Indefinite Kernel Logistic Regression With Concave-Inexact-Convex Procedure

Fanghui Liu, Xiaolin Huang, *Senior Member, IEEE*, Chen Gong, *Member, IEEE*, Jie Yang, and Johan A. K. Suykens, *Fellow, IEEE*

***Abstract*— In kernel methods, the kernels are often required to be positive definitethat restricts the use of many indefinite kernels. To consider those nonpositive definite kernels, in this paper, we aim to build an indefinite kernel learning framework for kernel logistic regression (KLR). The proposed indefinite KLR (IKLR) model is analyzed in the reproducing kernel Kreĭn spaces and then becomes nonconvex. Using the positive decomposition of a nonpositive definite kernel, the derived IKLR model can be decomposed into the difference of two convex functions. Accordingly, a concave-convex procedure (CCCP) is introduced to solve the nonconvex optimization problem. Since the CCCP has to solve a subproblem in each iteration, we propose a concave-inexact-convex procedure (CCICP) algorithm with an inexact solving scheme to accelerate the solving process. Besides, we propose a stochastic variant of CCICP to efficiently obtain a proximal solution, which achieves the similar purpose with the inexact solving scheme in CCICP. The convergence analyses of the above-mentioned two variants of CCCP are conducted. By doing so, our method works effectively not only in a deterministic setting but also in a stochastic setting. Experimental results on several benchmarks suggest that the proposed IKLR model performs favorably against the standard (positive definite) KLR and other competitive indefinite learning-based algorithms.**

***Index Terms*— Concave-inexact-convex procedure (CCICP), indefinite kernel learning, kernel logistic regression (KLR), stochastic gradient descent (SGD).**

Manuscript received July 13, 2017; revised December 5, 2017, March 1, 2018, and June 7, 2018; accepted June 22, 2018. Date of publication July 26, 2018; date of current version February 19, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61572315, Grant 6151101179, Grant 61603248, and Grant 61602246, in part by the 973 Plan of China under Grant 2015CB856004, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20171430, in part by the Fundamental Research Funds for the Central Universities under Grant 30918011319, in part by the Open Project of the State Key Laboratory of Integrated Services Networks, Xidian University, under Grant ISN19-03, in part by the Summit of the Six Top Talents Program under Grant DZXX-027, in part by FWO under Grant G0A4917N, Grant G.0377.12, and Grant G.088114N, in part by IUAP P7/19 DYSCO, and in part by KU Leuven CoE PFV/10/002 (OPTEC). *(Corresponding authors: Jie Yang; Xiaolin Huang.)*

F. Liu, X. Huang, and J. Yang are with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: lfhsgre@sjtu.edu.cn; xiaolinhuang@sjtu.edu.cn; jieyang@sjtu.edu.cn).

C. Gong is with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information, Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: chen.gong@njust.edu.cn).

J. A. K. Suykens is with the Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, 3001 Leuven, Belgium (e-mail: johan.suykens@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2851305

I. Introduction

KERNEL methods [1], [2] have been successfully applied to many machine learning tasks such as classification [3], [4], regression [5], and clustering [6]. In these algorithms, a kernel function $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is employed to evaluate the similarity between two data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Herein, a positive definite (PD) kernel $\mathcal{K}$ results in a positive semidefinite (PSD) kernel matrix $\boldsymbol{K}$ to satisfy Mercer's condition. Consequently, the above-mentioned approaches with PD kernels can be theoretically analyzed in the reproducing kernel Hilbert spaces (RKHSs) [7].

Nevertheless, in real-world applications, we might meet some *indefinite* (real, symmetric, but not PD) [8] similarity measures due to the following reasons. First, one can comprehensively exploit the domain-specific structure in data and accordingly design a certain similarity measure. The kernel matrix derived by such measure often achieves promising empirical performance without any positive definiteness requirement on it. For example, one can utilize the Smith Waterman score [9] for the protein sequence, the optimal assignment kernels [10] for graph classification, or dynamic time warping [11], [12] for time series. In these cases, the corresponding kernel matrices generated by such similarities are not PSD. Second, the developed similarity measurements might be contaminated by outliers or noises [13] that make the initial PSD kernel matrix degenerate into an indefinite one. Third, the Mercer condition is difficult to verify even if a kernel is PD in essence. In this case, we have to tackle the kernel as an indefinite one. Finally, a PD kernel cannot be guaranteed to be PD embedded in another space. For instance, the Gaussian kernel is perhaps the most popular PD kernel with widespread applications. In a Riemannian manifold, the geodesic distance is more accurate than the Euclidean distance [14], [15], and accordingly, it seems natural to use the geodesic distance in the Gaussian kernel [16]. However, the kernel derived in this manner is not PD in general [17]. Based on the above-mentioned analyses, there are both algorithmic and theoretical requirements to consider these *indefinite* similarities. Here, we mainly discuss indefinite support vector machine (SVM) in above-mentioned two aspects.

In theory, a proper nonlinear feature mapping for indefinite kernels should be redefined because Mercer's theorem is no longer valid. Hence, Ong *et al.* [18] introduce the reproducing kernel Kreĭn spaces (RKKS) to give a characterization

in terms of primal and dual problems for SVM with indefinite kernels [19]. Compared with the conventional RKHS for PD kernels, the inner products might be negative for RKKS.

The solving algorithms for indefinite kernel learning can be grouped into two categories: spectrum modification and nonconvex optimization. In the first approach, the indefinite kernel matrix is transformed into a PSD one by spectrum modification, and then a regular solver can be used. For example, "Flip" [20] uses the absolute value of eigenvalues, "Clip" [21] sets the nonnegative eigenvalues to zero, and "Shift" [22] plus a positive constant to all eigenvalues until the smallest one is zero. However, the above-mentioned operations actually change the indefinite matrix itself, which results in the inconsistency between the training and test kernel. In comparison, some solvers can directly deal with the nonconvex dual problem within indefinite kernel matrices. In [23], SVM with indefinite kernels can still be solved by the SMO-type algorithm. Note that this algorithm still converges but the solution is just a stationary point. Besides, Akoa [24] and Xu *et al.* [25] directly introduce a nonconvex approach, the concave-convex procedure (CCCP) [26] to solve such problem effectively.

In this paper, we focus on kernel logistic regression (KLR) with indefinite kernels. KLR is a powerful and representative classifier and has been shown to be effective for classification tasks. However, KLR equipped with indefinite kernels has not yet been investigated in the past. Formally, the contributions of this paper are summarized as follows.

1) We build the indefinite KLR (IKLR) model in the RKKS and directly focus on its nonconvex primal form, the formulation of which keeps consistency to the standard KLR.
2) To accelerate the solving process, we propose a concave-inexact-convex procedure (CCICP) algorithm with an early termination technique to obtain a proximal solution during each iteration. We provide a convergence analysis of such approximation algorithm.
3) A stochastic variant of CCICP is proposed with convergence guarantees, which achieves the similar effect with the inexact solving scheme.

This paper is the extended version of our previous work [27]. Apart from details added in several sections, the main extension contains three parts: first, we incorporate stochastic gradient descent (SGD) into CCCP to solve the proposed IKLR model, and then, a stochastic variant of CCICP is presented to further accelerate the solving process. Second, the convergence analysis of such stochastic optimization algorithm is theoretically demonstrated. Third, we provide more experiments results on several benchmarks. And accordingly, more parameters' comparison analysis and computational complexity analysis are also provided.

The remainder of this paper is organized as follows. Section II briefly reviews KLR. Section III introduces the proposed IKLR model. The optimization algorithm of the IKLR model is presented in Section IV. Experimental results on several data sets are presented in Section V, and the conclusion is given in Section VI.

II. Review: Kernel Logistic Regression

In this section, we briefly review the regular KLR in the binary classification task. Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ with its label $y_i \in \{+1, -1\}$ be n training points, we concern the inference of a function $f\colon \mathcal{X} \to \mathcal{Y}$ that predicts a target $y \in \mathcal{Y}$ of a data point $\boldsymbol{x} \in \mathcal{X}$. KLR can be fit in the regularization framework of loss+penalty using the exponential loss function $\ell(f) = \ln(1 + e^{-yf})$. Thus, for a given positive regularization parameter λ, KLR is the minimum of the following regularized empirical risk functional:

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ln(1 + e^{-y_i f(\boldsymbol{x}_i)}) + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2 \tag{1}$$

with the RKHS $\mathcal{H}$ generated by the PD kernel $\mathcal{K}(\cdot, \cdot)$. Using the representer theorem [28] in RKHS, the optimal function f^* is

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

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_n]^\top$ is the coefficient vector. Combining this to (1), KLR is reformulated as

$$\min_{\boldsymbol{\alpha}} \frac{1}{n} \sum_{i=1}^{n} \ln \left(1 + \exp\left(-y_i \sum_{j=1}^{n} \alpha_j \boldsymbol{K}_{ij}\right)\right) + \frac{\lambda}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j \boldsymbol{K}_{ij} \tag{2}$$

with $\boldsymbol{K}_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$. After some straightforward algebraic manipulations, we obtain a compact form of KLR

$$\min_{\boldsymbol{\alpha}} \frac{1}{n} \mathbf{1}^\top \ln(\mathbf{1} + \exp(-\boldsymbol{y} \odot \boldsymbol{K}\boldsymbol{\alpha})) + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{K} \boldsymbol{\alpha} \tag{3}$$

where $\mathbf{1}$ is the all-one vector, $\boldsymbol{y}$ is the label vector, and $\odot$ denotes the Hadamard product. Traditionally, $\boldsymbol{K}$ in (3) is required to be PSD, and accordingly, the optimization problem is formulated as a convex, unconstrained quadratic programming.

III. Indefinite Kernel Logistic Regression Model

The functional space spanned by indefinite kernels belong to the RKKS [29] instead of RKHS. We first introduce Kreĭn spaces and then derive the IKLR model.

Definition 1 (Kreĭn Space [29]): An inner product space is a Kreĭn space $\mathcal{H}_\mathcal{K}$ if there exist two Hilbert spaces $\mathcal{H}_+$ and $\mathcal{H}_-$ such that: 1) all $f \in \mathcal{H}_\mathcal{K}$ can be decomposed into $f = f_+ + f_-$, where $f_+ \in \mathcal{H}_+$ and $f_- \in \mathcal{H}_-$, respectively, and 2) $\forall f, g \in \mathcal{H}_\mathcal{K}$, $\langle f, g\rangle_{\mathcal{H}_\mathcal{K}} = \langle f_+, g_+\rangle_{\mathcal{H}_+} - \langle f_-, g_-\rangle_{\mathcal{H}_-}$.

