Multiple Kernel SVM Based on Two-Stage Learning

XINGRUI GONG\(^1\), BIN ZOU\(^1\), YUZE DUAN\(^1\), JIE XU\(^2\), QINGXIN LUO\(^1\), AND YAN YANG\(^3\)

\(^1\)Hubei Key Laboratory of Applied Mathematics, Faculty of Mathematics and Statistics, Hubei University, Wuhan 430062, China
\(^2\)Faculty of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China
\(^3\)Faculty of Computer Science, Wuhan Qingchuan University, Wuhan 430062, China

Corresponding authors: Bin Zou (zoubin0502@gmail.com) and Jie Xu (frangipani@hubu.edu.cn)

This work was supported in part by NSFC Project under Grant 61772011, and in part by the Open Project Foundation of Intelligent Information Processing Key Laboratory of Shanxi Province under Grant CICIP2018002.

ABSTRACT In this paper we introduce the idea of two-stage learning for multiple kernel SVM (MKSVM) and present a new MKSVM algorithm based on two-stage learning (MKSVM-TSL). The first stage is the pre-learning and its aim is to obtain the information of data such that the “important” samples for classification can be generated in the formal learning stage and these samples are uniformly ergodic Markov chain (u.e.M.c.). To study comprehensively the proposed MKSVM-TSL algorithm, we estimate the generalization bound of MKSVM based on u.e.M.c. samples and obtain its fast learning rate. And in order to show the performance of the proposed MKSVM-TSL algorithm for better, we also perform the numerical experiments on various publicly available datasets. From the experimental results, we can find that compared to three classical multiple kernel learning (MKL) algorithms, the proposed MKSVM-TSL algorithm has better performance in three aspects of the total time of sampling and training, the accuracy and the sparsity of classifiers, respectively.

INDEX TERMS Two-stage learning, multiple kernel SVM, uniformly ergodic Markov chain, learning rate.

I. INTRODUCTION

Support Vector Machine (SVM) is one of the most widely applied machine learning methods for pattern recognition problems [1]. Besides its good theoretical property in learning rate and consistency [2]–[4], SVM can solve nonlinear classification problem by introducing kernel functions, which is called the kernel method. Although the kernel method has offered plenty of opportunities to solve some complicated problems, it brought lots of interdisciplinary challenges in statistics, optimization theory and applications [5]. And choosing the optimal kernel from a set of candidates and its parameters is a central choice, which usually must be made by a human user using the prior knowledge of data. In other words, the classical kernel-based classifiers are based on a single kernel, and in practice, an ideal classifier is usually based on a combination of multiple kernels.

Therefore, multiple kernel learning (MKL) has been widely concerned and studied. For example, Pavlidis et al. [6] researched gene functional classification from heterogeneous data, which was based on SVM with an unweighted sum of heterogeneous kernels. Ben-Hur and Noble [7] presented unweighted pairwise kernels for improving the performance of predicting protein-protein interactions. Lanckriet et al. [8] introduced the method which learns the kernel matrix with semi-definite programming (SDP) to search the optimal of unrestricted kernel combination weights and showed that MKL is comparable with the best soft margin SVM for radial basis function (RBF) kernel. Although SDP with the unrestricted kernel combination weights and nonnegative kernel combination weights generated lots of linear combination algorithms for MKL, the computational complexity is high when the size of training samples is bigger. Then the method of convex combination of multiple kernels appeared in vast literatures in order to reduce the algorithmic complexity of MKL, such as Argyriou et al. [9], [10] proposed the method of choosing appropriate kernel from the convex hull of basic kernels, which are continuously parameterized by a compact infinite set simultaneously, where the chosen kernel can minimize the convex regularization functional. Sonnenburg et al. [11] considered the semi-infinite linear programs (SILP) to obtain the optimal convex combination of kernels by iteratively updating the kernel weights and the vector of dual variables.
Yu et al. [12] studied the semi-infinite programming (SIP) and proved that SIP can be applied well to biomedical data fusion. The research results in [12] showed that least squares SVM (LSSSVM) algorithm with SIP has comparable performance and even more efficiently than the conventional MKL method. Unfortunately, this method of convex combination of multiple kernels needs to solve single kernel SVM for many times. To get out of the loop, the method of obtaining the kernel combination weights by using kernel similarity measures was introduced by Kandola et al. in [13].

