Indefinite Kernel Logistic Regression

Fanghui Liu
Institute of Image Processing and Pattern Recognition
Shanghai Jiao Tong University
Shanghai, China 200240
lfhsgre@outlook.com

Xiaolin Huang
Institute of Image Processing and Pattern Recognition
Shanghai Jiao Tong University
Shanghai, China 200240
xiaolinhuang@sjtu.edu.cn

Jie Yang∗
Institute of Image Processing and Pattern Recognition
Shanghai Jiao Tong University
Shanghai, China 200240
jieyang@sjtu.edu.cn

ABSTRACT

Traditionally, kernel learning methods requires positive definiteness on the kernel, which is too strict and excludes many sophisticated similarities, that are indefinite, in multimedia area. To utilize those indefinite kernels, indefinite learning methods are of great interests. This paper aims at the extension of the logistic regression from positive semi-definite kernels to indefinite kernels. The model, called indefinite kernel logistic regression (IKLR), keeps consistency to the regular KLR in formulation but it essentially becomes non-convex. Thanks to the positive decomposition of an indefinite matrix, IKLR can be transformed into a difference of two convex models, which follows the use of concave-convex procedure. Moreover, we employ an inexact solving scheme to speed up the sub-problem and develop a concave-inexact-convex procedure (CCICP) algorithm with theoretical convergence analysis. Systematical experiments on multi-modal datasets demonstrate the superiority of the proposed IKLR method over kernel logistic regression with positive definite kernels and other state-of-the-art indefinite learning based algorithms.

KEYWORDS

indefinite kernel, logistic regression, concave-inexact-convex procedure

ACM Reference format:
Fanghui Liu, Xiaolin Huang, and Jie Yang. 2017. Indefinite Kernel Logistic Regression. In Proceedings of "***, ***", ***; 8 pages.
DOI: ******

1 INTRODUCTION

Kernel methods [17] are powerful statistical machine learning techniques, which have been widely and successfully used. The representative kernel-based algorithms include Support Vector Machine (SVM, [20]) with kernels, Kernel Logistic Regression (KLR, [25]), Kernel Fisher Discriminant Analysis (KFDA, [13]), and so on. In these kernel-based methods, the corresponding kernel matrix is required to be symmetric and positive semi-definite to satisfy Mercer’s condition. By doing so, the above methods can be well analyzed with solid theoretical foundations in the reproducing kernel Hilbert spaces (RKHS) [5].

However, in practice, we often meet some sophisticated similarity or dissimilarity measures that are either indefinite (real, symmetric, but not positive semi-positive) or for which the Mercer’s condition is difficult to verify. For example, in multimedia area, one can use the human-judged similarities between concepts and words in music recommendation [21], video recommendation [18], or utilize dynamic time warping [9] for time series, or consider the Kullback-Leibler divergence between probability distributions. In these cases, many learning models boil down to be non-convex due to the used indefinite kernel which violates Mercer’s condition. Hence, there is both practical and theoretical need to properly handle these measures.

To use indefinite similarities in classification task, there have been some discussions, mainly on SVM. In theory, learning with indefinite kernels is discussed in the Reproducing Kernel Krein Spaces (RKKS) [11, 12], instead of the conventional reproducing kernel Hilbert spaces (RKHS) for PSD kernels. In practice, there are two kinds of algorithms to deal with indefinite kernels: i) kernel approximation and ii) non-convex optimization. Kernel approximation is to transform the indefinite kernel matrix into a positive semi-definite matrix by spectrum modification. For example, “clip”: all eigenvalues plus a positive constant until the smallest eigenvalue is zero. However, above operations actually change the indefinite matrix itself, and thus may cause in the loss of some important information involved with the kernel. The second approach is to directly solve the corresponding non-convex problem. For example, for SVM with indefinite kernels, [4] applies the SMO-type algorithm and [1, 23] uses the concave-convex procedure (CCCP) [24] algorithm that decomposes the objective function into the difference of two convex functions.

In this paper, we investigate the use of indefinite kernels on kernel logistic regression (KLR). KLR is a representative classifier and has been widely and successfully applied in many fields. However, indefinite kernel logistic regression (IKLR) has not yet been investigated in the past. To extend kernel used in KLR from PSD ones to indefinite kernels, we need to carefully discuss the indefinite model and the corresponding algorithm. In formulation, based on the representer theorem in Reproducing Kernel Krein Spaces (RKKS), the IKLR model shares the similar formulation with that of the regular KLR. However, using indefinite kernel makes the problem non-convex and hard to solve. To tackle this issue, we decompose the objective function into the difference of two convex functions and then the CCCP algorithm is applicable. Moreover, aiming at
large-scale problems in practice, a concave-inexact-convex procedure (CCICP) algorithm is proposed to obtaining early stop during each iteration. We theoretically demonstrate the convergence of our CCICP algorithm with the provable guarantee of the upper bounded error. Experiments on various multi-modal datasets suggest that in most cases our IKLR method outperforms not only the conventional kernel logistic regression with positive kernels but also other recent algorithms with indefinite kernels.

2 REVIEW: KERNEL LOGISTIC REGRESSION

The kernel logistic regression algorithm has been proven to be a powerful classifier with several merits [8] when compared with other traditional classifiers. It can naturally provide probabilities and straightforward extend to multi-class classification problems, and only require solving an unconstrained quadratic programming. Specifically, with a proper optimization algorithm [10], the computation time can be much less than that of other methods, such as SVM which needs to solve a constrained quadratic optimization problem.