If $\mathcal{H}_+$ and $\mathcal{H}_-$ are RKHSs, $\mathcal{H}_\mathcal{K}$ is an RKKS with a unique indefinite kernel $\mathcal{K}$ such that the reproducing property holds: for all $f \in \mathcal{H}_\mathcal{K}$, $f(\boldsymbol{x}) = \langle f, k(\boldsymbol{x}, \cdot)\rangle_{\mathcal{H}_\mathcal{K}}$. In this space, the squared norm and the squared distance[1] induced by an indefinite kernel $\mathcal{K}$ can be negative in contrast to the Euclidean case. This definition may not define a metric, as it violates the triangle inequality. However, this squared distance function is able to provide a justification of data representation in this

[1]A corresponding squared distance is defined as $d^2(x, x') = \mathcal{K}(x, x) - 2\mathcal{K}(x, x') + \mathcal{K}(x', x')$.

vector space. Details about the interpretation of SVM with indefinite kernels in the feature space can be found in [30].

Based on above-mentioned analyses, our IKLR model with an indefinite kernel $\mathcal{K}$ is formulated as

$$\min_{f\in\mathcal{H}_{\mathcal{K}}} \frac{1}{n}\sum_{i=1}^{n}\ln(1+e^{-y_i f(\boldsymbol{x}_i)})+\frac{\lambda}{2}\|f\|^2_{\mathcal{H}_{\mathcal{K}}} \tag{4}$$

with the RKKS $\mathcal{H}_{\mathcal{K}}$ generated by the indefinite kernel $\mathcal{K}(\cdot,\cdot)$. By the representer theorem in RKKS [18], the optimal f^* admits

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

Accordingly, (4) can be rewritten as

$$\min_{\boldsymbol{\alpha}} \frac{1}{n}\mathbf{1}^\top\ln(\mathbf{1}+\exp(-\boldsymbol{y}\odot\boldsymbol{K}\boldsymbol{\alpha}))+\frac{\lambda}{2}\boldsymbol{\alpha}^\top\boldsymbol{K}\boldsymbol{\alpha}. \tag{5}$$

One can see that the proposed IKLR model in (5) is similar to the regular KLR in (3), but it must be analyzed in RKKS and becomes nonconvex because of the indefinite kernel matrix.

To solve such nonconvex problem, we also need the following proposition.

Proposition 1 [18]: A nonpositive definite kernel $\mathcal{K}$ in RKKS admits a positive decomposition on a given set

$$\mathcal{K}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\mathcal{K}_+(\boldsymbol{x}_i,\boldsymbol{x}_j)-\mathcal{K}_-(\boldsymbol{x}_i,\boldsymbol{x}_j)\quad\forall\boldsymbol{x}_i,\boldsymbol{x}_j\in\mathcal{X}$$

with two PD kernels $\mathcal{K}_+$ and $\mathcal{K}_-$.

Hence, the proposed IKLR model in (5) can be further expressed as

$$\min_{\boldsymbol{\alpha}} \frac{1}{n}\mathbf{1}^\top\ln(\mathbf{1}+\exp(-\boldsymbol{y}\odot\boldsymbol{K}\boldsymbol{\alpha}))+\frac{\lambda}{2}\boldsymbol{\alpha}^\top(\boldsymbol{K}_+-\boldsymbol{K}_-)\boldsymbol{\alpha} \tag{6}$$

with two PSD kernel matrices $\boldsymbol{K}_+$ and $\boldsymbol{K}_-$, which can be obtained by eigenvalue decomposition of $\boldsymbol{K}$. To be specific, $\boldsymbol{K}=\boldsymbol{V}^\top\boldsymbol{\Lambda}\boldsymbol{V}$, where $\boldsymbol{V}$ is an orthogonal matrix and the diagonal matrix is $\boldsymbol{\Lambda}=\mathrm{diag}(\mu_1,\mu_2,\ldots,\mu_n)$ with eigenvalues $\mu_1\geq\mu_2\geq\cdots\geq\mu_n$. Without loss of generality, suppose that the first s eigenvalues are nonnegative and the remaining $n-s$ ones are negative, $\boldsymbol{K}_+$ and $\boldsymbol{K}_-$ can thus be given as

$$\begin{cases}\boldsymbol{K}_+=\boldsymbol{V}^\top\mathrm{diag}(\mu_1+\tau,\ldots,\mu_v+\tau,\tau,\ldots,\tau)\boldsymbol{V}\\ \boldsymbol{K}_-=\boldsymbol{V}^\top\mathrm{diag}(\tau,\ldots,\tau,\rho-\mu_{v+1},\ldots,\tau-\mu_n)\boldsymbol{V}\end{cases}$$

where τ is chosen by $\tau>-\mu_n$ to ensure that these two matrices $\boldsymbol{K}_+$ and $\boldsymbol{K}_-$ are PSD. After conducting this positive decomposition, we decompose the objective function in (6) as difference of two convex functions $g(\boldsymbol{\alpha})$ and $h(\boldsymbol{\alpha})$

$$\begin{cases}g(\boldsymbol{\alpha})=\frac{1}{n}\mathbf{1}^\top\ln(\mathbf{1}+\exp(-\boldsymbol{y}\odot\boldsymbol{K}\boldsymbol{\alpha}))+\frac{\lambda}{2}\boldsymbol{\alpha}^\top\boldsymbol{K}_+\boldsymbol{\alpha}\\ h(\boldsymbol{\alpha})=\frac{\lambda}{2}\boldsymbol{\alpha}^\top\boldsymbol{K}_-\boldsymbol{\alpha}.\end{cases} \tag{7}$$

IV. CCICP Optimization for IKLR

This section first introduces a CCICP algorithm with two approximation schemes to solve the nonconvex optimization problem, and then provides convergence analyses of the proposed CCICP algorithm.

A. CCICP in the IKLR Model

The CCCP [26] is a typical nonconvex algorithm to solve d.c. (difference of convex functions) programs. By decomposing the nonconvex objective function in (6) into the difference of two convex functions $g(\boldsymbol{\alpha})$ and $h(\boldsymbol{\alpha})$, the CCCP is an iterative procedure with

$$\boldsymbol{\alpha}_{k+1}\in\operatorname*{argmin}_{\boldsymbol{\alpha}}\ g(\boldsymbol{\alpha})-\boldsymbol{\alpha}^\top\nabla h(\boldsymbol{\alpha}_k).$$

The core idea of CCCP is to linearize the concave part of the nonconvex objective function, i.e., $-h(\boldsymbol{\alpha})$, around its current solution $\boldsymbol{\alpha}_k$. At each iteration, its convex approximation is formulated as

$$\mathcal{F}_k(\boldsymbol{\alpha})\triangleq\mathcal{F}(\boldsymbol{\alpha},\boldsymbol{\alpha}_k)=g(\boldsymbol{\alpha})-[h(\boldsymbol{\alpha}_k)+\nabla h^\top(\boldsymbol{\alpha}_k)(\boldsymbol{\alpha}-\boldsymbol{\alpha}_k)] \tag{8}$$

where $\mathcal{F}_k(\boldsymbol{\alpha})$ can be solved by an off-the-shelf convex algorithm such as the GD method to obtain $\boldsymbol{\alpha}_{k+1}$. To be specific, in our model, $h(\boldsymbol{\alpha})$ is replaced by its first-order Taylor approximation at $\boldsymbol{\alpha}_k$

$$\tilde{h}(\boldsymbol{\alpha}_k)=h(\boldsymbol{\alpha}_k)+\lambda\boldsymbol{\alpha}_k^\top\boldsymbol{K}_-(\boldsymbol{\alpha}-\boldsymbol{\alpha}_k).$$

Combining this to (8), we have

$$\mathcal{F}_k(\boldsymbol{\alpha})=\frac{\lambda}{2}\boldsymbol{\alpha}^\top\boldsymbol{K}_+\boldsymbol{\alpha}+\frac{1}{n}\mathbf{1}^\top\ln(\mathbf{1}+\exp(-\boldsymbol{y}\odot\boldsymbol{K}\boldsymbol{\alpha}))-\tilde{h}(\boldsymbol{\alpha}_k). \tag{9}$$

Nonetheless, one can see that at each iteration, the CCCP needs to solve the subproblem, which makes CCCP inefficient especially for a large-scale problem. Based on this, we attempt to obtain an inexact solution of the subproblem to speed up the solving process, termed as the CCICP. To this end, we develop two approaches, one is the GD method with an early termination condition, termed as "CCICP-GD," and the other is incorporated with SGD to achieve the similar effect, termed as "CCICP-SGD."

1) Solving With CCICP-GD: In our CCICP-GD method, the subproblem is solved by the GD method, in which the gradient $\nabla_{\boldsymbol{\alpha}}\mathcal{F}_k(\boldsymbol{\alpha})$ is

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_k(\boldsymbol{\alpha})=\lambda\boldsymbol{K}_+\boldsymbol{\alpha}-\frac{1}{n}\boldsymbol{y}\odot\boldsymbol{K}\boldsymbol{\beta}-\lambda\boldsymbol{K}_-\boldsymbol{\alpha}_k \tag{10}$$

where $\boldsymbol{\beta}=(\beta_1,\beta_2,\ldots,\beta_n)^\top$ is defined by

$$\beta_i=\frac{1}{1+\exp\left(y_i\sum_{j=1}^{n}\beta_j\boldsymbol{K}_{ij}\right)}\quad\forall i=1,2,\ldots,n. \tag{11}$$

Using an early termination condition, the inexact solution $\boldsymbol{\alpha}_{k+1}\triangleq\boldsymbol{\alpha}_k^{(T)}$ after T iterations is obtained by $\boldsymbol{\alpha}_{k+1}\approx\operatorname{argmin}_{\boldsymbol{\alpha}}\mathcal{F}_k(\boldsymbol{\alpha})$. In Section IV-B, we detail the definition of such early termination condition and then theoretically demonstrate the convergence analyses of such approximation.

2) Solving With CCICP-SGD: The SGD method can also achieve the similar effect with such inexact scheme to solve the subproblem. Since it only processes a minibatch of data points [31] or even one data point [32] in each iteration, the computational cost per iteration dramatically decreases. To be specific, in our algorithm, we randomly pick up only

one data point to compute the gradient in the subproblem. The updating scheme is given as

$$\boldsymbol{\alpha}^{(t+1)} \leftarrow \boldsymbol{\alpha}^{(t)} - \eta_t(-y_j \boldsymbol{K}_j \beta_j + \lambda \boldsymbol{K}_+ \boldsymbol{\alpha}^{(t)}) \tag{12}$$

where η_t is the step size in the tth iteration, and $\boldsymbol{K}_j$ represents the (randomly picked) jth column of the kernel matrix $\boldsymbol{K}$. Finally, the detailed procedure of the CCICP algorithm with GD and SGD for IKLR is summarized in Algorithm 1.

