernels which dissatisfy the common Mercer's conditions, many state-of-the-art theoretical conclusions about PD kernel methods to being invalid in indefinite kernel models. However, from a broader perspective, both PD and indefinite kernels can actually be regarded as similarity functions. As a result, a more general theory of learning with similarity functions [5] can provide us a feasible way to select kernels effectively.

The theory starts from what a "good similarity function" for a given learning problem. Without loss of generality, here we consider the pairwise similarity function $K(\boldsymbol{x}, \boldsymbol{x}')$ mapping pairs of samples to numbers in the range $[-1, 1]$. Intuitively, K is an (ϵ, γ)-*good similarity function* for a learning problem $\boldsymbol{P}$ if at least a $1 - \epsilon$ probability mass of examples $\boldsymbol{x}$ satisfy

$$\boldsymbol{E}_{\boldsymbol{x}' \sim \boldsymbol{P}}[K(\boldsymbol{x}, \boldsymbol{x}')|y(\boldsymbol{x}) = y(\boldsymbol{x}')] \geq \boldsymbol{E}_{\boldsymbol{x}' \sim \boldsymbol{P}}[K(\boldsymbol{x}, \boldsymbol{x}')|y(\boldsymbol{x}) \neq y(\boldsymbol{x}')] + \gamma \tag{22}$$

Balcan et al. further presented a formal definition on a "*good similarity function*" in the first-order hinge loss [5]. However, due to the non-convexity of our models, here we adopt the smooth quadratic hinge loss in the objective functions out of easily being optimized. So we firstly generalize the definition from the first-order hinge loss to the quadratic one as follows

Definition 1 A similarity function K is an (ϵ, γ)-*good similarity function* in quadratic hinge loss for $\boldsymbol{P}$ if there exists a weighting function $w(\boldsymbol{x}') \in [0, 1]$ for all $\boldsymbol{x}' \in \mathcal{X}$ such that

$$\boldsymbol{E}_{\boldsymbol{x}}[[1 - y(\boldsymbol{x})g(\boldsymbol{x})/\gamma]_+^2] \leq \epsilon \tag{23}$$

where $g(\boldsymbol{x}) = \boldsymbol{E}_{\boldsymbol{x}' \sim \boldsymbol{P}}[y(\boldsymbol{x}')w(\boldsymbol{x}')K(\boldsymbol{x}, \boldsymbol{x}')]$ is the similarity-based prediction made using $w(\cdot)$, and $[1 - z]_+^2 = \max(0, 1 - z)^2$ is the quadratic hinge loss.

For a single kernel, an (ϵ, γ)-*good similarity function* K can ensure that given a certain amount of samples, there exists a separator with low-error and large margin in the space induced by K.

Theorem 2 (Single Kernel) *Let K be an* (ϵ, γ)-good similarity function *in quadratic hinge loss for $\boldsymbol{P}$. For any $\epsilon_1 > 0$ and $0 < \delta < 3\epsilon_1^2\gamma^2/16$, let $\boldsymbol{S} = \{\hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_n\}$ be a sample set*

with the size $n = 16\log(1/\delta)/(\epsilon_1\gamma)^2$ *drawn from* $\boldsymbol{P}$. *Consider the mapping* $\phi^s : \mathcal{X} \to \boldsymbol{R}^n$ *defined as follows*

$$\phi_i^s(\boldsymbol{x}) = K(\boldsymbol{x}, \hat{\boldsymbol{x}}_i)/\sqrt{n},\ i \in \{1, \cdots, n\}$$

With probability at least $1-\delta$ *over the random sample* $\boldsymbol{S}$, *the induced distribution* $\phi^S(\boldsymbol{P})$ *in* $\boldsymbol{R}^n$ *has a separator achieving hinge-loss at most* $\epsilon + \epsilon_1^2$ *at margin* γ.

Theorem 2 extends Theorem 4 in [5] to the quadratic hinge loss scenario. Assume that the kernel K is an (ϵ, γ)-*good similarity function*, the theorem has shown that we more likely obtain (with probability at least $1-\delta$) a predictor in the ϕ^S-space induced by K with error rate $\epsilon + \epsilon_1^2$. A subsequent problem arises naturally: how to find such a kernel satisfied that it is an (ϵ, γ)-*good similarity function*.

In fact, following the proof of Theorem 4 in [5], let $w(\boldsymbol{x}_i') = \sqrt{n}\beta_i / y(\boldsymbol{x}_i')$, we have

$$g(\boldsymbol{x}) = \boldsymbol{E}_{\boldsymbol{x}'\sim\boldsymbol{P}}[y(\boldsymbol{x}')w(\boldsymbol{x}')K(\boldsymbol{x}, \boldsymbol{x}')] = \sum_{i=1}^{n} \sqrt{n} p_i \beta_i K(\boldsymbol{x}, \boldsymbol{x}_i')$$

where p_i is a prior distribution for $\boldsymbol{x}_i'$, $\sum_{i=1}^n p_i = 1$. We uniformly set $p_i = 1/n$, then

$$g(\boldsymbol{x}) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \beta_i K(\boldsymbol{x}, \boldsymbol{x}_i') = \frac{1}{\sqrt{n}} \boldsymbol{K}^i \boldsymbol{\beta} \tag{24}$$

For a given problem $\boldsymbol{P}$, substituting (24) into (23), we obtain

$$\boldsymbol{E}_{\boldsymbol{x}\sim\boldsymbol{P}}[[1 - y(\boldsymbol{x})g(\boldsymbol{x})/\gamma]_+^2] = \sum_{i=1}^{n} p_i' \max(0, 1 - \frac{1}{\sqrt{n}} y_i \boldsymbol{K}^i \boldsymbol{\beta}/\gamma)^2$$

Let $\gamma = 1/\sqrt{n}$ and $p_i' = 1/n$, then

$$\boldsymbol{E}_{\boldsymbol{x}\sim\boldsymbol{P}}[[1 - y(\boldsymbol{x})g(\boldsymbol{x})/\gamma]_+^2] = 1/n \sum_{i=1}^{n} \max(0, 1 - y_i \boldsymbol{K}^i \boldsymbol{\beta})^2$$

So far, the problem has been transformed into looking for an appropriate kernel to satisfy the value of the quadratic hinge loss less than a small threshold ϵ. However, for various learning problems, ϵ is actually difficult to determine in advance. If ϵ is set larger, there are more likely many kernels satisfied the loss less than ϵ. On the contrary, if ϵ is set smaller, it is possibly hard to find a suitable kernel especially in indefinite kernel scenarios. So in our experiments, we turn to take another approach instead of setting ϵ beforehand.