Here we briefly introduce the kernel logistic regression in the binary classification setting. In this setting, given a training set \( \{(x_i, y_i)\}_{i=1}^n \), an instance space \( \mathbf{X} \), an output space \( \mathbf{Y} \), and a training sample \( x_i \in \mathbf{X} \) with its corresponding label \( y_i \in \{+1, -1\} \) in the space \( \mathbf{Y} \). We aim to learn a function \( f : \mathbf{X} \to \mathbf{Y} \) based on these \( n \) training samples, so that when given a new input \( z \in \mathbb{R}^m \) (\( m \) is the feature dimension) from the test sample set \( Z = \{z_1, z_2, \cdots, z_s\} \) with \( s \) test samples, we can predict its label \( y \). Many people have noted the relationship between a classifier (e.g. SVM, logistic regression) and regularized function estimation in the reproducing kernel Hilbert spaces (RKHS) [5]. For instance, fitting a logistic regression problem is equivalent to:

\[
\min_{f \in \mathcal{H}} \frac{1}{2} \|f\|_{\mathcal{H}}^2 + \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + \exp\left(-y_i f(x_i)\right)\right),
\]

(1)

where \( \mathcal{H} \) is the RKHS generated by the kernel \( \mathcal{K}(\cdot, \cdot) \), and \( \lambda \) is the regularization parameter. Generally, the discriminant function is formulated as \( f(x) = w^\top x + b \), where \( w \in \mathbb{R}^m \) is a weight vector parameterizing the space of linear functions mapping from \( \mathbf{X} \) to \( \mathbf{Y} \). By using the representor theorem [16] in RKHS, the optimal \( f^*(x) \) can be formulated as:

\[
f^*(x) = \sum_{i=1}^{n} \beta_i \mathcal{K}(x_i, x),
\]

(2)

where \( \mathcal{K} \) is a kernel function in RKHS and the coefficient vector \( \beta \in \mathbb{R}^n \). Accordingly, the formulation of kernel logistic regression can be obtained as:

\[
\begin{align*}
\min_{\beta} & \quad \frac{1}{2} \beta^\top \mathcal{K} \beta + \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + \exp\left(-y_i \sum_{j=1}^{n} \beta_j \mathcal{K}_{ij}\right)\right),
\end{align*}
\]

(3)

where \( \mathcal{K}_{ij} = \mathcal{K}(x_i, x_j) \) is a kernel matrix. With some abuse of notation, in [25], Eq. (3) can be written in a compact form:

\[
\begin{align*}
\min_{\beta} & \quad \frac{1}{2} \beta^\top \mathcal{K} \beta + \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + \exp\left(-y \circ \mathcal{K} \beta\right)\right),
\end{align*}
\]

(4)

where \( 1 \) denotes all-one vector and \( y = (y_1, y_2, \cdots, y_n)^\top \). Traditionally, in Eq. (4), we require the positiveness on the kernel matrix \( \mathcal{K} \), and thus the optimization problem is formulated as a convex unconstrained quadratic programming. To find the optimal \( \beta \), the Newton-Raphson method can be used to iteratively solve the objective function.

3 INDEFINITE LEARNING IN KERNEL LOGISTIC REGRESSION

3.1 The IKLR Model

In indefinite learning, using indefinite kernels in Eq. (4) makes Mercer’s theorem not applicable, which means that the functional space spanned by indefinite kernels does not belongs to RKHS, and thus the optimal \( f^*(x) \) cannot be represented by that form in Eq. (2). To tackle indefinite kernels in theory, the Reproducing Kernel Kreın Spaces (RKKS) [12] is introduced to provide a justification for feature space interpretation. In this case, the primal optimization problem of our IKLR model is formulated as a stabilization problem instead of a minimization problem. We reformulate Eq. (1) in RKKS, namely:

\[
\begin{align*}
\text{stabilize} & \quad \frac{\lambda}{2} \|f\|_{\mathcal{H}_k}^2 + \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + \exp\left(-y_i f(x_i)\right)\right),
\end{align*}
\]

(5)

where \( \mathcal{H}_k \) is the RKKS generated by the kernel \( \mathcal{K}(\cdot, \cdot) \). In [12], Ong et al. verify the existence of the representer theorem in RKKS. That is, if the optimization problem in Eq. (5) has a saddle point, it admits the following expansion, namely:

\[
f^* = \sum_{i=1}^{n} \beta_i \mathcal{K}(x_i, \cdot),
\]

(6)

where \( \mathcal{K} \) is a kernel function in RKKS and \( \beta \) is the coefficient vector. Since this condition is easily satisfied, the logistic regression problem with indefinite kernels can be expressed in RKKS, namely:

\[
\begin{align*}
\text{stabilize} & \quad \frac{\lambda}{2} \beta^\top \mathcal{K} \beta + \frac{1}{n} \sum_{i=1}^{n} \ln\left(1 + \exp\left(-y \circ \mathcal{K} \beta\right)\right),
\end{align*}
\]

(7)

where the label matrix \( y \in \mathbb{R}^{n \times n} \) is a diagonal matrix, the \( i \)th diagonal element of which is \( y_{ii} \). It can be seen that Eq. (7) shares the similar formulation with Eq. (4). However, due to the indefinite property of the kernel matrix \( \mathcal{K} \) in Eq. (7), such non-convex optimization problem must be analysed in the Kreın space.

3.2 Kernels in Kreın Space

The feature space in indefinite learning is given by a finite-dimensional Kreın space [6], which is an indefinite inner product space endowed with a Hilbertian topology, yet its inner product is no longer positive. The Kreın space is with explicitly definition in [3], namely:

Definition 3.1. An inner product space is a Kreın space \( \mathcal{H}_k \) if there exist two Hilbert spaces \( \mathcal{H}_a \) and \( \mathcal{H}_c \) spanning \( \mathcal{H}_k \) such that (1) All \( f \in \mathcal{H}_k \) can be decomposed into \( f = f_+ + f_- \), where \( f_+ \in \mathcal{H}_a \) and \( f_- \in \mathcal{H}_c \), respectively. (2) \( \forall f, g \in \mathcal{H}_k \), \( \langle f, g \rangle_{\mathcal{H}_a} = \langle f_+, g_+ \rangle_{\mathcal{H}_a} - \langle f_-, g_- \rangle_{\mathcal{H}_c} \).