Algorithm 1 CCICP for the IKLR Model

Input: an indefinite kernel matrix $\boldsymbol{K}$ and its positive decomposition $\boldsymbol{K}_+$ and $\boldsymbol{K}_-$.
Output: the coefficient vector $\boldsymbol{\alpha}$.

Set: stopping criterion: the inexact parameter $\epsilon = 1$, $k_{\max} = 20$, the learning rate $\eta = 0.02$, and $\rho = 0.8$;
Initialize $k = 0$ and $\boldsymbol{\alpha}_0$;
The parameter λ is chosen by cross-validation;
Repeat
  Obtain $\tilde{h}(\boldsymbol{\alpha}_k) = \lambda \boldsymbol{K}_- \boldsymbol{\alpha}_k$ and the subproblem $\mathcal{F}_k(\boldsymbol{\alpha})$ by Eq. (9);
  `// Solve the subproblem.`
  Initialize $t = 0$ and compute $\mathcal{F}_k(\boldsymbol{\alpha}_k^{(0)})$;
  while $\|\mathcal{F}_k(\boldsymbol{\alpha}_k^{(t+1)}) - \mathcal{F}_k(\boldsymbol{\alpha}_k^{(t)})\| > \epsilon$ **do**
    Obtain the gradient $\nabla\mathcal{F}_k(\boldsymbol{\alpha}_k^{(t)})$ by Eq. (10);
    GD: $\boldsymbol{\alpha}_k^{(t+1)} := \boldsymbol{\alpha}_k^{(t)} - \eta_t \nabla\mathcal{F}_k(\boldsymbol{\alpha}_k^{(t)})$;
    SGD: Randomly pick a $j \in \{1, 2, \cdots, n\}$ and then update $\boldsymbol{\alpha}_k^{(t+1)}$ by Eq. (12);
    $\eta := \rho\eta$;
    $t := t + 1$;
  end
  Output the inexact solution $\boldsymbol{\alpha}_{k+1} := \boldsymbol{\alpha}_k^{(t)}$ of Eq. (9);
  `// Complete the inner loop.`
  $k := k + 1$;
Until $k = k_{\max}$;
Output the stationary point $\boldsymbol{\alpha}_{k_{\max}}$ of (7).

One can see that Algorithm 1 contains two loops. In each iteration of the outer loop, an unconstrained quadratic programming is solved by GD and its complexity is $\mathcal{O}(dn)$, where d is the feature dimension and n is the number of training examples. Finally, the total computational complexity of our CCICP algorithm is $\mathcal{O}(Tkdn)$, where T is the number of convergence iterations and k is the number of classes.

When we obtain $\boldsymbol{\alpha}^*$ by Algorithm 1, a test data point $\boldsymbol{z}$ can be predicted by

$$p(\boldsymbol{z}) = \frac{\exp(\boldsymbol{K}_{\boldsymbol{z}}\boldsymbol{\alpha}^*)}{1 + \exp(\boldsymbol{K}_{\boldsymbol{z}}\boldsymbol{\alpha}^*)}$$

with $\boldsymbol{K}_{\boldsymbol{z}} = [\mathcal{K}(\boldsymbol{x}_1, \boldsymbol{z}), \mathcal{K}(\boldsymbol{x}_2, \boldsymbol{z}), \ldots \mathcal{K}(\boldsymbol{x}_n, \boldsymbol{z})]$. If $p(\boldsymbol{z}) \geq 0.5$, its label is predicted by $+1$, and -1 otherwise.

B. Analysis of CCICP-GD

This section investigates the convergence of the proposed CCICP-GD. Since the GD algorithm is used to solve the subproblem, it satisfies $\mathcal{F}_k(\boldsymbol{\alpha}_{k+1}) \triangleq \mathcal{F}_k(\boldsymbol{\alpha}_k^{(T)}) \leq \mathcal{F}_k(\boldsymbol{\alpha}_k^*)$. Herein, the inexact solution $\boldsymbol{\alpha}_k^{(T)}$ satisfies

$$\boldsymbol{\alpha}_k^{(T)} \in \mathrm{U}_{\delta(\boldsymbol{\alpha})}(\boldsymbol{\alpha}_k^*) \triangleq \left\{\boldsymbol{\alpha} \mid \left\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_k^*\right\| \leq \delta(\boldsymbol{\alpha})\right\}$$

where $\boldsymbol{\alpha}_k^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \mathcal{F}_k(\boldsymbol{\alpha})$ is the optimal result. The notation $\delta(\boldsymbol{\alpha})$ depends on the current solution, and it should be bounded to guarantee the convergence of CCICP. In this case, such approximation solution $\boldsymbol{\alpha}_k^{(T)}$ does not satisfy the Karush–Kuhn–Tucker (KKT) condition. Suppose that

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_k(\boldsymbol{\alpha})|_{\boldsymbol{\alpha}=\boldsymbol{\alpha}_k^{(T)}} = \epsilon\|\boldsymbol{\alpha}_k\| \neq 0 \tag{13}$$

where ϵ depends on $\delta(\boldsymbol{\alpha})$, and its choice will be analyzed to guarantee the convergence of CCICP-GD in the following description.

The main result for CCICP-GD is demonstrated by Theorem 1, that is, when ϵ is upper bounded, the sequence $\{\boldsymbol{\alpha}_k\}_{k=0}^{\infty}$ generated by CCICP with an initial point $\boldsymbol{\alpha}_0 \in \mathbb{R}^n$ still converges. Before we proceed with the proof of Theorem 1, we need Lemma 1.

Lemma 1: Given a sigmoid function $R(x) = (1 + \mathrm{e}^{cx})^{-1}$ with $c \in \{+1, -1\}$ on $\mathbb{R}$, for any $x_1 < x_2$, we have

$$|R(x_1) - R(x_2)| \leq \frac{1}{4}|x_1 - x_2|. \tag{14}$$

Proof: Since $R(x)$ is a differentiable function, by the Lagrange mean value theorem, there exists at least one point $\xi \in (x_1, x_2)$ such that

$$|R(x_1) - R(x_2)| = |(x_1 - x_2)R'(\xi)|$$

where $|R'(\xi)|$ admits

$$|R'(\xi)| = \frac{\mathrm{e}^{c\xi}}{(1 + \mathrm{e}^{c\xi})^2} = \frac{1}{\mathrm{e}^{c\xi} + \mathrm{e}^{-c\xi} + 2} \leq \frac{1}{4}$$

which concludes the proof. □

We now present the convergence theorem for CCICP-GD.

Theorem 1: Let $\{\boldsymbol{\alpha}_k\}_{k=0}^{\infty}$ be any sequence generated by CCICP-GD, its limit point is a stationary point if ϵ in (13) satisfies

$$\epsilon < \lambda(\|\boldsymbol{K}_+\| - \|\boldsymbol{K}_-\|) - \frac{\|\boldsymbol{K}\|^2}{4n}. \tag{15}$$

Proof: Let $\phi : U \subset \mathbb{R}^n \to \mathbb{R}^n$ be a point-to-set map such that

$$\phi(\boldsymbol{\alpha}_k) = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \ \mathcal{F}_k(\boldsymbol{\alpha})$$

which generates an inexact sequence $\{\boldsymbol{\alpha}_k\}_{k=0}^{\infty}$. Besides, $\phi(\boldsymbol{\alpha}_k)$ satisfies

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}_k(\boldsymbol{\alpha})|_{\boldsymbol{\alpha}=\phi(\boldsymbol{\alpha}_k)} = \epsilon\|\boldsymbol{\alpha}_k\|.$$

In the following, we aim to prove that the map ϕ is a nonexpansive mapping for two arbitrary points $\boldsymbol{p}, \boldsymbol{q} \in int(U)$ such that

$$\|\phi(\boldsymbol{p}) - \phi(\boldsymbol{q})\| \leq \kappa\|\boldsymbol{p} - \boldsymbol{q}\|$$

where the nonexpansive coefficient is $\kappa \in [0, 1)$. Suppose that $\phi(\boldsymbol{p})$ and $\phi(\boldsymbol{q})$ satisfy

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}(\boldsymbol{\alpha}, \boldsymbol{p})|_{\boldsymbol{\alpha}=\phi(\boldsymbol{p})} = \epsilon_1\|\boldsymbol{p}\| \tag{16}$$

$$\nabla_{\boldsymbol{\alpha}}\mathcal{F}(\boldsymbol{\alpha}, \boldsymbol{q})|_{\boldsymbol{\alpha}=\phi(\boldsymbol{q})} = \epsilon_2\|\boldsymbol{q}\| \tag{17}$$

where ϵ_1 and ϵ_2 correspond to the bounded error. For simplicity, suppose $\epsilon_1 \leq \epsilon_2$, so the difference between (16) and (17) is given as

$$\lambda \boldsymbol{K}_+[\phi(\boldsymbol{p}) - \phi(\boldsymbol{q})] = \lambda \boldsymbol{K}_-(\boldsymbol{p} - \boldsymbol{q}) + \frac{1}{n}\boldsymbol{y} \odot \boldsymbol{K}\boldsymbol{h} + \epsilon_1\|\boldsymbol{p}\| - \epsilon_2\|\boldsymbol{q}\| \quad (18)$$

where $\boldsymbol{h} = [h_1, h_2, \ldots, h_n]^\top$ is defined by

$$h_i = \frac{1}{1 + \exp(y_i \boldsymbol{K}^{(i)}\phi(\boldsymbol{p}))} - \frac{1}{1 + \exp(y_i \boldsymbol{K}^{(i)}\phi(\boldsymbol{q}))}.$$