Igel et al. [14] studied the gradient-based optimization of kernel-target alignment for sequence kernels and applied it to bacterial gene start detection. Cristianini et al. [15] considered the method of maximizing the kernel alignment. In recent years, the MKL method of iteratively updating kernel weights to obtain the optimal kernel combination has been successfully applied in many fields. For example, Chavaltada et al. [16] proposed a method of automated product categorisation by using MKL to improve feature combination in e-commerce. Rahimi and Gönen [17] discriminated early-and late-stage cancers using multiple kernel learning on gene sets, and the performance of their proposed algorithm outperformed other algorithms in terms of predictive power. Wilson et al. [18] applied MKL to genomic data mining and prediction. Lauriola et al. [19] enhanced deep neural networks via MKL, and the method proposed in [19] gave an effective way to design the output computation in deep networks.

However, all the MKL methods mentioned above are still to solve the single kernel SVM for one or more times and recall that the complexity of a single kernel SVM is about $O(n^q)$ [20], where $n$ is the size of training samples and $q \in [2,3]$. Hence, the algorithmic complexity of MKL combining multiple kernels is very high for the big size of training samples. This implies that although MKL method has good learning performance, MKL method is usually very time-consuming and even difficult to implement when the scale of training samples is larger. Then a problem is posed: How to reduce the algorithmic complexity of MKL method (i.e., multiple kernel SVM) and keep its better classification accuracy at the same time?

To solve this problem, we present the idea of two-stage learning for multiple kernel SVM (MKSVM) algorithm in this paper. The two-stage learning consists of the pre-learning and the formal learning. The reasons of introducing two-stage learning are as follows: First, with the advent of the high-tech era, the capacity of data is growing rapidly, and the scale of data is getting larger and larger, so the classical MKL methods usually time-consuming and even difficult to implement in the case of a large training sample size. Second, the larger the amount of data, the lower the value density of data will be, which means that there are many noise samples in big data. A large number of machine learning experiments indicate that the noise samples will not only lead to the increase of storage space, but also affect the accuracy and efficiency of learning. In addition, according to statistical learning theory [1], we know that the most “important” samples for classification problem are the samples close to the interface of two classes. Thus the aim of two-stage learning is to generate the “important” samples from the given training set in the stage of formal learning by making use of the information obtained in the stage of pre-learning. To our knowledge, this is the first algorithm of MKL method. For these Markov samples are drawn ergodically in the stage of formal learning, we consider uniformly ergodic Markov chain (u.e.M.c.) in this paper.

In this paper, we first introduce a new MKSVM algorithm based on two-stage learning (MKSVM-TSL) and then we study the generalization bound of MKSVM algorithm based on u.e.M.c. samples and obtain its fast learning rate. We also compare the proposed MKSVM-TSL with three classical MKL algorithms by numerical studies of various publicly available datasets. We highlight some contributions of this paper as follows.

- The generalization bound of MKSVM based on u.e.M.c. samples is obtained and the optimal learning rate is established.
- A new MKSVM algorithm, MKSVM-TSL is proposed, and its competitive performance is proved by numerical experiment.

The rest of our paper states as follows. Section II formulates the classical multiple kernel SVM (MKSVSVM) algorithm. Section III introduces a new MKSVM algorithm based on two-stage learning and analyzes its algorithm complexity. Section IV presents the main theoretical analysis on the learning rate and generalization bound of MKSVM algorithm with u.e.M.c. samples. Section V gives the experimental results of comparing MKSVM-TSL with three classical MKL algorithms. Finally, we conclude this paper in Section VI.