The existence of RKKS implies that an indefinite kernel \( \mathcal{K} \) has a positive decomposition such that

\[
\mathcal{K}(u, v) = \mathcal{K}_+(u, v) - \mathcal{K}_-(u, v), \forall u, v \in \mathbf{X}.
\]

(8)
where $K_+$ and $K_-$ are two positive semi-definite kernels. Thus the objective function in Eq. (7) can be rewritten as:

$$
\text{stab}(\beta) = \frac{\lambda}{2} \beta^T (K_+ - K_-) \beta + \frac{1}{n} (K^T Y_k \beta) \ln \left(1 + \exp(-Y_k \beta) \right),
$$

(9)

To obtain $K_+$ and $K_-$, we often decompose the symmetric indefinite kernel matrix $K$ by eigenvalue decomposition, namely $K = V^T \Lambda V$, where $V$ is an orthogonal matrix and the diagonal matrix $\Lambda$ is defined as $\Lambda = \text{diag}(\mu_1, \mu_2, \cdots, \mu_n)$, elements of which are eigenvalues of $K$ with $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$. Without loss of generality, we assume that the first $v$ eigenvalues in $\Lambda$ are nonnegative and the remaining $n - v$ eigenvalues are smaller than zero. As a result, $K_+$ and $K_-$ can be formulated as:

$$
\begin{cases}
K_+ = V^T \text{diag}(\mu_1 + \rho, \cdots, \mu_v + \rho, \rho, \cdots, \rho) V; \\
K_- = V^T \text{diag}(\rho, \cdots, \rho, \rho - \mu_{v+1}, \cdots, \rho - \mu_n) V.
\end{cases}
$$

(10)

where $\rho$ is chosen as $\rho > -\mu_n$ to guarantee these two matrices $K_+$ and $K_-$ are positive definite. By this decomposition of $K$, the objective function in Eq. (9) can be decomposed as $f(\beta) = g(\beta) - h(\beta)$ with

$$
\begin{align*}
g(\beta) &= \frac{\lambda}{2} \beta^T K_+ \beta + \frac{1}{n} (K^T Y_k \beta) \ln \left(1 + \exp(-Y_k \beta) \right); \\
h(\beta) &= \frac{\lambda}{2} \beta^T K_- \beta.
\end{align*}
$$

(11)

4 IKLR MODEL WITH THE CCICP ALGORITHM

In this section, we further consider the IKLR model and then present a CCICP algorithm to efficiently solve it. Specifically, the convergence analysis of the CCICP algorithm in our IKLR model is also theoretically demonstrated.

4.1 Solving with CCICP

Based on above discussion, the objective function in Eq. (9) can be formulated as the difference of two convex functions $g(\beta)$ and $h(\beta)$. Therefore the CCCP is an appropriate choice to solve such problem. Here we briefly introduce the main idea of the CCCP and then detail our CCICP algorithm.

The CCCP decomposes the non-convex objective function $f(\beta)$ into the difference of two convex functions $g(\beta)$ and $h(\beta)$, namely $f(\beta) = g(\beta) - h(\beta)$. In each iteration, $h(\beta)$ is replaced by its first order Taylor approximation $h(\beta)$ around its current solution, and then the original non-convex objective function $f(\beta)$ can be approximated by the convex function $f(\beta) = g(\beta) - h(\beta)$. Accordingly, the sub-problem $f(\beta)$ can be formulated as a simpler convex form and then solved by an off-the-shelf convex solver (e.g., a gradient descent method). Theoretical analyses suggest that CCCP is able to converge to a local minima [19].

Nonetheless, it can be observed that such sub-problem needs to be solved at each iteration in the CCCP, which makes the solving process inefficient especially for a large-scale dataset. To tackle this issue, we propose a concave-inexact-convex procedure (CCICP), that only requires an inexact solution for the sub-problem. By doing so, the CCICP algorithm is able to effectively speed up the solving process. To be specific, the inexact solution $\beta^{(t+1)}$ lies in an $\delta$-neighborhood around the actual result $\beta^{(t)} = \text{argmin} \ f(\beta)$, that is $\hat{f}(\beta^{(t+1)}) \leq \hat{f}(\beta^{(t)})$. Here $\beta^{(t+1)}$ is bounded by $\beta^{(t)}$ with the following formula:

$$
\beta^{(t+1)} \in U_\delta(\beta^{(t)}) \triangleq \{ \beta \mid ||\beta - \beta^{(t)}|| \leq \delta \}.
$$

(12)

In this case, the KKT condition for $\beta^{(t+1)}$ does not hold, namely:

$$
\nabla_\beta \hat{f}(\beta) \nabla \beta ||\beta - \beta^{(t+1)}|| \neq 0.
$$

(13)

Without loss of generality, we assume that

$$
\nabla_\beta \hat{f}(\beta) \nabla \beta ||\beta - \beta^{(t+1)}|| = \epsilon ||\beta^{(t)}||,
$$

(14)

where $\epsilon$ corresponds to the bounded error, and the choice of which will be discussed in Section 4.2.