Using Lemma 1, we have

$$|h_i| \leq \frac{1}{4}|\boldsymbol{K}^{(i)}\phi(\boldsymbol{p}) - \boldsymbol{K}^{(i)}\phi(\boldsymbol{q})| \quad \forall i = 1, 2, \ldots, n$$

and accordingly $\|\boldsymbol{h}\|_\infty$ satisfies[2]

$$\|h\|_\infty \leq \frac{|\boldsymbol{K}^{(s)}\phi(\boldsymbol{p}) - \boldsymbol{K}^{(s)}\phi(\boldsymbol{q})|}{4} \leq \frac{\|\boldsymbol{K}^{(s)}\|_1 \cdot \|\phi(\boldsymbol{p}) - \phi(\boldsymbol{q})\|_\infty}{4} \leq \frac{1}{4}\|\boldsymbol{K}\|_\infty \|\phi(\boldsymbol{p}) - \phi(\boldsymbol{q})\|_\infty$$

with $s = \underset{i}{\operatorname{argmin}} |\boldsymbol{K}^{(i)}\phi(\boldsymbol{p}) - \boldsymbol{K}^{(i)}\phi(\boldsymbol{q})|, i = 1, 2, \ldots, n$. Since $\boldsymbol{K}_+$ is PD, (18) can be rewritten as

$$\phi(\boldsymbol{p}) - \phi(\boldsymbol{q}) = \frac{1}{\lambda}\boldsymbol{K}_+^{-1}\left\{\lambda \boldsymbol{K}_-(\boldsymbol{p} - \boldsymbol{q}) + \frac{1}{n}\boldsymbol{y} \odot \boldsymbol{K}\boldsymbol{h} + \epsilon_1\|\boldsymbol{p}\| - \epsilon_2\|\boldsymbol{q}\|\right\}.$$

Accordingly, $\|\phi(\boldsymbol{a}) - \phi(\boldsymbol{b})\|$ can be bounded by (19), as shown at the bottom of this page, which leads to

$$\|\phi(\boldsymbol{p}) - \phi(\boldsymbol{q})\| \leq \frac{\|\boldsymbol{K}_-\| + \frac{\epsilon_2}{\lambda}}{\|\boldsymbol{K}_+\| - \frac{1}{4\lambda n}\|\boldsymbol{K}\|^2}\|\boldsymbol{p} - \boldsymbol{q}\|. \quad (20)$$

Likewise, if $\epsilon_2 < \epsilon_1$, we have

$$\|\phi(\boldsymbol{q}) - \phi(\boldsymbol{p})\| \leq \frac{\|\boldsymbol{K}_-\| + \frac{\epsilon_1}{\lambda}}{\|\boldsymbol{K}_+\| - \frac{1}{4\lambda n}\|\boldsymbol{K}\|^2}\|\boldsymbol{q} - \boldsymbol{p}\|. \quad (21)$$

Accordingly, (20) and (21) can be reformulated as

$$\|\phi(\boldsymbol{p}) - \phi(\boldsymbol{q})\| \leq \frac{\|\boldsymbol{K}_-\| + \frac{\max\{\epsilon_1, \epsilon_2\}}{\lambda}}{\|\boldsymbol{K}_+\| - \frac{1}{4\lambda n}\|\boldsymbol{K}\|^2}\|\boldsymbol{p} - \boldsymbol{q}\|$$

where we choose $\epsilon = \max\{\epsilon_1, \epsilon_2\}$. Thereby, the map ϕ is a nonexpansive mapping with the following condition:

$$\kappa \triangleq \frac{\|\boldsymbol{K}_-\| + \frac{\epsilon}{\lambda}}{\|\boldsymbol{K}_+\| - \frac{1}{4\lambda n}\|\boldsymbol{K}\|^2} < 1$$

which derives the upper bound of ϵ presented in (15) by some straightforward algebraic manipulations. Hence, by the fixed point theorem [33], the map ϕ is theoretically demonstrated to be a nonexpensive mapping if ϵ is upper bounded. □

Theorem 1 demonstrates that the CCICP with early termination condition is theoretically guaranteed to converge if the inexact parameter ϵ is upper bounded. In our IKLR model, such convergence condition is easily satisfied, and thus the inexact parameter can be set to a relatively large one in practice. Note that the early termination condition in Algorithm 1 is given by variations of the subproblem function value $\mathcal{F}(\boldsymbol{\alpha})$ between the two consecutive iterations instead of the gradient variations such as $\|\nabla\mathcal{F}_k(\boldsymbol{\alpha}_k^{(t+1)}) - \nabla\mathcal{F}_k(\boldsymbol{\alpha}_k^{(t)})\| < \epsilon$. This is because it is relatively easier to compute the subproblem function value than the gradient computation, especially when the SGD method is considered.

C. Analysis With CCICP-SGD

Apart from an early stop scheme in GD to obtain an inexact solution of the subproblem, effective stochastic gradient-based methods [34], [35] can also be used as underlying solvers to accelerate the solving process that achieves the similar approximation in expectation. Specifically, since the subproblem becomes strongly convex, the combination of CCCP and SGD in our model achieves fast convergence and theoretical guarantees when compared to directly using SGD to solve the initial nonconvex problem.

Before we prove that a sequence $\{\boldsymbol{\alpha}_k\}_{k=0}^\infty$ generated by CCICP-SGD converges to a stationary point, we need a few additional results. We denote the nonconvex objective function by $F(\boldsymbol{\alpha}) = g(\boldsymbol{\alpha}) - h(\boldsymbol{\alpha})$. The idea of the following lemma is to show that the objective function is monotonic decent in probability.

Lemma 2: The sequence $\{\boldsymbol{\alpha}_k\}_{k=0}^\infty$ generated by CCICP-SGD satisfies the monotonic decent property in probability

$$\mathbb{E}[F(\boldsymbol{\alpha}_{k+1})] \leq \mathbb{E}[F(\boldsymbol{\alpha}_k)]. \quad (22)$$

Proof: Due to the convexity of $g(\boldsymbol{\alpha})$ and $h(\boldsymbol{\alpha})$, we denote $\hbar(\boldsymbol{\alpha}) = -h(\boldsymbol{\alpha})$ as a concave function, and thus $F(\boldsymbol{\alpha}) = g(\boldsymbol{\alpha}) + \hbar(\boldsymbol{\alpha})$. In the kth iteration, SGD is used to solve the subproblem $\mathcal{F}_k(\boldsymbol{\alpha})$ in (8), yielding $\boldsymbol{\alpha}_{k+1}$ to satisfy the KKT condition in expectation, namely $\mathbb{E}[\nabla\mathcal{F}_k(\boldsymbol{\alpha}_{k+1})] = 0$. Accordingly, we have the following equation:

$$\mathbb{E}[\nabla g(\boldsymbol{\alpha}_{k+1})] = \mathbb{E}[\nabla h(\boldsymbol{\alpha}_k)] = -\mathbb{E}[\nabla\hbar(\boldsymbol{\alpha}_k)].$$

For $\boldsymbol{\alpha}_k$ and $\boldsymbol{\alpha}_{k+1}$, it satisfies

$$\begin{cases} g(\boldsymbol{\alpha}_k) \geq g(\boldsymbol{\alpha}_{k+1}) + (\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})\nabla g(\boldsymbol{\alpha}_{k+1}) \\ \hbar(\boldsymbol{\alpha}_k) \geq \hbar(\boldsymbol{\alpha}_{k+1}) + (\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})\nabla\hbar(\boldsymbol{\alpha}_k). \end{cases} \quad (23)$$

[2]Here we use $|\boldsymbol{p}^\top\boldsymbol{q}| = \|\boldsymbol{p}\|_a\|\boldsymbol{q}\|_b$, where $(1/a) + (1/b) = 1$.

$$\begin{aligned} \|\phi(\boldsymbol{p}) - \phi(\boldsymbol{q})\| &\leq \frac{1}{\lambda}\boldsymbol{K}_+^{-1}\left\{\lambda \boldsymbol{K}_-(\boldsymbol{p} - \boldsymbol{q}) + \frac{1}{n}\boldsymbol{y} \odot \boldsymbol{K}\boldsymbol{h} + \epsilon_1\|\boldsymbol{p}\| - \epsilon_2\|\boldsymbol{q}\|\right\} \\ &\leq \left\|\boldsymbol{K}_+^{-1}\boldsymbol{K}_-\right\|\|\boldsymbol{p} - \boldsymbol{q}\| + \frac{\left\|\boldsymbol{K}_+^{-1}\boldsymbol{K}\right\|}{\lambda n}\|\boldsymbol{h}\| + \frac{\epsilon_2}{\lambda}\left\|\boldsymbol{K}_+^{-1}\right\|\|\boldsymbol{p}\| - \|\boldsymbol{q}\|| \\ &\leq \left\|\boldsymbol{K}_+^{-1}\right\|\|\boldsymbol{K}_-\|\|\boldsymbol{p} - \boldsymbol{q}\| + \frac{\left\|\boldsymbol{K}_+^{-1}\boldsymbol{K}\right\|\|\boldsymbol{K}\|}{4\lambda n}\|\phi(\boldsymbol{p}) - \phi(\boldsymbol{q})\| + \frac{\epsilon_2}{\lambda}\left\|\boldsymbol{K}_+^{-1}\right\|\|\boldsymbol{p} - \boldsymbol{q}\| \end{aligned} \quad (19)$$

Then, we take the expectation of (23) and use the expectation independence

$$\begin{cases}\mathbb{E}[g(\boldsymbol{\alpha}_k)] \geq \mathbb{E}[g(\boldsymbol{\alpha}_{k+1})] + \mathbb{E}[(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})]\mathbb{E}[\nabla g(\boldsymbol{\alpha}_{k+1})] \\ \mathbb{E}[\hbar(\boldsymbol{\alpha}_k)] \geq \mathbb{E}[\hbar(\boldsymbol{\alpha}_{k+1})] + \mathbb{E}[(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})]\mathbb{E}[\nabla \hbar(\boldsymbol{\alpha}_k)].\end{cases}$$

Combining the above-mentioned two subequations, we have

$$\mathbb{E}[g(\boldsymbol{\alpha}_k) + \hbar(\boldsymbol{\alpha}_k)] \geq \mathbb{E}[g(\boldsymbol{\alpha}_{k+1}) + \hbar(\boldsymbol{\alpha}_{k+1})] \tag{24}$$

which completes the proof. □

Lemma 2 demonstrates the monotonic decent property in probability. However, such analysis is not complete, as the monotone descent property by itself is not sufficient to claim the convergence of $\{\boldsymbol{\alpha}_k\}_{k=0}^{\infty}$. The similar situation in the initial CCCP version has been discussed in [36].

In Section V, we first give the definition of Lipschitz smoothness required by the subsequent analyses, and then investigate the convergence of $\{\boldsymbol{\alpha}_k\}_{k=0}^{\infty}$.