Specifically in single kernel scenarios, given a candidate indefinite kernel set, we use Algorithm 1 Primal IKSVM to train a classifier for each candidate kernel. When our stopping criterion arrives, the difference between the values of the quadratic hinge loss in two adjacent iterations is already very small, that is to say, the value of the quadratic hinge loss is relatively stable. Considering the non-convexity of our model, we run the algorithm by ten-fold cross-validation. In each fold, the smallest value of the quadratic hinge loss is set to be our ϵ, and then a possible "best" kernel is determined by ϵ. After ten folds, the final kernel is selected from the ten possible "best" kernels by voting.

In multiple kernels scenarios, we adopt the similar strategy to learn the combination coefficients and thus the obtained kernel combination $K = \sum_{j=1}^{M} \mu_j K_j$ is more likely an (ϵ, γ)-*good similarity function*. We can easily generalize the similar conclusion in Theorem 2 to Primal MIKSVM-1 which emphasizes on the convex combination of indefinite base kernels $K_1, \cdots, K_M$, and further analyze its generalization performance.

Theorem 3 (Convex Combination of Multiple Kernels) *Suppose* $K_1, \cdots, K_M$ *are similarity functions such that some (unknown) convex combination of them* $K = \sum_{j=1}^{M} \mu_j K_j$ *is* (ϵ, γ)-good *in quadratic hinge loss. If one draws a set* $\boldsymbol{S} = \{\hat{x}_1, \cdots, \hat{x}_n\}$ *from* $\boldsymbol{P}$ *containing* $n = 16\log(1/\delta)/(\epsilon_1\gamma)^2$ *instances, then with probability at least* $1 - \delta$*, the mapping* $\phi^{\boldsymbol{S}} : \mathcal{X} \to \boldsymbol{R}^{Mn}$ *defined as* $\phi^{\boldsymbol{S}}(\boldsymbol{x}) = \rho^{\boldsymbol{S}}(\boldsymbol{x})/\sqrt{Mn}$, $\rho^{\boldsymbol{S}}(\boldsymbol{x}) = (K_1(\boldsymbol{x}, \hat{\boldsymbol{x}}_1), \cdots, K_1(\boldsymbol{x}, \hat{\boldsymbol{x}}_n), \cdots, K_M(\boldsymbol{x}, \hat{\boldsymbol{x}}_1), \cdots, K_M(\boldsymbol{x}, \hat{\boldsymbol{x}}_n)$ *has the property that the induced distribution* $\phi^{\boldsymbol{S}}(\boldsymbol{P})$ *in* $\boldsymbol{R}^{Mn}$ *has a separator achieving hinge-loss at most* $\epsilon + \epsilon_1^2$ *at margin* $\gamma/(\|\boldsymbol{\mu}\|\sqrt{M})$.

Different from Primal MIKSVM-1, Primal MIKSVM-2 considers the non-convex combination of PD base kernels. Without loss of generality, we restrict the absolute value of each combination coefficient yielding $|\mu_j| \leq 1/M$, $j = 1, \cdots, M$ and obtain Theorem 4 as below

Theorem 4 (Non-convex Combination of Multiple Kernels) *Suppose* $K_1, \cdots, K_M$ *are similarity functions such that some* linear *combination of them* $K = \sum_{j=1}^{M} \mu_j K_j$ *is* (ϵ, γ)-good *in quadratic hinge loss, where* $|\mu_j| \leq 1/M$, $j = 1, \cdots, M$. *If one draws a set* $\boldsymbol{S} = \{\hat{x}_1, \cdots, \hat{x}_n\}$ *from* $\boldsymbol{P}$ *containing* $n = 16\log(1/\delta)/(\epsilon_1\gamma)^2$ *instances, then with probability at least* $1 - \delta$*, the mapping* $\phi^{\boldsymbol{S}} : \mathcal{X} \to \boldsymbol{R}^{Mn}$ *defined as* $\phi^{\boldsymbol{S}}(\boldsymbol{x}) = \rho^{\boldsymbol{S}}/\sqrt{(Mn)}$, $\rho^{\boldsymbol{S}}(\boldsymbol{x}) = (K_1(\boldsymbol{x}, \hat{\boldsymbol{x}}_1), \cdots, K_1(\boldsymbol{x}, \hat{\boldsymbol{x}}_n), \cdots, K_M(\boldsymbol{x}, \hat{\boldsymbol{x}}_1), \cdots, K_M(\boldsymbol{x}, \hat{\boldsymbol{x}}_n)$ *has the property that the induced distribution* $\phi^{\boldsymbol{S}}(\boldsymbol{P})$ *in* $\boldsymbol{R}^{Mn}$ *has a separator achieving hinge-loss at most* $\epsilon + \epsilon_1^2$ *at margin* γ.

Proof Performing the mapping $\hat{\phi}^{\boldsymbol{S}}(\boldsymbol{x}) : \mathcal{X} \to \boldsymbol{R}^n$, defined as $\hat{\phi}^{\boldsymbol{S}}(\boldsymbol{x}) = \hat{\rho}^{\boldsymbol{S}}(\boldsymbol{x})/\sqrt{n}$, $\hat{\rho}^{\boldsymbol{S}} = (K(\boldsymbol{x}, \hat{\boldsymbol{x}}_1), \cdots, K(\boldsymbol{x}, \hat{\boldsymbol{x}}_n))$, where $K(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sum_{j=1}^{M} \mu_j K_j(\boldsymbol{x}, \hat{\boldsymbol{x}})$.