Based on above analyses, we detail the CCICP algorithm in our IKLR model. The function $h(\beta)$ is linearized by its Taylor approximation at $\beta^{(t)}$, namely: $h(\beta^{(t)}) = \lambda K_- \beta^{(t)}$. As a result, the sub-problem is reformulated as:

$$
\hat{f}(\beta, \beta^{(t)}) = \frac{\lambda}{2} \beta^T K_+ \beta + \frac{1}{n} \ln \left(1 + \exp(-Y_k \beta) \right) - \beta^T h(\beta^{(t)}).
$$

(15)

To solve this convex optimization problem, we employ the gradient descent method to find the optimal $\beta^{(t+1)}$ in Eq. (15), in which the gradient of $\hat{f}(\beta, \beta^{(t)})$ with respect to $\beta$ is computed as:

$$
\nabla_\beta \hat{f}(\beta, \beta^{(t)}) = \lambda K_+ \beta - \frac{1}{2} \lambda K_- Y_k W q - \lambda K_- \beta^{(t)},
$$

(16)

where $W = \text{diag} \left[ \exp(-Y_k \beta) \right]$ is a diagonal matrix whose ith diagonal element is $\exp(-y_i K^{(i)} \beta)$, and $q = (q_1, q_2, \cdots, q_n)^T$ by defining

$$
\begin{align}
q_i &= \frac{1}{1 + \exp(-y_i \sum_{j=1}^n \beta_j K_{ij})}, \quad \forall i = 1, 2, \cdots, n.
\end{align}
$$

(17)

To obtain the inexact solution $\beta^{(t+1)} \approx \text{argmin} \ \hat{f}(\beta, \beta^{(t)})$ in the inner loop, the terminate condition is occupied by Eq. (14). Under such bounded error assumption, the rationality of such approximation and the convergence of the CCICP algorithm will be theoretically demonstrated in Section 4.2 with provable guarantees. The detailed procedure of the CCICP algorithm in our IKLR model is summarized in Algorithm 1.

After obtaining the output $\hat{\beta}$ by Algorithm 1, in the test process, we firstly construct the test kernel matrix $K$ associated with the training sample set $X$ and the test sample set $Z$, namely, $K_{ij} = K(x_i, z_j)$, and then compute the classification score of the $i$th test sample $p(z_i)$, which is defined as:

$$
p(z_i) = \frac{\exp \left[ K^{(i)} z_i \right]}{1 + \exp \left[ K^{(i)} z_i \right]}, \quad \forall i = 1, 2, \cdots, s.
$$

(18)

where $K^{(i)}$ represents the $i$th row of the test kernel matrix $K$. If the classification score $p(z_i) > 0.5$, we label $z_i$ with $+1$, otherwise it is assigned to $-1$, which completes a predict progress for a test sample.
Algorithm 1: The CCICP for indefinite kernel logistic regression.

Input: the indefinite kernel matrix $K$ and two positive semi-definite kernel matrices $K_+$ and $K_-$, the label matrix $Y$, and the regularization parameter $\lambda$.

Output: the coefficient vector $\beta$.

1: Set: stopping criterion: $t_{\text{max}} = 15$, the stepsize $\eta = 0.2$, and the decay factor $\tau = 0.5$.
2: Initialize $t = 0$ and $\beta^{(0)}$, and compute $\varepsilon$.
3: Repeat
   4: Obtain $\hat{h}(\beta^{(t)}) = \lambda K_- \beta^{(t)}$.
   5: Obtain the sub-problem $\hat{f}(\beta)$ by Eq. (15);
     // Inner Loop: Solve $\beta^{(t+1)} = \text{argmin} \hat{f}(\beta)$.
   6: Initialize $k = 0$ and $\beta_k^{(t)} := \beta^{(t)}$;
   7: while $\|\nabla \hat{f}(\beta_k^{(t)})\| > \varepsilon\|\beta_k^{(t)}\|$ do
     8: Obtain the gradient $\nabla \hat{f}(\beta_k^{(t)})$ by Eq. (16);
     9: $\beta_{k+1}^{(t)} := \beta_k^{(t)} - r\eta \nabla \hat{f}(\beta_k^{(t)})$;
    10: $k := k + 1$;
   11: end
   12: Output $\beta^{(t+1)} := \beta_k^{(t)}$ that minimizes Eq. (15);
     // Inner Loop completes.
   13: $t := t + 1$;
   14: Until $t = t_{\text{max}} \vee \|\beta^{(t)} - \beta^{(t-1)}\|_{\infty} \leq \varepsilon$;
15: Output the stationary point $\hat{\beta}$ that minimizes Eq. (11).

4.2 Convergence Analysis of the CCICP

With the aforementioned inexact operation, the CCICP algorithm is expected to speed up the optimization process. For the ease of such algorithm in theory, we carefully consider the convergence of CCICP in which an inexact sequence $\{\beta^{(t)}\}_{t=1}^\infty$ generated by Algorithm 1, and then further analyze its convergence rate in our IKLR model.

The key convergence analysis result of the CCICP is summarized by Theorem 4.2, that is, when the error $\varepsilon$ is upper bounded, given a point $\beta^{(0)} \in \mathbb{R}^n$ generates a sequence $\{\beta^{(t)}\}_{t=1}^\infty$, the sequence still converges to a local minimum or a stationary point.

To prove Theorem 4.2, we need the following Lemma 4.1 to aid the proof.

Lemma 4.1. Given a sigmoid function $R(x) = (1 + e^{cx})^{-1}$ where $c \in \{+1, -1\}$, and for two arbitrary variables $x_1, x_2 \in (-\infty, +\infty)$, there exists a bound such that
\[
R(x_1) - R(x_2) \leq \frac{1}{4}|x_1 - x_2|.
\]

Proof. Because $R(x)$ is a differential function, by using Lagrange mean value theorem, there exists at least one point $x \in (\min(x_1, x_2), \max(x_1, x_2))$ such that
\[
|R(x_1) - R(x_2)| = |(x_1 - x_2)R'(x)|,
\]
where the range of $|R'(x)|$ satisfies:
\[
|R'(x)| = \frac{e^{cx}}{(1 + e^{cx})^2} \leq \frac{1}{4}. \tag{21}
\]