II. MULTIPLE KERNEL SVM FORMULATION

Considering a compact metric space $(\mathcal{X},d)$, let $\mathcal{Y} \in \{-1,1\}$ and $\rho$ be an unknown probability distribution on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where we take random variable $(X,Y)$ form $\mathcal{Z}$. The goal of learning is to find a good prediction rule $h : \mathcal{X} \rightarrow \mathcal{Y}$ which will assign values to objects such that if new objects are given, the function $h$ will forecast them correctly. The misclassification error of $h$ is defined by the probability of the event \( h(X) \neq Y \), i.e., \( R(h) = P(h(X) \neq Y) \). In this paper, we assume that $h$ depends on a Reproducing Kernel Hilbert Space (RKHS). Thus MKSVM is given by $M$ different RKHS $\mathcal{H}_p$ $(\mathcal{H}_p)^M_{p=1}$, each of which associated with a Mercer kernel $K_p$ [21]. Let $K_p : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be continuous, symmetric and positive semidefinite, i.e., for any finite set of distinct points $\{x_i\}_{i=1}^m \subset \mathcal{X}$, the matrix $(K_p(x_i, x_j))_{i,j=1}^m$ is positive semidefinite. Such a function is called a Mercer kernel. The RKHS $\mathcal{H}_p$ associated with the kernel $K_p$ is defined to be the closure of the linear span of the set of functions $\{K_p,x = K_p(x,:), x \in \mathcal{X}\}$ with the inner product $(\cdot, \cdot)_p = \langle \cdot, \cdot \rangle_{K_p}$ satisfying $(K_p(x_i,x_j), K_p(x_i,x_j))_{K_p} = K_p(x_i, x_j)$, $(\sum_i \alpha_i K_p(x_i, \cdot), \sum_j \beta_j K_p(\cdot, x_j))_{K_p} = \sum_{i,j} \alpha_i \beta_j K_p(x_i, x_j)$.
Let $C(\mathcal{X})$ be the space of continuous functions on $\mathcal{X}$ equipped with the norm $\|f_p\|_p = \sup_{x \in \mathcal{X}} |f_p(x)|$. Let $\kappa = \sup_{x \in \mathcal{X}} \sqrt{K_p(x, x)}$ for $1 \leq p \leq M$, then the reproducing property tells us that $\|f_p\|_p \leq \kappa \|f_p\|_{K_p}, \forall f_p \in \mathcal{H}_p$. The original optimization problem of MKSVM involving a set of sample $z = \{z_i = (x_i, y_i)\}_{i=1}^n$ can be stated as follows:

$$f_z = \arg \min_{f_p \in \mathcal{H}_p} \frac{1}{2} \sum_{p=1}^M \frac{1}{d_p} \|f_p\|_{K_p}^2 + C \sum_{i=1}^n \xi_i$$

$$\text{s.t. } \sum_{p=1}^M y_i f_p(x_i) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad 1 \leq i \leq n,$$

$$\sum_{p=1}^M d_p = 1, \quad d_p \geq 0,$$ (1)

where $C$ is a positive trade-off parameter and $\xi_i$ are slack variables [22]. In this paper, we set $d_p = 1/M$.

The MKSVM algorithm (1) can be rewritten to be a regularized algorithm form: Define loss function $l(f, z) = 0$ for $f(x)y > 1$ and $l(f, z) = 1 - f(x)y$ for $f(x)y \leq 1$. The corresponding expected risk and the empirical risk are defined as $E(f) = E[l(f, z)], \quad E_k(f) = \frac{1}{n} \sum_{i=1}^n l(f, z_i)$. Then for the RKHS $\mathcal{H}_K = \mathcal{H}_{K_1} \times \cdots \times \mathcal{H}_{K_M}$ and $f = \frac{1}{M} \sum_{p=1}^M f_p$, $f_p \in \mathcal{H}_p$, the algorithm (1) can be rewritten as:

$$f_z = \arg \min_{f \in \mathcal{H}_K} E_k(f) + \lambda \|f\|_K^2,$$ (2)

where $\|f\|_K^2 = \frac{1}{M} \sum_{p=1}^M \|f_p\|_{K_p}^2$ is regularization term and $\lambda$ is regularization parameter [23]–[24]. The corresponding MKSVM classifier is defined as $\text{sign}(f_z)$, where the sign function is defined as $\text{sign}(f(x)) = 1$ for $f(x) \geq 0$ and $\text{sign}(f(x)) = -1$ for $f(x) < 0$.

### III. ALGORITHM AND COMPLEXITY ANALYSIS

In this section, we present the MKSVM algorithm based on two-stage learning (MKSVM-TSL) and then we analyze the complexity of the MKSVM-TSL algorithm.

#### A. MKSVM-TSL ALGORITHM

Inspired by the works in [23]–[26], the MKSVM-TSL algorithm can be stated as follows.