Following Theorem 2, with probability $1-\delta$, the induced distribution $\hat{\phi}^{\boldsymbol{S}}(\boldsymbol{x})$ in $\boldsymbol{R}^n$ would have a separator achieving hinge-loss at most $\epsilon + \epsilon_1^2$ at margin at least γ. Let $\hat{\boldsymbol{\beta}} \in \boldsymbol{R}^n$ be the vector corresponding to such a separator. So with probability $1 - \delta$, we have

$$\boldsymbol{E}_{\boldsymbol{x}\sim\boldsymbol{P}}[[1 - y(\boldsymbol{x})g(\boldsymbol{x})/\gamma]_+^2] \leq \epsilon + \epsilon_1^2$$

In other words, with probability $1 - \delta$, the separator follows

$$\boldsymbol{E}_{\boldsymbol{x}\sim\boldsymbol{P}}[y(\boldsymbol{x})\left\langle\hat{\boldsymbol{\beta}}, \hat{\phi}^{\boldsymbol{S}}(\boldsymbol{x})\right\rangle \geq \gamma] > 1 - (\epsilon + \epsilon_1^2) \tag{25}$$

where $\|\hat{\beta}\| \leq 1$, $\|\hat{\phi}^{\boldsymbol{S}}(\boldsymbol{x}) \leq 1\|$.

Now let us convert $\hat{\boldsymbol{\beta}}$ into a vector in $\boldsymbol{R}^{Mn}$ by replacing each coordinate $\hat{\beta}_i$ with the M values $(\mu_1\hat{\beta}_i, \cdots, \mu_M\hat{\beta}_i)$. Denote the resulting vector as $\bar{\boldsymbol{\beta}}$. Note that for any $\boldsymbol{x}$, we have

$$\left\langle\bar{\beta}, \phi^{\boldsymbol{S}}(\boldsymbol{x})\right\rangle = 1/\sqrt{M} \cdot \left\langle\hat{\boldsymbol{\beta}}, \hat{\phi}^{\boldsymbol{S}}(\boldsymbol{x})\right\rangle \tag{26}$$

We substitute (26) into the inequality (25). So with probability $1-\delta$, the separator in $\boldsymbol{R}^{Mn}$ has

$$\boldsymbol{E}_{\boldsymbol{x}\sim\boldsymbol{P}}[y(\boldsymbol{x})\left\langle\bar{\boldsymbol{\beta}}, \phi^{\boldsymbol{S}}(\boldsymbol{x})\right\rangle \geq \gamma/\sqrt{M}] > 1 - (\epsilon + \epsilon_1^2)$$

For $\|\bar{\boldsymbol{\beta}}\| = \|\boldsymbol{\mu}\|\|\hat{\boldsymbol{\beta}}\| = \|\hat{\boldsymbol{\beta}}\|\sqrt{\sum_{j=1}^{M}\mu_j^2} \leq \|\hat{\boldsymbol{\beta}}\|\sqrt{\sum_{j=1}^{M}1/M^2} = \|\hat{\boldsymbol{\beta}}\|/\sqrt{M} \leq 1/\sqrt{M}$ and $\|\phi^{\boldsymbol{S}}(\boldsymbol{x})\| \leq 1$, we further obtain

$$\boldsymbol{E}_{\boldsymbol{x}\sim\boldsymbol{P}}\left[\frac{y(\boldsymbol{x})\left\langle\bar{\boldsymbol{\beta}}, \phi^{\boldsymbol{S}}(\boldsymbol{x})\right\rangle}{\|\bar{\boldsymbol{\beta}}\|\|\phi^{\boldsymbol{S}}(\boldsymbol{x})\|} \geq \gamma\right] > 1 - (\epsilon + \epsilon_1^2)$$

That is, with probability $1-\delta$, the separator in $\boldsymbol{R}^{Mn}$ actually has

$$\boldsymbol{E}_{\boldsymbol{x}\sim\boldsymbol{P}}\left[[1-\frac{y(\boldsymbol{x})\left\langle\bar{\boldsymbol{\beta}},\phi^{\boldsymbol{S}}(\boldsymbol{x})\right\rangle}{\|\bar{\boldsymbol{\beta}}\|\|\phi^{\boldsymbol{S}}(\boldsymbol{x})\|}/\gamma]_+^2\right]\leq\epsilon+\epsilon_1^2$$

In conclusion, the induced $\phi^{\boldsymbol{S}}(\boldsymbol{P})$ in $\boldsymbol{R}^{Mn}$ has a separator achieving hinge-loss at most $\epsilon+\epsilon_1^2$ at margin γ.

6 Experiments

To evaluate the effectiveness of the proposed models, we perform a series of experiments systematically on several real world classification problems, including some common datasets from UCI, IDA and USPS databases, and more complicated bioinformatics datasets. All the experiments are performed on a server with Xeon(R) X5460 3.16GHz processor and 32766MB RAM.

6.1 Experiments on Common Datasets

Ten common-used datasets are used for experiments, including two datasets Wdbc (569, 30) and Spambase (4601, 57) from UCI Machine Learning Repository, seven datasets Diabetis (768, 8), German (1000, 20), Titanic (2201, 3), Image (2310, 18), Waveform (5000, 21), Banana (5300, 2), Ringnorm (7400, 20) from IDA database [45], and one handwritten digits dataset USPS (11000, 256), where the number and dimension of samples are listed in the bracket.

For the UCI and IDA datasets, we randomly divide the samples into two non-overlapping training and testing sets which contain almost half of samples in each class. For the USPS dataset, we classify odd vs. even digits, and randomly select 50 samples in each digit to form training set as well as the remaining as testing set. The processes are repeated twenty times to generate twenty independent runs for each dataset, and then the average results are reported.

We firstly compare the single kernel models, that is, the proposed Primal IKSVM with classical PD kernel SVM and five popular indefinite methods Clip [43], Flip [15], Shift [47], Dual IKSVM [7] and ESVM [35]. SVM is also used to the subsequent classifier in Clip, Flip and Shift. The parameter λ is fixed to 0.01, which refers to the setting in SimpleMKL [44]. The indefinite Sigmoid and Gausscombination kernels [38] are applied, whose candidate kernel parameter sets are $\{2^{-6}, 2^{-5}, ..., 2^5, 2^6\}$ and $\{0.5, 1, 2, 5, 7, 10, 12, 15, 17, 20\}$ as in [44] respectively. The PD Gaussian kernel is used in SVM whose candidate kernel parameter set is the same as Gausscombination kernel's.