Definition 2: A function F is gradient Lipschitz smooth if there exists $L_F > 0$ such that

$$\|\nabla F(\boldsymbol{\alpha}) - \nabla F(\boldsymbol{\alpha}')\|_2 \leq L_F \|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|_2 \quad \forall \boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \text{dom } F.$$

Furthermore, if F is also a convex function over a (closed) convex domain, it satisfies

$$F(\boldsymbol{\alpha}) - F(\boldsymbol{\alpha}') \leq \nabla F^{\top}(\boldsymbol{\alpha}')(\boldsymbol{\alpha} - \boldsymbol{\alpha}') + \frac{L_F}{2}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|_2^2.$$

Suppose that $\mathcal{F}_k$ is the $L_{\mathcal{F}}$-smooth function, we have

$$\mathcal{F}_k(\boldsymbol{\alpha}) - \mathcal{F}_k(\boldsymbol{\alpha}_k) \leq \nabla \mathcal{F}_k^{\top}(\boldsymbol{\alpha}_k)(\boldsymbol{\alpha} - \boldsymbol{x}_k) + \frac{L_{\mathcal{F}}}{2}\|\boldsymbol{\alpha} - \boldsymbol{\alpha}_k\|_2^2.$$

To obtain $\boldsymbol{\alpha}_{k+1}$, we use SGD to solve $\mathcal{F}_k(\boldsymbol{\alpha})$ with T iterations, i.e., $\boldsymbol{\alpha}_{k+1} \triangleq \boldsymbol{\alpha}_k^{(T)}$. Specifically, the first iteration is defined as $\boldsymbol{\alpha}_k^{(1)} = \boldsymbol{\alpha}_k - \eta\widehat{\boldsymbol{g}}$, where the stepsize η is set to $(1/L_{\mathcal{F}})$ and the produced vector $\widehat{\boldsymbol{g}}$ satisfies $\mathbb{E}[\widehat{\boldsymbol{g}}] = \nabla\mathcal{F}_k(\boldsymbol{\alpha}_k)$. Therefore, we have

$$\begin{aligned}&\mathcal{F}_k\big(\boldsymbol{\alpha}_k^{(1)}\big) - \mathcal{F}_k(\boldsymbol{\alpha}_k) \\ &\quad\leq \nabla\mathcal{F}_k^{\top}(\boldsymbol{\alpha}_k)\big(\boldsymbol{\alpha}_k^{(1)} - \boldsymbol{\alpha}_k\big) + \frac{L_{\mathcal{F}}}{2}\big\|\boldsymbol{\alpha}_k^{(1)} - \boldsymbol{\alpha}_k\big\|_2^2 \\ &\quad= -\frac{1}{L_{\mathcal{F}}}\nabla\mathcal{F}_k^{\top}(\boldsymbol{\alpha}_k)\widehat{\boldsymbol{g}} + \frac{1}{2L_{\mathcal{F}}}\|\widehat{\boldsymbol{g}}\|_2^2.\end{aligned} \tag{25}$$

We take the expectation

$$\begin{aligned}&\mathbb{E}\big[\mathcal{F}_k\big(\boldsymbol{\alpha}_k^{(1)}\big)\big] \\ &\quad\leq \mathbb{E}[\mathcal{F}_k(\boldsymbol{\alpha}_k)] - \frac{1}{L_{\mathcal{F}}}\mathbb{E}\big[\nabla\mathcal{F}_k^{\top}(\boldsymbol{\alpha}_k)\widehat{\boldsymbol{g}}\big] + \frac{1}{2L_{\mathcal{F}}}\mathbb{E}\big[\|\widehat{\boldsymbol{g}}\|_2^2\big]\end{aligned} \tag{26}$$

where the formula satisfies $\mathbb{E}[\nabla\mathcal{F}_k^{\top}(\boldsymbol{\alpha}_k)\widehat{\boldsymbol{g}}] = \mathbb{E}[\nabla\mathcal{F}_k^{\top}(\boldsymbol{\alpha}_k)]\mathbb{E}[\widehat{\boldsymbol{g}}]$ because they are independent of each other, and $\mathbb{E}[\mathcal{F}_k(\boldsymbol{\alpha}_{k+1})] \triangleq \mathbb{E}[\mathcal{F}_k(\boldsymbol{\alpha}_k^{(T)})] \leq \mathbb{E}[\mathcal{F}_k(\boldsymbol{\alpha}_k^{(1)})]$ by Lemma 2. By combining (25) and (26), we obtain

$$\mathbb{E}[\mathcal{F}_k(\boldsymbol{\alpha}_{k+1})] \leq \mathbb{E}[\mathcal{F}_k(\boldsymbol{\alpha}_k)] - \frac{1}{2L_{\mathcal{F}}}\mathbb{E}\big[\big\|\nabla\mathcal{F}_k^{\top}(\boldsymbol{\alpha}_k)\big\|_2^2\big]. \tag{27}$$

Besides, $h(\boldsymbol{\alpha})$ is a convex function, and it satisfies $h(\boldsymbol{\alpha}_k) - h(\boldsymbol{\alpha}_{k+1}) \leq \nabla h^{\top}(\boldsymbol{\alpha}_k)(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})$, and then

$$\begin{aligned}\mathbb{E}[f(\boldsymbol{\alpha}_{k+1})] &= \mathbb{E}[g(\boldsymbol{\alpha}_{k+1}) - h(\boldsymbol{\alpha}_{k+1})] \\ &\leq \mathbb{E}[g(\boldsymbol{\alpha}_{k+1}) - h(\boldsymbol{\alpha}_k) + \nabla h^{\top}(\boldsymbol{\alpha}_k)(\boldsymbol{\alpha}_k - \boldsymbol{\alpha}_{k+1})] \\ &= \mathbb{E}[\mathcal{F}_k(\boldsymbol{\alpha}_{k+1})].\end{aligned} \tag{28}$$

Note that $f(\boldsymbol{\alpha}_k) = \mathcal{F}_k(\boldsymbol{\alpha}_k)$, from (27) and (28), we obtain

$$\mathbb{E}[f(\boldsymbol{\alpha}_{k+1})] \leq \mathbb{E}[f(\boldsymbol{\alpha}_k)] - \frac{1}{2L_{\mathcal{F}}}\mathbb{E}\big[\|\nabla f(\boldsymbol{\alpha}_k)\|_2^2\big].$$

Finally, we arrive at the following formula as we expect:

$$\mathbb{E}[f(\boldsymbol{\alpha}_k) - f(\boldsymbol{\alpha}_{k+1})] \geq \frac{1}{2L_{\mathcal{F}}}\mathbb{E}\big[\|\nabla f(\boldsymbol{\alpha}_k)\|_2^2\big]. \tag{29}$$

By Lemma 2 and the above-mentioned formula, we verify the convergence of $\{\boldsymbol{\alpha}_k\}_{k=0}^{\infty}$. Such proof is definitely suitable for the proposed IKLR model. To be specific, $L_{\mathcal{F}}$ in our IKLR model can be solved during the proof of Theorem 1, and thus it is set to $L_{\mathcal{F}} = \lambda\|\boldsymbol{K}_+\| + (\|\boldsymbol{K}\|/4n)$.

Remark: The convergence analysis of a stochastic proximal difference of the convex algorithm has been discussed in [35], which relies on a known bounded residual error δ, namely, $\mathcal{F}_k(\boldsymbol{\alpha}_{k+1}) \leq \mathcal{F}_k(\boldsymbol{\alpha}_k^*) + \delta$ as demonstrated. However, the residual error δ is usually not known. In our analysis, the decrease is related to the gradient, which could be calculated or estimated in each iteration.

V. Experiments

In this section, we carry out experiments to show the performance of the IKLR model with two indefinite kernels on a collection of multimodal data sets from computer vision and machine learning fields. The experiments implemented in MATLAB are repeated over 10 runs on a PC with Intel i5-6500 CPU (3.20 GHz) and 8-GB memory. The source code of the proposed method can be found in the website.[3]

A. Experiment Setup

Here, we describe kernel settings, the compared algorithms, and other settings of the experiments.

1) Kernel Setting: Three kernels including one PD and two indefinite ones are chosen to fully evaluate the performance of our method. As a representative PD kernel, the radius basis function (RBF) kernel, i.e., $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2/\sigma^2)$ with the kernel width σ, is chosen for comparison.

For indefinite kernels, we first choose the truncated ℓ_1 distance (TL1) indefinite kernel [37], namely, $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max\{\tau - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_1, 0\}$, and then incorporate it into our model. As discussed in [37], the performance of the TL1 kernel is robust to τ, and thus, we set $\tau = 0.7$ m as suggested.

Apart from a delicately designed TL1 kernel, we extend the RBF kernel from the Euclidean space to a Riemannian manifold with a geodesic metric [16]. Here, we use the covariance matrix descriptor [38] on the space of $d \times d$ symmetric positive definite (SPD) matrices, namely, Sym_d^+. Let $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ be the two descriptors (SPD matrices), if the Euclidean distance in the Gaussian kernel between such two descriptors is used, the derived Gaussian kernel is PD. In comparison, to define a kernel on the Riemannian manifold, we would like to replace the Euclidean distance by a more accurate geodesic distance on the manifold. However, not all geodesic metrics yield a PD kernel. Feragen *et al.* [17] point out that the geodesic Gaussian kernels on the Riemannian manifolds are PD only if

[3] http://www.lfhsgre.org

TABLE I

PROPERTIES OF DIFFERENT METRICS ON Sym_d^+. WE ANALYZE POSITIVE DEFINITENESS OF GAUSSIAN KERNELS GENERATED BY DIFFERENT METRICS

Metric	Formulation (d)	Geodesic distance?	Does $k = \exp(-d^2/\sigma^2)$ define a positive definite kernel?	Time complexity
Euclidean	$\|\boldsymbol{S}_1 - \boldsymbol{S}_2\|_{\mathrm{F}}$	No	Yes	$\mathcal{O}(d^2)$
Log-Euclidean (Log-E)	$\|\log(\boldsymbol{S}_1) - \log(\boldsymbol{S}_2)\|_{\mathrm{F}}$	Yes	Yes	$\mathcal{O}(d^3)$
Affine-Invariant (Aff-I)	$\|\log(\boldsymbol{S}_1^{-\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{-\frac{1}{2}})\|_{\mathrm{F}}$	Yes	No	$\mathcal{O}(d^3)$

the geodesic metric space is *flat in the sense of Alexandrov*. Here, we summarize the definitions and properties of some representative metrics for Sym_d^+ in Table I. It can be observed that the geodesic Gaussian kernel ("log-Euclidean") is still PD since the geodesic metric space derived by log-Euclidean is *flat*, while the affine-invariant metric results in an indefinite one. Based on this, in our experiment, we take the affine-invariant kernel as an example of indefinite kernels to test the proposed IKLR model.