Then we can conclude the proof, which is:
\[
|R(x_1) - R(x_2)| \leq \frac{1}{4}|x_1 - x_2|. \tag{22}
\]

Theorem 4.2. The sequence $\{\beta^{(t)}\}_{t=1}^\infty$ with an inexact operation generated by the CCICP still converges to a local minimum or a stationary point if the bound error $\varepsilon$ in Eq. (14) (i.e. $\varepsilon_1$ and $\varepsilon_2$ in Eqs. (27) and (28)) satisfies:
\[
\max \{\varepsilon_1, \varepsilon_2\} < \lambda (\|K_\| - \|K_-\| - \frac{\|K\|^2}{4n}). \tag{23}
\]

Proof. Let $\phi : U \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a point-to-set map, $\beta^{(t+1)} \in \phi(\beta^{(t)})$ such that
\[
\phi(\beta^{(t)}) = \text{argmin} \hat{f}(\beta, \beta^{(t)}), \tag{24}
\]
which generates an inexact sequence $\{\beta^{(t)}\}_{t=1}^\infty$ through the rule $\beta^{(t+1)} \in \phi(\beta^{(t)})$, where $\phi(\beta^{(t)})$ satisfies the bounded error assumption, namely:
\[
\nabla \beta \hat{f}(\beta, \beta^{(t)})|_{\beta = \phi(\beta^{(t)})} = \varepsilon \|\beta^{(t)}\|. \tag{25}
\]

Specifically, the map $\phi$ is said to be global convergent if for any chosen initial point $\beta^{(0)}$, the sequence converges to a point for which a necessary condition of optimality holds. Therefore, the key is to prove that the map $\phi$ is a contraction mapping for two arbitrary points $a, b \in \text{int}(U)$ such that
\[
\|\phi(a) - \phi(b)\| \leq \alpha \|a - b\|, \tag{26}
\]
for a distance metric $\|\cdot\|$, where $\alpha \in (0, 1)$.

Suppose that $\phi(a)$ and $\phi(b)$ satisfy:
\[
\nabla \beta \hat{f}(\beta, a)|_{\beta = \phi(a)} = \varepsilon_1 \|a\|, \tag{27}
\]
\[
\nabla \beta \hat{f}(\beta, b)|_{\beta = \phi(b)} = \varepsilon_2 \|b\|, \tag{28}
\]
where $\varepsilon_1$ and $\varepsilon_2$ correspond to the bounded error, which leads to the inexact sequence $\{\beta^{(t)}\}_{t=1}^\infty$. For simplicity, suppose $\varepsilon_1 \leq \varepsilon_2$, and the subtraction between Eqs. (27) and (28) can be formulated as:
\[
\lambda K_+ \left[\phi(a) - \phi(b)\right] = \lambda K_- (a - b) + \frac{1}{n} KYh + \varepsilon_1 \|a\| - \varepsilon_2 \|b\|, \tag{29}
\]
where $h$ is a $n$-dimensional vector, the ith element of which is defined as:
\[
h_i = \frac{1}{1 + \exp \left(y_i K^{(i)}(a)\right)} - \frac{1}{1 + \exp \left(y_i K^{(i)}(b)\right)}. \tag{30}
\]
By using Lemma 4.1, we have:
\[
|h_i| \leq \frac{1}{4} |K^{(i)}(a) - K^{(i)}(b)|, \forall i = 1, 2, \cdots, n. \tag{31}
\]

It does not imply convergence to a global optimum for all initial values $\beta^{(0)}$.

If $\varepsilon_1 > \varepsilon_2$, we use the subtraction between Eq. (28) and (27).
and then $\|h\|_\infty$ satisfies:

$$
\|h\|_\infty \leq \frac{1}{4} \|K_s\| \|\phi(a) - \phi(b)\|_\infty \leq \frac{1}{4} \|K_s\| \|\phi(a) - \phi(b)\|_\infty
$$

(32)

where $s = \arg\min_{i} |K^{(i)}\phi(a) - K^{(i)}\phi(b)|$, $i = 1, 2, \cdots, n$.

Due to the positivity of $K_+$, Eq. (29) can be reformulated as:

$$
\phi(a) - \phi(b) \leq \frac{1}{n} \|K_+\|_n (\lambda K_+ (a - b) + \frac{1}{n} K Y h + \epsilon_1 \|a\| - \epsilon_2 \|b\|).
$$

(33)

Subsequently, Eq. (33) can be bounded by using $\|\cdot\|_\infty$ (we omit the notation for simplicity), that is:

$$
\|\phi(a) - \phi(b)\| \leq \frac{1}{4} K_s^{-1} |\lambda K_s (a - b) + \frac{1}{n} K Y h + \epsilon_1 \|a\| - \epsilon_2 \|b\||
\leq \frac{1}{4} \|K_s\| \|\phi(a) - \phi(b)\|
$$

(34)

Here we can obtain:

$$
\|\phi(a) - \phi(b)\| \leq \frac{\|K_s\| + \frac{\epsilon_2}{4}}{\|K_s\| + \frac{\epsilon_1}{4}} \|a - b\|.
$$

(35)

Likewise, if $\epsilon_2 < \epsilon_1$, we can also obtain:

$$
\|\phi(b) - \phi(a)\| \leq \frac{\|K_s\| + \frac{\epsilon_1}{4}}{\|K_s\| + \frac{\epsilon_2}{4}} \|b - a\|.
$$

(36)

We reformulate Eqs. (35) and (36) into a uniform framework, namely:

$$
\|\phi(a) - \phi(b)\| \leq \frac{\max(\epsilon_1, \epsilon_2)}{\|K_s\| + \frac{\epsilon_1}{4}} \|a - b\|.
$$

(37)

To guarantee that the map $\phi$ is a contraction mapping, we require

$$
\alpha = \frac{\|K_s\| + \max(\epsilon_1, \epsilon_2)}{\|K_s\| + \frac{\epsilon_1}{4}} < 1.
$$

(38)

After some straightforward algebraic manipulations, $\epsilon_1$ and $\epsilon_2$ can be upper bounded as shown in Eq. (23). Finally, the map $\phi$ served as a contraction mapping is well theoretical demonstrated if the error is upper bounded. By using the fixed point theorem, we can conclude the proof.