Table 2 reports the detailed experimental results of the seven algorithms. On each dataset, the mean classification accuracies as well as the standard deviations of each algorithm are recorded, where the best results are highlighted in bold and italic face. Their average accuracies and standard deviations on all datasets are also reported. Furthermore, to statistically measure the significance of performance difference, pairwise t tests at 95% significance level are conducted between the algorithms. Specifically, whenever Primal IKSVM achieves significantly better/worse performance than the compared algorithm on any dataset, a win/loss is counted and a marker $\bullet/\circ$ is shown. Otherwise, a tie is counted and no marker is given. The resulting win/tie/loss counts for Primal IKSVM against the compared algorithms are provided in the last line of the table.

From the table, we can see that Clip, Flip and Shift perform poorly on most datasets, which indicates that the simple spectrum transformation in fact can not solve the complex indefinite kernel problems effectively. Dual IKSVM seeks a proxy PD kernel to approximate the indefinite one and can achieve better performance than Clip, Flip and Shift on some datasets, such as Diabetis and German. But it is still worse than Primal IKSVM on most datasets, specifically its accuracies are lower than Primal IKSVM's close to 20% on the Image and Banana datasets. ESVM tries to obtain a stationary point in the Kreĭn space. Primal IKSVM is superior to ESVM on most UCI and IDA datasets. Especially on USPS dataset, the classification accuracy of IKSVM is higher than that of ESVM close to 21%. Primal IKSVM directly solves the non-convex indefinite kernel SVM problem from the primal and thus precedes the compared indefinite kernel algorithms on five datasets, whose average accuracy excels the other ones beyond 3%. Furthermore, it ties with classical PD kernel SVM. The corresponding statistical t tests verify our conclusions.

We further evaluate the proposed two multiple indefinite kernel models Primal MIKSVM-1 and Primal MIKSVM-2 on the datasets. Due to the scarcity of related works, we compare them with two successful multiple PD kernel methods SimpleMKL [44] and PrimalMKL [21], and one multiple indefinite kernel method Mixnorm [29]. For SimpleMKL, PrimalMKL and Primal MIKSVM-2, the PD Gaussian and Polynomial kernels are used as base kernels, where the candidate kernel parameter sets are {0.5, 1, 2, 5, 7, 10, 12, 15, 17, 20} and {1, 2, 3} respectively. Mixnorm and Primal MIKSVM-1 utilize both the PD Gaussian and Polynomial kernels and the indefinite Sigmoid and Gausscombination kernels as base kernels. The positive constant c in Primal MIKSVM-2 is set to 1. Another parameter settings are the same as the above single kernel scenarios.

Table 3 presents the average classification results of the compared algorithms. On most datasets, Primal MIKSVM-1 and Primal MIKSVM-2 significantly outperform the PD kernel methods SimpleMKL and PrimalMKL. Especially on the Image and USPS datasets, their accuracies exceed the ones of SimpleMKL and PrimalMKL beyond 5%. Primal MIKSVM-1 and Primal MIKSVM-2 also greatly excel Mixnorm, whose accuracies are sometimes even worse than those of the two models over 20%. Furthermore, comparing Primal MIKSVM-1 and Primal MIKSVM-2 themselves, their average accuracies are basically comparable except on the USPS dataset. The reason may be that the data in USPS are more suitable for the indefinite Sigmoid kernel, which further validates the dominance of using indefinite kernels in real world applications.

6.2 Experiments on Bioinformatics Datasets

To further investigate the effectiveness of our models, we verify their performance on more complicated bioinformatics datasets, including two DNA datasets DNA_large and DNA_small [44], and three biological datasets PSortPos, PSortNeg and Plant [30] (originally in [59]). They are all multi-class datasets which contain either three, four or five classes. For each dataset, we select two balanced classes for experiment. As above, we randomly divide the samples into two non-overlapping training and testing sets which have almost half of samples in each class. The kernel selections and parameter settings are the same as above. Specially, for three biological datasets, we use the 69 sequence kernels given in the database where more than half of them are indefinite kernels.

The classification results are shown in Tables 4 and 5 corresponding to single and multiple kernel scenarios respectively. In the single kernel scenarios, Primal IKSVM statistically wins the other six algorithms on most datasets. In particular, its accuracy is ahead of the

Table 2 Classification result (mean ± SD) of single kernel models on the ten datasets

Dataset	Classification accuracy						
	Clip	Flip	Shift	Dual IKSVM	SVM	ESVM	Primal IKSVM
Wdbc	94.99 ± 0.16•	95.93 ± 0.07•	94.72 ± 0.03•	91.97 ± 3.24•	94.25 ± 0.53•	96.04 ± 0.01•	**97.12** ± 0.14
Spambase	89.31 ± 0.38	82.88 ± 0.87•	83.09 ± 1.02•	81.02 ± 0.89•	**91.55** ± 0.44○	90.83 ± 0.01	90.10 ± 1.02
Diabetis	71.33 ± 4.97•	75.40 ± 1.27•	75.38 ± 0.68•	76.30 ± 2.55	76.47 ± 1.01	76.22 ± 0.02	**76.97** ± 0.10
German	71.42 ± 0.20•	72.28 ± 0.16•	73.28 ± 0.05•	**76.08** ± 0.25	75.60 ± 0.17	72.12 ± 0.02•	75.56 ± 0.14
Titanic	74.89 ± 0.26•	78.37 ± 0.02	76.44 ± 0.19•	77.95 ± 0.18	77.58 ± 0.36•	**78.82** ± 0.01	78.32 ± 0.09
Image	71.80 ± 1.33•	72.30 ± 0.41•	73.07 ± 0.04•	74.27 ± 0.27•	93.60 ± 0.41○	**94.12** ± 0.01○	92.81 ± 0.67
Waveform	81.98 ± 0.82•	88.93 ± 0.18•	88.36 ± 0.12•	87.29 ± 0.51•	90.12 ± 0.36•	89.67 ± 0.01•	**91.26** ± 0.15
Banana	68.22 ± 1.08•	71.50 ± 0.92•	58.59 ± 0.29•	65.41 ± 2.64•	85.47 ± 0.40○	**86.99** ± 0.02○	84.08 ± 2.71
Ringnorm	96.42 ± 0.17•	96.15 ± 0.02•	86.76 ± 0.44•	89.95 ± 0.56•	97.55 ± 0.13	97.11 ± 0.01	**97.65** ± 0.05
USPS	86.45 ± 0.02•	47.68 ± 1.42•	83.32 ± 0.09•	83.98 ± 0.28•	87.02 ± 0.16	66.35 ± 0.17•	**87.30** ± 0.05
Average	80.68 ± 0.94	78.14 ± 0.53	79.30 ± 0.30	80.42 ± 1.14	86.92 ± 0.40	84.83 ± 0.03	**87.12** ± 0.51
Win/tie/loss	9/1/0	9/1/0	10/0/0	7/3/0	3/4/3	4/4/2	/