2) Compared Methods: We conduct the proposed IKLR model with two versions, including CCICP-GD and CCICP-SGD, in which the inexact parameter ϵ in CCICP-GD and CCICP-SGD is set to 1 and 0.0001, respectively.[4] The proposed algorithms are compared with other representative indefinite kernel methods, "Flip," "Clip," and "Shift" [39]: these methods use the spectrum transformation to directly transform the non-PSD kernel matrix into a PSD one; "TDCASVM" [24]: an approach incorporates CCCP into the SMO-type algorithm for SVM with an indefinite kernel; and "KSVM" [19]: this method formulates the indefinite SVM as a min–max problem, and then solves this optimization problem in Kreĭn space. In essence, our method and "TDCASVM" directly solve a nonconvex optimization problem, while the objective function in other algorithms has been transformed into a convex form. Specifically, two PD kernels, i.e., the Gaussian kernel and log-Euclidean kernel are incorporated into SVM and KLR as baseline methods for comparisons.

3) Parameter Setting: In our experiment, we choose λ, the kernel width σ in the Gaussian kernel, and C in SVM by fivefold cross validation over $\{0.0001, 0.001, 0.01, 0.1, 1, 5, 10\}$ on the training set. For each data set, half of the data are randomly picked up for training and the rest for the test process.

B. Results on UCI Database

1) Description of Data Sets: Table II lists a brief description of 20 data sets from the UCI machine learning repository [40] including the feature dimension m, the number of data points n (the training and test data have been divided in some data sets such as *monks1*, *monks2*, and *monks3*), and the minimum and maximum eigenvalues $\mu_{\min}$ and $\mu_{\max}$ of the TL1 kernel.

2) Results on Small-Scale Data Sets: Table III reports the average classification accuracy on the test data and its standard deviation. One can see that the proposed IKLR model with CCICP-GD and CCICP-SGD algorithms achieves a promising performance in most data sets such as *australian*, *monks1*, and *splice*. TDCASVM, KSVM, and the baseline (KLR) also provide a comparable performance on some data sets including *monks2*, *spect*, and *diabetic*. Meanwhile, the performance of kernel approximation methods is often inferior to other indefinite learning-based algorithms. Besides, we observe that the training TL1 kernel in several data sets such as *climate* and *parkinsons* is still PD. In these data sets, there is no distinct difference on the classification accuracy for most compared algorithms.

In terms of these algorithms, comparing the baseline (KLR), we find that the used TL1 kernel is able to adaptively find the partition and locally fit nonlinearity. However, in this paper, we do not want to claim that the TL1 kernel is better than the RBF kernel, as their performance is actually dependent on the specific task. Instead, our aim is to show the performance of the indefinite kernels in KLR. Since the indefinite kernels contain definite ones, it can be expected that a suitable indefinite kernel can outperform a PD kernel.

Besides, it can be noted that KSVM investigates its dual form in RKKS, while we directly focus on the nonconvex primal form of the IKLR model using the representer theorem. The coefficient vector $\boldsymbol{\alpha}$ in IKLR should not be interpreted as a

TABLE II

STATISTICS FOR VARIOUS DATA SETS. SPECIFICALLY, THE LARGER THAN SMALL-SCALE DATA SETS ARE HIGHLIGHTED BY **BOLD**

Dataset	m(feature)	n(#num)	$\mu_{\min}$	$\mu_{\max}$
australian	14	690	-0.006	2212.5
breast-cancer	10	699	-4.408	1593.3
parkinsons	23	195	0.127	1200.4
climate	20	540	0.174	1944.7
diabetic	19	1151	-0.003	6133.1
fertility	9	100	-0.042	164.98
sonar	60	208	1.452	3024.6
SPECT	21	80	-1.145	353.11
haberman	3	306	-0.204	215.23
heart	13	270	-0.084	695.16
ionosphere	33	351	0.085	2489.7
monks1	6	124	-2.094	94.077
monks2	6	169	-2.535	131.14
monks3	6	122	-1.764	95.376
splice	60	1000	-1.325	2885.3
transfusion	4	748	-0.336	818.74
EEG	14	14980	-0.444	7312.0
ijcnn1-tr	26	35000	-0.018	28945
guide1-t	4	4000	-0.805	4116.7
madelon	500	2000	14.825	27015

[4]Since SGD achieves the similar purpose with the inexact scheme to solve the subproblem, the inexact parameter ϵ in CCICP-SGD is fixed with an "exact" one.

TABLE III
CLASSIFICATION ACCURACY (MEAN ± STD. DEVIATION) OF EACH COMPARED METHOD. THE BEST PERFORMANCE IS HIGHLIGHTED IN **BOLD**

	KLR(RBF)	Flip	Clip	Shift	TDCASVM	KSVM	CCICP-GD	CCICP-SGD
australian	0.790±0.021	0.795±0.059	0.790±0.078	0.672±0.085	0.857±0.013	0.835±0.018	0.846±0.027	**0.865**±0.007
breast	0.968±0.004	0.964±0.012	0.965±0.007	0.921±0.039	0.946±0.008	**0.971**±0.008	0.959±0.014	0.967±0.008
climate	**0.939**±0.012	0.909±0.047	0.838±0.036	0.839±0.061	0.904±0.031	0.918±0.011	0.912±0.013	0.923±0.018
diabetic	**0.625**±0.007	0.532±0.016	0.529±0.012	0.534±0.016	0.545±0.024	0.570±0.020	0.552±0.034	0.516±0.065
fertility	0.852±0.023	**0.896**±0.025	0.888±0.039	0.884±0.040	0.880±0.025	0.873±0.034	0.780±0.040	0.860±0.024
haberman	0.742±0.040	0.678±0.039	0.725±0.037	0.733±0.023	0.722±0.031	0.730±0.034	0.727±0.035	**0.766**±0.020
heart	0.816±0.044	0.758±0.076	0.760±0.035	0.747±0.059	**0.823**±0.031	0.757±0.043	0.803±0.046	0.809±0.023
ionosphere	0.907±0.029	0.891±0.032	0.900±0.032	0.841±0.022	0.903±0.022	0.899±0.014	0.901±0.015	**0.915**±0.028
monks1	0.668±0.052	0.695±0.075	0.648±0.070	0.685±0.063	0.678±0.021	0.671±0.035	**0.765**±0.065	0.752±0.041
monks2	0.662±0.071	0.611±0.047	0.593±0.025	0.489±0.092	**0.743**±0.018	0.626±0.037	0.669±0.093	0.617±0.046
monks3	0.779±0.073	0.723±0.090	0.805±0.021	0.870±0.036	0.734±0.045	0.640±0.083	0.830±0.072	**0.893**±0.031
parkinsons	**1.000**±0.000	0.990±0.010	0.999±0.003	0.998±0.007	**1.000**±0.000	0.945±0.039	**1.000**±0.000	**1.000**±0.000
sonar	0.789±0.022	0.546±0.045	0.539±0.042	0.504±0.054	0.704±0.024	0.608±0.072	**0.794**±0.060	0.690±0.121
SPECT	0.737±0.092	0.652±0.026	0.706±0.022	0.667±0.034	0.711±0.024	**0.893**±0.024	0.764±0.059	0.738±0.076
splice	0.642±0.093	0.513±0.017	0.619±0.057	0.604±0.033	0.751±0.075	0.515±0.029	**0.785**±0.050	0.588±0.047
transfusion	0.741±0.048	0.734±0.095	0.717±0.020	0.736±0.038	0.762±0.015	0.762±0.006	0.769±0.023	**0.774**±0.009

TABLE IV
RESULTS OF CCCP-GD, CCICP-GD, AND CCICP-SGD ON FOUR LARGER THAN SMALL-SCALE DATA SETS

Data sets	Methods	Accuracy	Training time(s)	Test time(s)
EEG	CCCP-GD	0.769±0.042	17171.0	0.1237
	CCICP-GD	0.725±0.042	848.89	0.1304
	CCICP-SGD	0.730±0.080	8776.4	0.1188
guide1-t	CCCP-GD	0.962±0.003	1314.3	0.0020
	CCICP-GD	0.955±0.003	47.23	0.0028
	CCICP-SGD	0.947±0.022	714.72	0.0021
ijcnn1-tr	CCCP-GD	0.912±0.002	51.22	0.2215
	CCICP-GD	0.914±0.003	14.65	0.2273
	CCICP-SGD	0.914±0.006	28.23	0.2258
madelon	CCCP-GD	0.624±0.080	305.29	0.0008
	CCICP-GD	0.609±0.051	8.129	0.0064
	CCICP-SGD	0.601±0.028	160.71	0.0011

Lagrange multiplier, which is different from the dual variable in SVM. Therefore, our IKLR model is more flexible to learn the data distribution than KSVM. Admittedly, KLR and SVM-based algorithms including TDCASVM and KSVM have their respective pros and cons, and each solution might depend on a case-by-case basis.

3) Results on Larger Than Small-Scale Data Sets: Apart from experiments on several small-scale data sets, we also conduct the proposed CCICP algorithm on four larger small-scale data sets to further validate its effectiveness. Table IV reports the test accuracy, training time, and test time of the original CCCP, and the proposed CCICP-GD and CCICP-SGD. Specifically, CCCP-GD is taken as a baseline to evaluate the performance of the proposed two algorithms. In CCCP-GD, the inexact parameter ϵ is fixed with 0 in theoretical aspect while we set it to 0.0001 in practice. This small value in our experiments means that CCCP-GD is not equipped with any inexact scheme, which further guarantees that CCCP-GD is able to yield an accurate solution during each iteration.

On *EEG*, *guide1-t*, and *madelon* data sets, CCCP-GD achieves the best performance on classification accuracy, which is narrowly followed by CCICP-GD and CCICP-SGD. On the *ijcnn1-tr* data set, the above-mentioned three algorithms achieve the similar classification performance without a distinct difference. In terms of the computational cost during training, CCICP-GD is the most efficient, while CCCP-GD is much time consuming. Note that ϵ in the proposed CCICP-SGD algorithm is set to 0.0001, which makes the training process relatively inefficient. Specifically, we also discuss CCICP-SGD with $\epsilon = 1$ in Section V-G to see how fast it is. From the above-mentioned analyses, we can conclude that the proposed IKLR model with CCICP is often slightly inferior to the CCCP setting in the terms of classification accuracy, but the inexact scheme makes our method much efficient during the training process.

The experimental results on UCI database demonstrate the superiority of our IKLR model with PD or indefinite kernels. Besides, the inexact scheme including the early termination condition and a stochastic version is able to accelerate the training process.