4.3 The Convergence Rate of our CCIPC

Here we investigate the convergence rate of the CCIPC in our IKLR model. As [15] has studied the local convergence of the CCCP, they showed that depending on the curvature of $\phi(\beta)$ and $h(\beta)$, CCCP would exhibit either quasi-Newton behavior with fast, typically superlinear convergence or extremely slow, first-order convergence behavior. Assume that the sequence $\{\beta^{(t)}\}_{t=1}^{\infty}$ converges to the fixed point $\hat{\beta}$, namely $\hat{\beta} = \phi(\hat{\beta})$, we can Taylor expand it in the neighborhood of the fixed point $\hat{\beta}$ since the mapping $\phi$ is continuous and differentiable. That is:

$$
\beta^{(t+1)} - \hat{\beta} = M'(\hat{\beta}) (\beta^{(t)} - \hat{\beta}),
$$

(39)

where $M'(\hat{\beta}) = \frac{\partial M}{\partial \beta} |_{\beta = \hat{\beta}}$. It is termed as the convergence matrix which controls the quasi-Newton behavior. Near the local optimum, this matrix is related to the curvature of the convex function $g(\beta)$ and the concave function $-h(\beta)$, which is given by:

$$
M'(\hat{\beta}) = \left[ \frac{\partial^2 h(\beta)}{\partial \beta \beta^\top} |_{\beta = \hat{\beta}} \right]^{-1},
$$

(40)

which can be interpreted as a ratio of concave curvature to convex curvature.

In our CCIPC algorithm, the fixed point $\hat{\beta}$ generated by the sequence $\{\beta^{(t)}\}_{t=1}^{\infty}$ with a bounded error. In this case, it can be also approximated by the Taylor expansion. As a result, we can still analyse the local convergence of the CCIPC in our model as aforementioned. After two Hessian matrices $\nabla^2 h(\beta)$ and $\nabla^2 g(\beta)$ obtained, the convergence matrix is determined by:

$$
M'(\hat{\beta}) = \lambda K_+ \left( \frac{1}{n} K Y h(\hat{\beta}) K + \lambda K_+ \right)^{-1},
$$

(41)

where $H = \text{diag}(q_1(1 - q_1), \cdots, q_n(1 - q_n))$, and $q_i$ is defined in Eq. (17). Given an indefinite kernel matrix $K$, the convergence rate is determined by the ratio of $K_-$ from the concave part and $K_+$ from the convex part. Generally, in indefinite kernel learning, eigenvalues of $K_+$ are often much larger than that of $K_-$. In this case, $K_+$ occupies a dominant while $K_-$ pales in importance. Hence, the CCIPC in our IKLR model will exhibit a quasi-Newton behavior and possess fast, typically superlinear convergence. In our experiments, such condition will be satisfied in real-world dataset and the convergence of the CCIPC will be further demonstrated. Specifically, we must point out that different convex-concave decompositions do not change the final results of our algorithm; while they only change the convergence rate.

5 EXPERIMENTS

In this section, we evaluate our IKLR model on two benchmarks with a collection of multi-modal dataset from multimedia and machine learning areas.

5.1 Experiment Setup

In our experiment, the regularization parameter $\lambda$ is set to 1. For the kernel setting, we choose a truncated $\ell_1$ distance (TL1) indefinite kernel [7] incorporated into our model, which is defined as $K(u, v) = \max\{\tau - \|u - v\|, 0\}$. As discussed in [7], the performance of the TL1 kernel is not too sensitive to the parameter $\tau$, and thus it is fixed to $\tau = 0.7m$ as suggested. The inexact parameter $\epsilon$ is fixed with 1 in CCIPC. In addition, as a representative positive definite kernel, the radial basis function (RBF) kernel is added for comparison, defined as: $K(u, v) = \exp(-\|u - v\|^2/\sigma^2)$. The spread parameter $\sigma$ is chosen by ten-fold cross-validation on the training set. One of these ten subsets is used for validation in turn and the remaining ones for training.
Table 1: Statistics for various datasets with \( n \) training samples represented by a \( m \)-dimensional feature. The notations \( \mu_{\text{max}} \) and \( \mu_{\text{min}} \) denote the maximum and minimum eigenvalues of the TL1 kernel over training samples.

| Dataset      | \( m(\text{feature}) \) | \( n(\text{num}) \) | \( \mu_{\text{min}} \) | \( \mu_{\text{max}} \) |
|--------------|-------------------------|---------------------|------------------------|------------------------|
| monks1       | 6                       | 124                 | -2.094                 | 94.077                 |
| monks2       | 6                       | 169                 | -2.355                 | 131.14                 |
| monks3       | 6                       | 122                 | -1.764                 | 95.376                 |
| parkinsons   | 23                      | 195                 | 0.127                  | 1200.4                 |
| sonar        | 60                      | 208                 | 1.452                  | 3024.6                 |
| SPECT        | 21                      | 80                  | -1.145                 | 353.11                 |
| transfusion  | 4                       | 748                 | -0.336                 | 818.74                 |
| splice       | 60                      | 1000                | -1.325                 | 2885.3                 |

5.2 Results on UCI Dataset

In this section, ten real-world datasets from UCI Machine Learning Repository [2] are used to evaluate the performance of our IKLR model with other seven algorithms. For each dataset normalized to \([0, 1]\), we randomly pick up half of the data for training and the rest for test. Table 1 lists a brief description of these ten datasets including the feature dimension \( m \), the number of training samples \( n \), the minimum and maximum eigenvalues of the training TL1 kernels. It can be observed that the absolute value of the maximum eigenvalue in each dataset is always much larger than that of the minimum one, which means that the CCICP will possess fast in our IKLR model as discussed in Section 4.3.