• Our model is significantly better than the corresponding methods on the criterion based on the t test at 95% significance level

Table 3 Classification result (mean ± SD) of multiple kernel models on the ten datasets

Dataset	Classification accuracy				
	SimpleMKL	PrimalMKL	Mixnorm	Primal MIKSVM-1	Primal MIKSVM-2
Wdbc	95.97 ± 0.11•	96.54 ± 0.12•	90.40 ± 0.20•	96.98 ± 0.09	**97.52** ± 0.09
Spambase	88.69 ± 0.08•	90.10 ± 0.07•	75.20 ± 0.65•	91.84 ± 0.06	**93.84** ± 0.05
Diabetis	75.53 ± 0.18•	76.88 ± 0.17	72.00 ± 0.36•	**77.02** ± 0.17	75.89 ± 0.20
German	70.07 ± 0.03•	74.09 ± 0.15•	70.80 ± 0.14•	**75.43** ± 0.10	73.52 ± 0.20
Titanic	78.09 ± 0.07•	77.41 ± 0.08•	77.10 ± 0.51•	78.77 ± 0.06	**79.01** ± 0.07
Image	87.65 ± 0.09•	88.67 ± 0.13•	75.60 ± 0.35•	92.44 ± 0.10	**96.76** ± 0.05
Waveform	89.96 ± 0.08•	90.64 ± 0.04•	70.60 ± 0.19•	**91.34** ± 0.05	90.61 ± 0.05
Banana	89.99 ± 0.05	89.53 ± 0.19	81.60 ± 0.36•	90.37 ± 0.05	**90.39** ± 0.04
Ringnorm	98.26 ± 0.02	97.94 ± 0.05•	74.60 ± 0.31•	**98.51** ± 0.02	98.39 ± 0.02
USPS	87.66 ± 0.06•	87.01 ± 0.07•	65.38 ± 0.26•	**92.07** ± 0.13	81.01 ± 0.64
Average	86.19 ± 0.08	86.88 ± 0.11	75.33 ± 0.33	**88.48** ± 0.08	87.69 ± 0.14
Win/tie/loss	8/2/0	8/2/0	10/0/0	/	/

Bold indicates best performance

• Our model is significantly better than the corresponding methods on the criterion based on the t test at 95% significance level

other ones close to 3% on the PSN dataset. Furthermore, in the multiple kernel scenarios, Primal MIKSVM-1 and Primal MIKSVM-2 also show much better performance than the compared algorithms, where their superiority is more significant on three biological datasets. Concretely, unlike SimpleMKL and PrimalMKL that only use PD kernels, Primal MIKSVM-1 can utilize both PD and indefinite base kernels given in the database effectively and thus possess the best accuracies on all the three datasets. Especially on the PSP and PSN datasets, it excels the two PD kernel methods close to 5%. Mixnorm can also employ all base kernels, however, it still performs poorly due to the instability of the algorithm itself. Primal MIKSVM-2 is likewise worse than Primal MIKSVM-1, since that it is in nature designed to solve the non-convex combination of PD kernels and thus only uses the given PD base kernels which are far from enough to characterize the samples in the complex biological problems.

We further empirically evaluate the convergence properties of our proposed models. In each dataset, we plot the variations on the classification accuracies and objective values of the formulations in (13), (20) and (21). The corresponding plots are presented in Figs. 1, 2 and 3. From the figures, we can see that both classification accuracies and objective values in the three models gradually converge to stable values within twenty iterations. This more likely suggests that the iterative algorithms can reach local minima yielding acceptable classification accuracies.

In order to investigate the practical effects of indefinite kernels in multiple kernel scenarios more clearly, we also compare the sums of combination coefficients in Primal MIKSVM-1 and Primal MIKSVM-2. The results are shown in Fig. 4. On one hand, Primal MIKSVM-1 considers the convex combination of base kernels which can be both PD and indefinite kernels, and thus the sum of combination coefficients corresponding to different kernels is equal to 1. From Fig. 4a, it is obvious that indefinite base kernels predominate in the kernel combinations on all the datasets, which indicates that they actually count more with the learning problems than PD base kernels. Moreover, combining the superiority of Primal

Table 4 Classification result (mean ± SD) of single kernel models on the bioinformatics datasets