C. Results on Yale Face Database

Apart from using the designed TL1 kernel on UCI database, we also illustrate the use of the geodesic Gaussian kernel on the Riemannian manifold in the IKLR model for face recognition on Sym_d^+. In the experiment, we choose the Yale face database B[5] to evaluate the performance of the proposed IKLR model with the affine-invariant kernel. This database contains 5760 single light source images of 10 subjects with each shot under 576 viewing conditions (9 poses × 64 illumination conditions). For every subject in a particular pose, an image with ambient (background) illumination is also captured. Hence, the total number of images is in fact $5760 + 90 = 5850$. All images have been cropped based on the location of eyes. The size of each image is 192 × 168. Fig. 1 shows some image examples of this database.

To compute the affine-invariant kernel (shown in Table I) on the Riemannian manifold, we use covariance descriptors [38] computed from the feature vector $[x, y, I_{xy}, |G_x|, |G_y|, (G_x^2 + G_y^2)^{1/2}, |G_{xx}|, |G_{yy}|, \arctan(|G_x|/|G_y|)]$,

[5]http://vision.ucsd.edu/content/yale-face-database

TABLE V
STATISTICS OF THE AFFINE-INVARIANT (AFF-I) KERNEL AND THE TEST CLASSIFICATION ACCURACY(%) ON YALE FACE DATABASE B

$\mu_{\min}$	$\mu_{\max}$	SVM(Log-E)	KLR(Log-E)	Flip	Clip	Shift	TDCASVM	KSVM	CCICP-GD	CCICP-SGD
-160.28	670.05	97.6±0.5	97.3±0.3	97.4±0.2	95.6±2.9	96.0±1.6	97.7±0.5	97.8±0.3	98.3±0.8	97.8±1.7

Fig. 1. Some examples from the Yale Face Database B.

where x and y are the pixel locations, I_{xy} is the grayscale value at xy-coordinate location, and G_x and G_y are the first-order intensity derivatives. Likewise, G_{xx} and G_{yy} are the second-order intensity derivatives. Accordingly, each face image in this database can be represented by the covariance matrix, i.e., a 9×9 SPD matrix. Then, the similarity between two images can be evaluated by the affine-invariant kernel.

Table V presents the statistics including the minimum and maximum eigenvalues of the affine-invariant kernel, i.e., $\mu_{\min}$ and $\mu_{\max}$. Note that such kernel on this data shows highly indefinite. We compare the proposed CCICP-GD and CCICP-SGD algorithms with five indefinite learning-based methods with the affine-invariant kernel including "Flip," "Clip," "Shift," "TDCASVM," and "KSVM." Also, the log-Euclidean kernel [16], a PD kernel, is incorporated into two representative classifiers such as SVM and KLR for comparisons. Table V reports the average test accuracy(%) across the above-mentioned algorithms. One can see that these kernel approximation-based methods "Flip," "Clip," and "Shift" do not achieve satisfactory performance when compared with other PD/indefinite kernel learning-based algorithms. This is because the above-mentioned methods actually change the indefinite matrix itself. Among these indefinite learning-based algorithms, the proposed CCICP-GD algorithm performs better than "TDCASVM" and "KSVM" with a margin of 0.6% and 0.5% on the test classification accuracy. Besides, the proposed CCICP-SGD algorithm achieves a comparable performance among these methods. The experimental results on this data set reinforce to demonstrate the effectiveness of the proposed algorithms with various indefinite kernels.

D. Effect of the Inexact Parameter ϵ

As aforementioned, the only difference between CCCP-GD and CCICP-GD is the selection of the inexact parameter ϵ. When ϵ approaches to zero, the CCICP-GD algorithm degenerates to a standard CCCP-GD algorithm. In our experiment, ϵ is set to 0.0001 in the CCCP-GD algorithm while we choose $\epsilon = 1$ in CCICP-GD. Based on this, this section investigates how its variation (i.e., 0.0001, 0.001, 0.01, 0.1, 0.5, 1, and 5) in the inexact solving scheme influences the test accuracy and

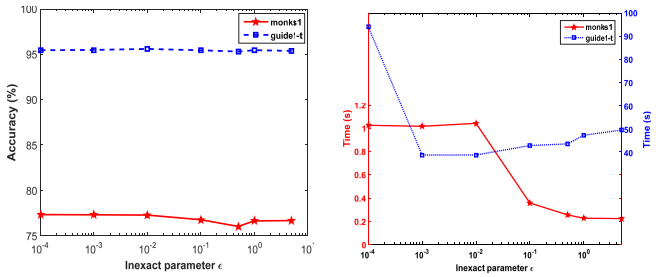

Fig. 2. Influence of tuning ϵ on *monks1* (red) and *guide1-t* (blue), with (a) accuracy and (b) time versus iteration.

TABLE VI
CCICP-GD AND CCICP-SGD WITH DIFFERENT RANDOM INITIALIZATIONS ON THE *MONKS1* AND *GUIDE1-T* DATA SET

Method	CCICP-GD		CCICP-SGD	
Initialization	*monks1*	*guide1-t*	*monks1*	*guide1-t*
$\boldsymbol{\alpha}^{(0)} = \mathbf{0}$	0.694	0.925	0.662±0.050	0.926±0.003
$\boldsymbol{\alpha}^{(0)} = \mathbf{1}$	0.697	0.925	0.675±0.037	0.919±0.016
$\boldsymbol{\alpha}^{(0)} = -\mathbf{1}$	0.694	0.925	0.661±0.043	0.923 ±0.008
$\alpha_i^{(0)} \in (0, 1)$	0.706	0.927	0.663±0.054	0.926±0.049

the computational cost during training. Two data sets appeared in Section V-B are used here for our experiments, namely, a small-scale data set *monks1* and a larger one *guide1-t*.

Fig. 2(a) illustrates that the performance of CCICP-GD is generally not sensitive to ϵ on such two different types of data sets. In Fig. 2(b), on *monks1*, the training time cost does not dramatically decrease when ϵ ranges from 0.0001 to 0.01, and then it rapidly falls down. We can conclude that CCICP-GD ($\epsilon = 1$) is much efficient than CCCP-GD ($\epsilon = 0.0001$), and thus such tendency demonstrates the effectiveness of the proposed inexact scheme. Meanwhile, on *guide1-t*, the setting with $\epsilon = 0.001$ spends the minimum time during training, which almost cuts by half when compared with the situation of initial value $\epsilon = 0.0001$. Afterward, the time cost steadily increases, which shows an "abnormal" tendency on ϵ. This is because the algorithm with an inexact solution sometimes requires more iterations to converge to a stationary point. However, CCICP-GD ($\epsilon = 1$) is still efficient than CCCP-GD ($\epsilon = 0.0001$) in this data set. Generally, CCICP-GD with larger ϵ often spends less training time than the setting with smaller one to converge. This is because the termination condition can be significantly relaxed, which has been well demonstrated on these two data sets.

E. Algorithm Convergence

Fig. 3 shows the convergence of IKLR with three optimization algorithms on *monks1* and *ijcnn1-tr*. It can be observed

TABLE VII
RESULTS OF CCICP-SGD AND CCICP-SGD-I ON FOUR LARGER THAN SMALL-SCALE DATA SETS

Dataset	EEG		guide1-t		ijcnn1-tr		madelon	
Method	CCICP-SGD	CCICP-SGD-I	CCICP-SGD	CCICP-SGD-I	CCICP-SGD	CCICP-SGD-I	CCICP-SGD	CCICP-SGD-I
Accuracy	0.630±0.042	0.590±0.052	0.947±0.022	0.876±0.037	0.914±0.006	0.912±0.003	0.601±0.028	0.550±0.065
Training time	8776.4	71.2754	714.72	1.4146	28.23	2.445	160.71	2.543
Test time	0.1188	0.0235	0.0021	0.0023	0.2258	0.2378	0.0011	0.0025

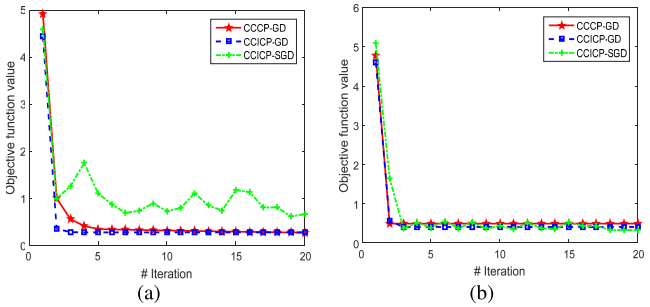

Fig. 3. Convergence plots for CCCP-GD (red), CCICP-GD (blue), and CCICP-SGD (green) on (a) *monks1* and (b) *ijcnn1-tr*, respectively.

that in Fig. 3(a), on the *monks1* data set, CCICP-GD converges within five iterations but CCCP-GD takes 16 iterations to converge. In Fig. 3(b), both CCCP-GD and CCICP-GD converge fast on the *ijcnn1-tr* data set. The above-mentioned two gradient-based algorithms (CCCP-GD and CCICP-GD) monotonically decrease in each iteration. However, in our CCICP-SGD version, it cannot be guaranteed to monotonically decrease due to its random scheme. Instead, it just converges to a stationary point in expectation as shown in Fig. 3.

F. Different Random Initializations

The proposed algorithms, CCICP-GD and CCICP-SGD, have been experimentally demonstrated to converge as illustrated in Section V-E. Since the IKLR model is nonconvex, different initializations might lead to different stationary points. Here, we choose two data sets, *monks1* and *guide1-t*, to investigate the influence of our algorithms with different initializations on the final classification accuracy. Such two data sets are conducted with 10 runs on a fixed (or predefined) training and test data for fair comparisons. As suggested in [41], in our experiment, we choose four different initializations with small values, i.e., $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$, $\boldsymbol{\alpha}^{(0)} = \mathbf{1}$, $\boldsymbol{\alpha}^{(0)} = -\mathbf{1}$, and the randomly initialization $\alpha_i^{(0)} \in (0, 1)$. By doing so, such small initialization values can guarantee that the objective function value in (6) is always positive during the optimization process. Table VI demonstrates that different initializations near zero often lead to slight fluctuation on the final classification accuracy.

G. Discussion on CCICP-SGD

As aforementioned in Section V-A2, the inexact parameter ϵ in CCICP-SGD is fixed to 0.0001 because SGD achieves the similar purpose with the inexact scheme, i.e., $\epsilon = 1$. However, in Table IV, it can be observed that the CCICP-GD is much efficient than CCICP-SGD. Such time cost reduction motivates us to see how fast our algorithm can be when SGD comes to the inexact scheme. Accordingly, we investigate the performance of CCICP-SGD with the early termination condition (i.e., $\epsilon = 1$), termed as "CCICP-SGD-I."