We compare our IKLR method with other representative state-of-the-art indefinite kernel learning based algorithms including: “Flip”, “Clip”, and “Shift” [22]: three methods directly convert the indefinite kernel matrix generated by TL1 kernel into a positive semi-definite matrix by using the spectrum transformation. Then we take the modified kernel matrix into kernel logistic regression. “SVM(RBF)”: a representative classification method uses SVM with the RBF kernel. “KSVM” [11]: a method transforms TL1 kernel from indefinite matrix generated by TL1 kernel into a positive semi-definite matrix by using the spectrum transformation. Then we take the modified kernel matrix into kernel logistic regression. “IKLR” [25]: a representative classification method uses logistic regression with the RBF kernel just for self-veriﬁcation.

We test above algorithms on these ten datasets, where the procedure is repeated 10 times, and then the average classiﬁcation accuracy and its standard deviation across frames with average pooling operation. As a result, a feature vector created in this way is treated as an input to effectively represent a speech clip. For these speech clips in ten classes, we randomly divide these clips in each class into two non-overlapping training and testing sets which contain almost half of the samples in each class. Learning is performed with a 5-fold cross-validation regime.

5.3 Results on ESC Dataset

Environmental sound classiﬁcation is one of the obstacles in research activities. We accomplish this auditory recognition task by our IKLR model on ESC-10 dataset [14]. The ESC-10 dataset is a selection of 10 classes that represents three general groups of sounds, namely transient sounds (sneezing, dog barking, clock ticking), sounds events with strong harmonic content (crying baby, crowing rooster), and sound event with structured noise (rain, sea waves, fire crackling, helicopter, chainsaw).

In our experiment, we extract a ubiquitous feature in speech processing, namely mel-frequency cepstral coefficients (MFCC), where each speech clip is divided into numerous frames. For each frame, a 12-dimensional MFCCs is extracted to represent the current frame in each clip with default settings\(^5\). By doing so, a speech clip is represented by a MFCC matrix where each row of this matrix is a 12-dimensional MFCCs for a frame. Then we compute their means and standard deviations across frames with average pooling operation. As a result, a feature vector created in this way is treated as an input to effectively represent a speech clip. For these speech clips in ten classes, we randomly divide these clips in each class into two non-overlapping training and testing sets which contain almost half of the samples in each class. Learning is performed with a 5-fold cross-validation regime.

\(^5\)http://dx.doi.org/10.5281/zenodo.12714
Table 2: Test classification accuracy of (mean±std. deviation) of each compared algorithm on UCI datasets. The best performance is highlighted in bold.

| Dataset     | KLR(RBF) [25] | Flip | Clip | Shift | KSVM [11]   | CCICP       |
|-------------|---------------|------|------|-------|--------------|-------------|
| monks1      | 0.668±0.052   | 0.695±0.075 | 0.648±0.070   | 0.685±0.063   | 0.586±0.102 | **0.765±0.065** |
| monks2      | 0.662±0.071   | 0.498±0.110   | 0.506±0.116   | 0.489±0.092   | 0.626±0.037 | **0.669±0.093** |
| monks3      | 0.779±0.073   | 0.723±0.090   | 0.805±0.021   | **0.870±0.036** | 0.640±0.083 | 0.830±0.072   |
| parkinsons  | **1.000±0.000** | 0.999±0.010   | 0.999±0.003   | 0.998±0.007   | 0.945±0.039 | **1.000±0.000** |
| sonar       | 0.789±0.022   | 0.546±0.045   | 0.539±0.042   | 0.504±0.054   | 0.608±0.072 | **0.794±0.060** |
| SPECT       | 0.737±0.092   | 0.652±0.026   | 0.706±0.022   | 0.667±0.034   | **0.893±0.024** | 0.764±0.059   |
| splice      | 0.642±0.093   | 0.513±0.017   | 0.619±0.057   | 0.604±0.033   | 0.515±0.029 | **0.785±0.050** |
| transfusion | 0.741±0.048   | 0.734±0.095   | 0.717±0.020   | 0.736±0.038   | **0.762±0.006** | 0.726±0.129   |

Table 3: Results of CCCP and CCICP on several large-scale data sets.

| Dataset               | Method | CCCP | CCICP | CCCP | CCICP | CCCP | CCICP |
|-----------------------|--------|------|-------|------|-------|------|-------|
|                       | Accuracy | 0.769±0.042 | 0.725±0.042 | 0.962±0.003 | 0.955±0.003 | 0.624±0.080 | 0.699±0.051 |
|                       | Training time | 1717.1 | 848.8 | 1314.3 | 47.2 | 305.2 | 8.1 |
|                       | Test time | 0.1237 | 0.1304 | 0.0020 | 0.0028 | 0.0008 | 0.0064 |

| $\mu_{\text{min}}$ | $\mu_{\text{max}}$ | SVM(RBF) | KSVM | IKLR |
|-------------------|-------------------|----------|------|------|
| −0.068            | 722.45            | 64.3%    | 68.1% | 75.7% |

Table 4: Comparison of average classification accuracy (%) of different algorithms where $\mu_{\text{max}}$ and $\mu_{\text{min}}$ denote the maximum and minimum eigenvalues of the TL1 kernel over training samples.

Here we choose three representative classifiers including SVM with RBF, KSVM [11] with TL1 kernel, and our IKLR algorithm to evaluate the classification performance. Tab. 4 reports the average test accuracy (%) across above three algorithms. We can see that the average classification accuracy ranges from 64.3% for SVM with the RBF kernel to 75.7% for our IKLR method, with KSVM with the middle (68.1%). This result reinforces to demonstrate the effectiveness of our IKLR algorithm with the TL1 indefinite kernel.