Dataset	Classification accuracy						
	Clip	Flip	Shift	Dual IKSVM	SVM	ESVM	Primal IKSVM
DNA_large	80.05 ± 0.07•	80.30 ± 0.06•	85.43 ± 0.04•	93.15 ± 0.45•	94.01 ± 0.44	94.76 ± 0.01	**94.98** ± 0.06
DNA_small	87.19 ± 0.07•	86.10 ± 0.08•	83.72 ± 0.07•	90.60 ± 0.42	**91.96** ± 0.78	91.95 ± 0.01	91.15 ± 0.08
PSP	70.00 ± 0.40•	76.54 ± 0.94•	80.80 ± 0.52•	86.49 ± 0.84•	87.12 ± 0.46•	89.87 ± 0.02	**90.24** ± 0.16
PSN	78.90 ± 0.30•	79.31 ± 0.20•	81.64 ± 0.24•	85.21 ± 0.23•	85.70 ± 0.59•	87.73 ± 0.02•	**90.00** ± 0.13
P	73.01 ± 0.90•	73.25 ± 0.86•	89.23 ± 0.20•	85.36 ± 0.66•	73.59 ± 0.65•	88.61 ± 0.03•	**90.72** ± 0.16
Average	77.83 ± 0.35	79.10 ± 0.43	84.16 ± 0.21	88.16 ± 0.52	86.48 ± 0.58	90.58 ± 0.02	**91.42** ± 0.12
Win/tie/loss	5/0/0	5/0/0	5/0/0	4/1/0	3/2/0	2/3/0	/

Bold indicates best performance

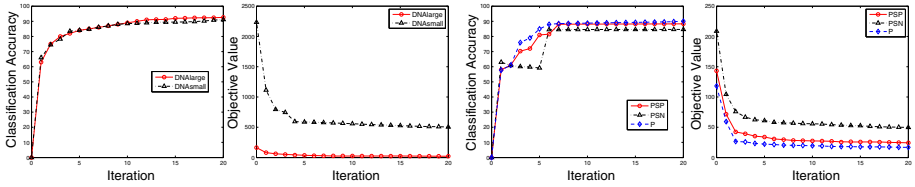
• Our model is significantly better than the corresponding methods on the criterion based on the t test at 95% significance level

Table 5 Classification result (mean ± SD) of multiple kernel models on the bioinformatics datasets

Dataset	Classification accuracy				
	SimpleMKL	PrimalMKL	Mixnorm	Primal MIKSVM-1	Primal MIKSVM-2
DNA_large	94.97 ± 0.05•	95.23 ± 0.46	76.49 ± 0.40•	95.51 ± 0.48	**95.79** ± 1.17
DNA_small	94.06 ± 0.05	92.87 ± 0.57•	76.42 ± 0.55•	**94.28** ± 0.61	93.11 ± 2.19
PSP	86.91 ± 0.30•	76.29 ± 0.68•	76.60 ± 0.45•	**90.30** ± 0.74	88.14 ± 0.39
PSN	82.35 ± 0.21•	89.15 ± 0.30•	58.80 ± 0.29•	**93.09** ± 0.42	84.04 ± 0.54
P	66.39 ± 0.31•	87.29 ± 0.41•	69.40 ± 0.55•	**91.34** ± 0.34	81.39 ± 1.55
Average	84.94 ± 0.18	88.17 ± 0.48	71.54 ± 0.45	**92.90** ± 0.52	88.49 ± 1.17
Win/tie/loss	4/1/0	4/1/0	5/0/0	/	/

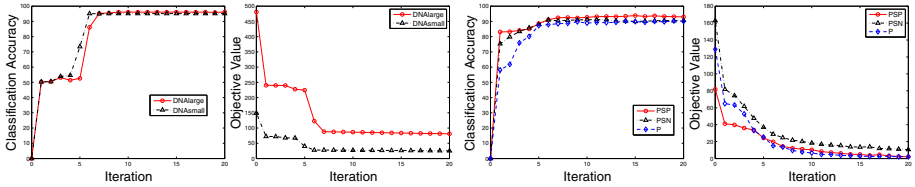
Bold indicates best performance

• Our model is significantly better than the corresponding methods on the criterion based on the t test at 95% significance level



(a) DNA datasets **(b)** Biological datasets

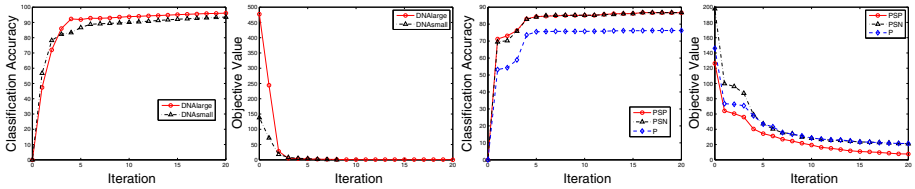
Fig. 1 Covergence analysis on Primal IKSVM on the bioinformatics datasets: **a** two DNA datasets and **b** three biological datasets



(a) DNA datasets **(b)** Biological datasets

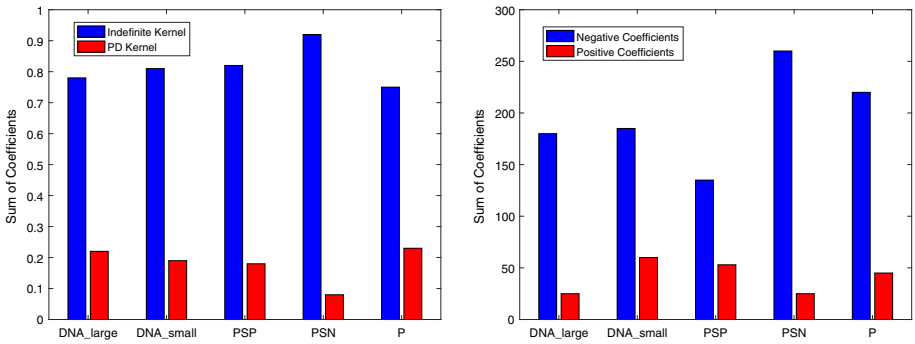
Fig. 2 Covergence analysis on Primal MIKSVM-1 on the bioinformatics datasets: **a** two DNA datasets and **b** three biological datasets

MIKSVM-1 in classification performance, it is sufficient to illustrate the effectiveness and necessity of using indefinite kernels in MKL to solve complex learning problems. On the other hand, Primal MIKSVM-2 takes the non-convex combination of PD base kernels into account, therefore the combination coefficients can be negative. We count the sums of positive and negative coefficients respectively. For negative coefficients, we report the absolute values of the sums in Fig. 4b. Although Primal MIKSVM-2 only uses PD base kernels, the kernels with negative coefficients are likewise dominant in the kernel combinations, which further validates the effectiveness of indefinite kernels.