Table VII reports the classification accuracy and the computation cost of CCICP-SGD and CCICP-SGD-I on four larger small-scale data sets. One can see that CCICP-SGD-I degrades the test accuracy to some extent when compared with CCICP-SGD on *EEG*, *guide1-t*, and *madelon*. However, CCICP-SGD-I equipped with the inexact scheme extremely accelerates the training process, of which the training time is about one-hundreds or less than that of CCICP-SGD. On *ijcnn1-tr*, CCICP-SGD-I is more efficient than CCICP-SGD without too much degeneracy on the classification accuracy.

VI. CONCLUSION

In this paper, we investigate KLR with indefinite kernels in theoretical and algorithmic aspects. The derived IKLR model is nonconvex and further analyzed in RKKS with explicit demonstration due to the nonpositive definite kernels. Such nonconvex problem can be effectively and efficiently solved by the proposed CCICP equipped with two approximation schemes. Its GD version using an early stop scheme is able to make the training process efficient; the stochastic variant of CCICP also has the capability of accelerating the solving process. The convergence analyses of CCICP-GD and CCICP-SGD are conducted with theoretical guarantees and experimental validation. The classification accuracy of the proposed IKLR model on several benchmarks demonstrates its effectiveness when compared to other PD/indefinite kernel learning methods.

ACKNOWLEDGMENT

The authors would like to thank J. Xie from Shanghai Jiao Tong University for his work on the implementation of TDCASVM, and also sincerely appreciate the anonymous reviewers for their insightful comments.

REFERENCES

[1] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2003.
[2] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 2000.
[3] J. Zhu and T. Hastie, "Kernel logistic regression and the import vector machine," *J. Comput. Graph. Statist.*, vol. 14, no. 1, pp. 185–205, 2002.
[4] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3249–3260, Jul. 2016.

[5] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 155–161.
[6] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 551–556.
[7] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Adv. Comput. Math.*, vol. 13, Apr. 2000, Art. no. 1.
[8] F.-M. Schleif and P. Tino, "Indefinite proximity learning: A review," *Neural Comput.*, vol. 27, no. 10, pp. 2039–2096, 2015.
[9] H. Saigo, J.-P. Vert, N. Ueda, and T. Akutsu, "Protein homology detection using string alignment kernels," *Bioinformatics*, vol. 20, no. 11, pp. 1682–1689, 2004.
[10] N. M. Kriege, P.-L. Giscard, and R. Wilson, "On valid optimal assignment kernels and applications to graph classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1623–1631.
[11] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358–386, 2005.
[12] P.-F. Marteau and S. Gibet, "On recursive edit distance kernels with application to time series classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1121–1133, Jun. 2015.
[13] Y. Ying, C. Campbell, and M. Girolami, "Analysis of SVM with indefinite kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2205–2213.
[14] J. Zhang, L. Wang, L. Zhou, and W. Li, "Learning discriminative stein kernel for SPD matrices and its applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 1020–1033, May 2015.
[15] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang, "Deformed graph Laplacian for semisupervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2261–2274, Oct. 2015.
[16] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi, "Kernel methods on the Riemannian manifold of symmetric positive definite matrices," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2013, pp. 73–80.
[17] A. Feragen, F. Lauze, and S. Hauberg, "Geodesic exponential kernels: When curvature and linearity conflict," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3032–3042.
[18] C. S. Ong, X. Mary, S. Canu, and A. J. Smola, "Learning with non-positive kernels," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 81–89.
[19] G. Loosli, S. Canu, and C. S. Ong, "Learning SVM in Kreĭn spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 6, pp. 1204–1216, Jun. 2016.
[20] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer, "Classification on pairwise proximity data," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 11, 1999, pp. 438–444.
[21] E. Pekalska, P. Paclik, and R. P. W. Duin, "A generalized kernel approach to dissimilarity-based classification," *J. Mach. Learn. Res.*, vol. 2, pp. 175–211, Mar. 2002.
[22] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann, "Optimal cluster preserving embedding of nonmetric proximity data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1540–1551, Dec. 2003.
[23] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.
[24] F. B. Akoa, "Combining DC algorithms (DCAs) and decomposition techniques for the training of nonpositive–semidefinite kernels," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1854–1872, Nov. 2008.
[25] H.-M. Xu, H. Xue, X.-H. Chen, and Y.-Y. Wang, "Solving indefinite kernel support vector machine with difference of convex functions programming," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1610–1616.
[26] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915–936, 2003.
[27] F. Liu, X. Huang, and J. Yang, "Indefinite kernel logistic regression," in *Proc. ACM Multimedia Conf.*, 2017, pp. 846–853.
[28] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. Conf. Comput. Learn. Theory*, 2000, pp. 416–426.
[29] J. Bognár, *Indefinite Inner Product Spaces*. New York, NY, USA: Springer, 1974.
[30] B. Haasdonk, "Feature space interpretation of SVMs with indefinite kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 482–492, Apr. 2005.
[31] A. Cotter, O. Shamir, N. Srebro, and K. Sridharan, "Better mini-batch algorithms via accelerated gradient methods," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1647–1655.
[32] I. Mitliagkas, C. Caramanis, and P. Jain, "Memory limited, streaming PCA," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2886–2894.
[33] K. Goebel and W. A. Kirk, "A fixed point theorem for asymptotically nonexpansive mappings," *Proc. Amer. Math. Soc.*, vol. 35, no. 1, pp. 171–174, 1972.
[34] G. Lan and S. Ghadimi, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: Shrinking procedures and optimal algorithms," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2061–2089, 2010.
[35] A. Nitanda and T. Suzuki, "Stochastic difference of convex algorithm and its application to training deep boltzmann machines," in *Proc. Artif. Intell. Stat.*, 2017, pp. 470–478.
[36] G. R. Lanckriet and B. K. Sriperumbudur, "On the convergence of the concave-convex procedure," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1759–1767.
[37] X. Huang, J. A. K. Suykens, S. Wang, J. Hornegger, and A. Maier, "Classification with truncated ℓ_1 distance kernel," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 2025–2030, May 2018.
[38] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 589–600.
[39] G. Wu, E. Y. Chang, and Z. Zhang, "An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines," Univ. California, Santa Barbara, Santa Barbara, CA, USA, Tech. Rep., 2005.
[40] C. Blake and J. Christopher Merz, "UCI repository of machine learning databases," Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 1998.
[41] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2933–2941.

Fanghui Liu received the B.S. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2014. He is currently pursuing the Ph.D. degree with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China.

His current research interests include machine learning and optimization with respect to kernel learning, visual tracking, and optimization.

Xiaolin Huang (S'10–M'12–SM'18) received the B.S. degree in control science and engineering and the B.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2006, and the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2012.

From 2012 to 2015, he was a Post-Doctoral Researcher with ESAT-STADIUS, KU Leuven, Leuven, Belgium. He was with the Pattern Recognition Laboratory, Friedrich-Alexander-Universität Erlangen–Nürnberg, Erlangen, Germany, where he was a Group Head. Since 2016, he has been an Associate Professor with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China. His current research interests include machine learning, optimization, and their applications.

Dr. Huang was an Alexander von Humboldt Fellow. He was a recipient of the 1000-Talent Award (Young Program) in 2017.

Chen Gong (M'17) received the dual Ph.D. degrees from Shanghai Jiao Tong University (SJTU) and the University of Technology Sydney, in 2016, under the supervision of Prof. J. Yang and Prof. D. Tao, respectively.

He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China. He has authored or co-authored more than 30 technical papers at prominent journals and conferences such as the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CYBERNETICS, CVPR, AAAI, IJCAI, ICDM, and so on. His current research interests include machine learning and data mining.

Dr. Gong was a recipient of the Excellent Doctorial Dissertation Award at SJTU and the Chinese Association for Artificial Intelligence. He was also enrolled by the Summit of the Six Top Talents Program of Jiangsu Province, China.

Jie Yang received the Ph.D. degree from the Department of Computer Science, Hamburg University, Germany, in 1994.

He is currently a Professor with the Institute of Image Processing and Pattern recognition, Shanghai Jiao Tong University, Shanghai, China. He has been involved in research projects (e.g., National Science Foundation and 863 National High Tech. Plan). He has authored or co-authored one book in Germany, and authored more than 300 journal papers. His current research interests include object detection and recognition, data fusion and data mining, and medical image processing.

Johan A. K. Suykens (SM'05–F'15) was born in Willebroek, Belgium, in 1966. He received the M.S. degree in electromechanical engineering and the Ph.D. degree in applied sciences from Katholieke Universiteit Leuven (KU Leuven), Leuven, Belgium, in 1989 and 1995, respectively.

Since 1996, he has been a Visiting Post-Doctoral Researcher with the University of California at Berkeley, Berkeley, CA, USA. He has been a Post-Doctoral Researcher with the Fund for Scientific Research FWO Flanders, Brussels, Belgium. He is currently a Professor with KU Leuven. He received an ERC Advanced Grant in 2011 and 2017. He has authored *Artificial Neural Networks for Modelling and Control of Non-Linear Systems* (Kluwer Academic Publishers) and *Least Squares Support Vector Machines* (World Scientific), co-authored *Cellular Neural Networks, Multi-Scroll Chaos and Synchronization* (World Scientific), and edited *Nonlinear Modeling: Advanced Black-Box Techniques* (Kluwer Academic Publishers) and *Advances in Learning Theory: Methods, Models and Applications* (IOS Press).

Dr. Suykens was an Organizer of an International Workshop on Nonlinear Modeling with Time-series Prediction Competition in 1998. He has served as a Director and an Organizer for the NATO Advanced Study Institute on Learning Theory and Practice (Leuven 2002), a Program Co-Chair for the International Joint Conference on Neural Networks in 2004 and the International Symposium on Nonlinear Theory and its Applications in 2005, an Organizer for the International Symposium on Synchronization in Complex Networks in 2007, a Co-Organizer for the NIPS 2010 workshop on Tensors, Kernels, and Machine Learning, and the Chair for ROKS 2013. He was a recipient of the IEEE Signal Processing Society 1999 Best Paper (Senior) Award, several best paper awards at International Conferences, and the International Neural Networks Society INNS 2000 Young Investigator Award for significant contributions in the field of neural networks. He served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1997 to 1999 and from 2004 to 2007 and the IEEE TRANSACTIONS ON NEURAL NETWORKS from 1998 to 2009.