5.4 Algorithm Convergence

The experiments about the convergence of CCICP algorithm are conducted on the monks1l sequence as shown in Fig. 1. The CCICP algorithm only takes 5 iterations to converge on the monks1l data set, while the conventional CCCP algorithm converges with 16 iterations. Therefore, such inexact scheme makes the proposed IKLR model much more efficient.

6 CONCLUSION

This paper introduced the IKLR model to consider the indefinite kernel learning in logistic regression algorithm. The proposed CCICP algorithm effectively solves such non-convex problem by decomposition methods, and adopts an inexact scheme with early stopping the sub-problem to decrease the computational complexity. The convergence of our algorithm has been demonstrated with theoretical guarantees and experimental validation. Specifically, the CCICP exhibits quasi-Newton behavior or typically superlinear convergence because the convex part in our IKLR model dominates the concave part. Extensive comparative experiments from multi-modal datasets validate the superiority of the proposed IKLR model to other algorithms with positive definite/indefinite kernels. Besides, the results also enlighten us to design a proper indefinite kernel and does not limit to a positive definite kernel.

REFERENCES

[1] FranCois Bertrand Akoa. 2008. Combining DC Algorithms (DCAs) and Decomposition Techniques for the Training of Nonpositive-Semidefinite Kernels. *IEEE Transactions on Neural Networks*, 19, 11 (2008), 27:1–27:27. Issue 3. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

CCICP exhibits quasi-Newton behavior or typically superlinear convergence because the convex part in our IKLR model dominates the concave part. Extensive comparative experiments from multi-modal datasets validate the superiority of the proposed IKLR model to other algorithms with positive definite/indefinite kernels. Besides, the results also enlighten us to design a proper indefinite kernel and does not limit to a positive definite kernel.
[5] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. 2000. Regularization Networks and Support Vector Machines. Advances in Computational Mathematics 13, 1 (2000), 1–50.

[6] Bernard Haasdonk. 2005. Feature Space Interpretation of SVMs with Indefinite Kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 4 (2005), 482–492.

[7] Xiaolin Huang, Johan AK Suykens, Shuning Wang, Joachim Hornegger, and Andreas Maier. 2017. Classification With Truncated $\ell_1$ Distance Kernel. IEEE Transactions on Neural Networks and Learning Systems (2017).

[8] Tommi S. Jaakkola and David Haussler. 1998. Probabilistic Kernel Regression Models. In Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics.

[9] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2005. Exact indexing of dynamic time warping. Knowledge and Information Systems 7, 3 (2005), 358–386.

[10] Chih Jen Lin, Ruby C Weng, and S. Sathiya Keerthi. 2007. Trust region Newton methods for large-scale logistic regression. In Proceedings of the ACM twenty-fourth International Conference on Machine Learning. 561–568.

[11] Guelle Loosli, Stephane Canu, and Soon Ong. Cheng. 2016. Learning SVM in Krein Spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence 38, 6 (2016), 1204–1216.

[12] Cheng Soon Ong, Xavier Mary, and Alexander J. Smola. Learning with non-positive kernels. In Proceedings of the twenty-first International Conference on Machine Learning. 81–89.

[13] Elzbieta Pekalska and Bernard Haasdonk. 2009. Kernel Discriminant Analysis for Positive Definite and Indefinite Kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 6 (2009), 1017–1032.

[14] Karol J. Pszczał. 2015. ESC: Dataset for Environmental Sound Classification. In Proceedings of the ACM International Conference on Multimedia. 1015–1018.

[15] Rudi Salakhutdinov, Sam Roweis, and Zoubin Ghahramani. 2012. On the Convergence of Bound Optimization Algorithms. In Proceedings of the nineteenth Conference on Uncertainty in Artificial Intelligence. 509–516.

[16] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. 2000. A Generalized Representer Theorem. In Proceedings of the Conference on Computational Learning Theory. 416–426.

[17] Bernhard Schölkopf and Alexander J Smola. 2003. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press.

[18] Rajiv Ratn Shah, Yi Yu, and Roger Zimmermann. 2014. ADVISOR: Personalized Video Soundtrack Recommendation by Late Fusion with Heuristic Rankings. In Proceedings of the ACM International Conference on Multimedia. 607–616.

[19] Bharath K. Sriperumbudur and Gert R. G. Lanckriet. 2009. On the Convergence of the Concave-Convex Procedure: In Proceedings of the International Conference on Neural Information Processing Systems. 1759–1767.

[20] Vladimir N. Vapnik. 2000. The Nature of Statistical Learning Theory. Springer.

[21] Xinxi Wang and Ye Wang. 2014. Improving Content-based and Hybrid Music Recommendation using Deep Learning. In Proceedings of the ACM International Conference on Multimedia. 627–636.

[22] Gang Wu, Edward Y. Chang, and Zhihua Zhang. 2005. An Analysis of Transformation on Non-positive Semidefinite Similarity Matrix for Kernel Machines. In Proceedings of the ACM twenty-second International Conference on Machine Learning.

[23] Haiming Xu, Hui Xue, Xiaohong Chen, and Yunyun Wang. 2017. Solving Indefinite Kernel Support Vector Machine with Difference of Convex Functions Programming. In AAAI. 1610–1616.

[24] Alan L. Yuille and Anand Rangarajan. 2003. The Concave-Convex Procedure. Neural Computation 15, 4 (2003), 915–936.

[25] Ji Zhu and Trevor Hastie. 2002. Kernel Logistic Regression and the Import Vector Machine. Journal of Computational and Graphical Statistics 14, 1 (2002), 185–205.