(a) DNA datasets **(b)** Biological datasets

Fig. 3 Covergence analysis on Primal MIKSVM-2 on the bioinformatics datasets: **a** two DNA datasets and **b** three biological datasets



(a) Primal MIKSVM-1 **(b)** Primal MIKSVM-2

Fig. 4 Comparison about the sum of combination coefficients in Primal MIKSVM-1 and Primal MIKSVM-2 on the bioinformatics datasets

7 Conclusion

In this paper, we firstly analyze the reasonability of learning indefinite kernel SVM from primal problem in virtue of duality gap. Then a novel primal indefinite kernel SVM framework is constructed in the RKKS, which not only derives a single kernel model Primal IKSVM but also extends to multiple kernel scenarios. Based on different characteristics of base kernels and combination coefficients, we further propose a generalized MKL framework. From the framework, we deduce two multiple indefinite kernel models Primal MIKSVM-1 and Primal MIKSVM-2, where the former emphasizes on the convex combination of indefinite base kernels, and the latter focuses on the non-convex combination of PD base kernels. Two PRP conjugate gradient algorithms are presented to solve the non-convex optimizations in the models. Furthermore, a constructive approach for kernel selection in the algorithms is developed by virtue of the more general theory of similarity functions. Experimental results demonstrate the superiority of the proposed models in complex actual applications.

There are several directions for future study:

- *Optimization technique* In the paper, we apply conjugate gradient technique to solve the non-convex optimizations in the proposed models, which is widely used in primal kernel methods. However, these algorithms can only arrive at local minima. How to develop better non-convex optimization techniques for our models needs more systematic research.
- *Large-scale problem* In the experiments, we utilize the 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. Consequently, how to develop a fast algorithm for our models is another interesting topic for future study.

References

1. Ackermann MR, Blömer J, Sohler C (2010) Clustering for metric and nonmetric distance measures. ACM Trans Algorithms 6(4):59
2. Aiolli F, Donini M (2015) EasyMKL: a scalable multiple kernel learning algorithm. Neurocomputing 169:215–224
3. Alabdulmohsin IM, Gao X, Zhang X (2014) Support vector machines with indefinite kernels. In: Proceedings of 6th Asian conference on machine learning
4. Anzai Y (2012) Pattern recognition and machine learning. Elsevier, Amsterdam
5. Balcan MF, Blum A, Srebro N (2008) A theory of learning with similarity functions. Mach Learn 1–2:89–112
6. Chapelle O (2007) Training a support vector machine in the primal. Neural Comput 5:1155–1178
7. Chen J, Ye J (2008) Training SVM with indefinite kernels. In: Proceedings of the 25th international conference on machine learning. ACM, pp 136–143
8. Chung W, Kim J, Lee H, Kim E (2015) General dimensional multiple-output support vector regressions and their multiple kernel learning. IEEE Trans Cybern 11:2572–2584
9. Cortes C, Mohri M, Rostamizadeh A (2009) Learning non-linear combinations of kernels. In: Proceedings of 23rd conference on Advances in neural information processing systems, pp 396–404 (2009)
10. Donini M, Aiolli F (2016) Learning deep kernels in the space of dot product polynomials. Mach Learn 106:1–25
11. Duchi J, Shalev-Shwartz S, Singer Y, Chandra T (2008) Efficient projections onto the l 1-ball for learning in high dimensions. In: Proceedings of the 25th international conference on machine learning. ACM, pp 272–279 (2008)
12. Fan Q, Gao D, Wang Z (2016) Multiple empirical kernel learning with locality preserving constraint. Knowl Based Syst 105:107–118
13. Fan Q, Wang Z, Zha H, Gao D (2017) MREKLM: a fast multiple empirical kernel learning machine. Pattern Recognit 61:197–209
14. Gönen M, Alpaydın E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12(Jul):2211–2268
15. Graepel T, Herbrich R, Bollmann-Sdorra P, Obermayer K (1999) Classification on pairwise proximity data. Adv Neural Inf Process Syst 11:438–444
16. Gu S, Guo Y (2012) Learning SVM classifiers with indefinite kernels. In: Proceedings of the 27th AAAI conference on artificial intelligence
17. Haasdonk B (2005) Feature space interpretation of svms with indefinite kernels. IEEE Trans Pattern Ana Mach Intell 4:482–492
18. Haasdonk B, Pekalska E (2008) Indefinite kernel fisher discriminant. In: Proceedings of 19th international conference on pattern recognition, pp 1–4 (2008)
19. Haasdonk B, Pkalska E (2010) Indefinite kernel discriminant analysis. In: Proceedings of international conference on computational statistic. Springer, pp 221–230 (2010)
20. Han Y, Yang K, Ma Y, Liu G (2014) Localized multiple kernel learning via sample-wise alternating optimization. IEEE Trans Cybern 1:137–148
21. Hao Z, Yuan G, Yang X, Chen Z (2013) A primal method for multiple kernel learning. Neural Comput Appl 3–4:975–987
22. Hinrichs C, Singh V, Peng J, Johnson S (2012) Q-MKL: matrix-induced regularization in multi-kernel learning with applications to neuroimaging. In: Proceedings of 26th conference on Advances in neural information processing systems, pp 1421–1429 (2012)
23. Hoi SC, Jin R, Zhao P, Yang T (2013) Online multiple kernel classification. Mach Learn 2:289–316
24. Huang J, Xue H, Zhai Y (2012) Semi-supervised discriminatively regularized classifier with pairwise constraints. In: Pacific Rim international conference on artificial intelligence. Springer, pp 112–123 (2012)
25. Huang X, Maier A, Hornegger J, Suykens JA (2016) Indefinite kernels in least squares support vector machines and principal component analysis. Appl Comput Harmon Anal 43:162–172
26. Jacobs DW, Weinshall D, Gdalyahu Y (2000) Classification with nonmetric distances: Image retrieval and class representation. IEEE Trans Pattern Anal Mach Intell 6:583–600
27. Jin R, Yang T, Mahdavi M (2013) Sparse multiple kernel learning with geometric convergence rate. arXiv preprint arXiv:1302.0315

28. Kloft M, Brefeld U, Laskov P, Müller KR, Zien A, Sonnenburg S (2009) Efficient and accurate LP-norm multiple kernel learning. In: Proceedings of 23rd conference on Advances in neural information processing systems, pp 997–1005 (2009)
29. Kowalski M, Szafranski M, Ralaivola L (2009) Multiple indefinite kernel learning with mixed norm regularization. In: Proceedings of the 26th annual international conference on machine learning. ACM, pp 545–552 (2009)
30. Kumar A, Niculescu-Mizil A, Kavukcuoglu K, Daume III H (2012) A binary classification framework for two-stage multiple kernel learning. arXiv preprint arXiv:1206.6428
31. Li BYS, Yeung LF, Ko KT (2015) Indefinite kernel ridge regression and its application on QSAR modelling. Neurocomputing 18:127–133
32. Liu C (2004) Gabor-based kernel pca with fractional power polynomial models for face recognition. IEEE Trans Pattern Anal Mach Intell 5:572–581
33. Liu F, Xue X (2016) Subgradient-based neural network for nonconvex optimization problems in support vector machines with indefinite kernels. J Ind Manag Optim 1:285–301
34. Liwicki S, Zafeiriou S, Tzimiropoulos G, Pantic M (2012) Efficient online subspace learning with an indefinite kernel for visual tracking and recognition. IEEE Trans Neural Netw Learn Syst 10:1624–1636
35. Loosli G, Canu S, Ong CS (2016) Learning SVM in Kreĭn spaces. IEEE Trans Pattern Anal Mach Intell 6:1204–1216
36. Luss R, d'Aspremont A (2008) Support vector machine classification with indefinite kernels. In: Proceedings of 22nd conference on Advances in neural information processing systems, pp 953–960
37. Melacci S, Belkin M (2011) Laplacian support vector machines trained in the primal. J Mach Learn Res 12(Mar):1149–1184
38. Ong CS, Mary X, Canu S, Smola AJ (2004) Learning with non-positive kernels. In: Proceedings of the twenty-first international conference on machine learning. ACM, p 81 (2004)
39. Ong CS, Smola AJ, Williamson RC (2005) Learning the kernel with hyperkernels. J Mach Learn Res 6(Jul):1043–1071
40. Pavlidis P, Weston J, Cai J, Grundy WN (2001) Gene functional classification from heterogeneous data. In: Proceedings of the fifth annual international conference on computational biology. ACM, pp 249–255 (2001)
41. Pekalska E, Haasdonk B (2009) Kernel discriminant analysis for positive definite and indefinite kernels. IEEE Trans Pattern Anal Mach Intell 6:1017–1032
42. Pekalska E, Harol A, Duin RP, Spillmann B, Bunke H (2006) Non-euclidean or non-metric measures can be informative. In: Joint IAPR international workshops on statistical techniques in pattern recognition (SPR) and structural and syntactic pattern recognition (SSPR). Springer, pp 871–880 (2006)
43. Pekalska E, Paclik P, Duin RP (2001) A generalized kernel approach to dissimilarity-based classification. J Mach Learn Res 2(Dec):175–211
44. Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y (2008) SimpleMKL. J Mach Learn Res 9(Nov):2491–2521
45. Rätsch G, Onoda T, Müller KR (2001) Soft margins for adaboost. Mach Learn 3:287–320
46. Roth V, Laub J, Buhmann JM, Müller KR (2003) Going metric: denoising pairwise data. Adv Neural Inf Process Syst 15:841–848
47. Roth V, Laub J, Kawanabe M, Buhmann JM (2003) Optimal cluster preserving embedding of nonmetric proximity data. IEEE Trans Pattern Anal Mach Intell 12:1540–1551
48. Ruszczyński AP (2006) Nonlinear optimization. Princeton University Press, Princeton
49. Schleif FM, Gisbrecht A, Tino P (2015) Large scale indefinite kernel fisher discriminant. In: International workshop on similarity-based pattern recognition. Springer, pp 160–170 (2015)
50. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge
51. Wang Z, Chen S, Xue H, Pan Z (2010) A novel regularization learning for single-view patterns: multi-view discriminative regularization. Neural Process Lett 3:159–175
52. Wright S, Nocedal J (1999) Numerical optimization. Springer Sci 35:67–68
53. Xu Z, Jin R, Yang H, King I, Lyu MR (2010) Simple and efficient multiple kernel learning by group lasso. In: Proceedings of the 27th international conference on machine learning, pp 1175–1182 (2010)
54. Xue H, Chen S (2014) Discriminality-driven regularization framework for indefinite kernel machine. Neurocomputing 133:209–221
55. Xue H, Chen S, Huang J (2012) Discriminative indefinite kernel classifier from pairwise constraints and unlabeled data. In: Proceedings of 21st international conference on pattern recognition. IEEE, pp 497–500 (2012)
56. Yan S, Xu X, Xu D, Lin S, Li X (2015) Image classification with densely sampled image windows and generalized adaptive multiple kernel learning. IEEE Trans Cybern 3:381–390

57. Ying Y, Campbell C, Girolami M (2009) Analysis of SVM with indefinite kernels. In: Proceedings of 23rd conference on Advances in neural information processing systems, pp 2205–2213 (2009)
58. Zheng D, Wang J, Zhao Y (2006) Non-flat function estimation with a multi-scale support vector regression. Neurocomputing 1:420–429
59. Zien A, Ong CS (2007) Multiclass multiple kernel learning. In: Proceedings of the 24th international conference on machine learning. ACM, pp 1191–1198

Affiliations

Hui Xue[1,2] · Lin Wang[1,2] · Songcan Chen[3] · Yunyun Wang[4]

✉ Hui Xue
hxue@seu.edu.cn

Lin Wang
wanglin@seu.edu.cn

Songcan Chen
s.chen@nuaa.edu.cn

Yunyun Wang
wangyunyun@njupt.edu.cn

1 School of Computer Science and Engineering, Southeast University, Nanjing 210096, People's Republic of China

2 Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing, People's Republic of China

3 School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, People's Republic of China

4 Department of Computer Science and Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210016, People's Republic of China