Let $n$ be the size of training samples, which are randomly drawn from the original training set $D_{train}$. In Algorithm 1, $N_1, KN_2$ are the size of pre-learning and the size of formal learning, where $k$ is a positive integer. For simplicity, all the experimental results presented in this paper are based on $N_1 = N_2 = n/(k+1)$, $q = 1.2$ and $n_1 = 30$, where $k = 1, 3$ and the more detailed descriptions of $q$ and $n_1$ can be found in [27].

Compared Algorithm 1 with the corresponding algorithm introduced in [28], we can find that although Algorithm 1 is similar to the algorithm presented in [28], the differences are obvious: First, Algorithm 1 is a MKSVM algorithm while the algorithm presented in [28] is a SVM algorithm with a single kernel, which implies that Algorithm 1 extended the algorithm presented in [28] from a single kernel to the case of multiple kernels. In other words, the algorithm presented in [28] is a special case of Algorithm 1 proposed in this paper. Second, the total number of training samples of algorithm introduced in [28] is $2n$, which implies that compared to the classical SVM with a single kernel based on $n$ randomly training samples, the algorithm presented in [28] is time-consuming. While in Algorithm 1, the total number of training samples is equal to that of the classical MKSVM algorithm as $N_1$ and $N_2$ satisfy $N_1 + kN_2 = n$. This implies that Algorithm 1 improved the algorithm introduced in [28].

#### B. COMPLEXITY ANALYSIS

From Algorithm 1, we know that $N_1 = N_2 = n/(k+1)$, and the complexity of a single kernel SVM is about $O(n^2)$ [20], where $n$ is the size of training samples and $q \in [2, 3]$. In this paper, we choose the mean weights as the kernel weights of MKL SVM. Therefore, the complexity of MKSVM algorithm is about $O(n^2)$. But in Algorithm 1, we divided $n$ training samples into $k + 1$ pieces, thus, the complexity of Algorithm 1 is about $O\left((k+1)(n/(k+1))^q\right)$. If we assume

| Algorithm 1 MKSVM-TSL |
|-----------------------|
| **Input:** $D_{train}, n_1, N_1, N_2, k, q$ |
| **Output:** $f_k$ |
| **(Pre-learning)** Draw randomly $N_1$ samples $S_r = \{z_i\}_{i=1}^{N_1}$ from $D_{train}$, then the preliminary learning model $f_0$ can be obtained by MKSVM algorithm (2) with these samples $S_r$ |
| **(Formal learning)** Take randomly a sample $z$ from $D_{train}$ and let it be $z_a, n_2 \rightarrow 0$. for $t = 1, 2, \cdots, k$ do |
| **while** $i < N_2$ do |
| Draw randomly a sample $z_i$ from $D_{train}$, $p \leftarrow e^{-l(f_0, z)} / e^{-l(f_0, z_i)}$ |
| if $p \geq 1$ then |
| $z_a \leftarrow z_i, S_{mar} \leftarrow z_a, i \leftarrow i + 1, n_2 \rightarrow 0$ |
| end |
| if $p < 1$ then |
| $p \leftarrow qp$ |
| $z_a \leftarrow z_i, S_{mar} \leftarrow z_a, i \leftarrow i + 1, n_2 \rightarrow 0$ |
| end |
| end |
| Train $S_{mar} = \{z_i\}_{i=1}^{N_2}$ by MKSVM algorithm (2) and obtain $f_i, f_0 \leftarrow f_i$ |
| end |
| Generate Markov chain $S_{mar} = \{z_i\}_{i=1}^{N_2}$, train $S_{mar}$ by MKSVM algorithm (2) and obtain $f_{k}$ |

The complexity of Algorithm 1 is $O\left((k+1)(n/(k+1))^q\right)$. If we assume $q = 1.2$, then the complexity of Algorithm 1 is $O(n^2)$.
(k + 1) ≈ n^γ for any γ > 0, it is obvious that the complexity of our proposed MKSVM-TSL algorithm in this paper is lower than that of the classical MKSVM.

IV. THEORETICAL ANALYSIS OF MKSVM-TSL

In this section we estimate the generalization bound of MKSVM-TSL and establish its learning rate.

By the definitions of p, p₁ in Algorithm 1, we can find that p, p₁ are positive. In addition, the size n of training samples is finite. By the theory of Markov chain [29], we can conclude that the sample sequence Sₘₐₓ = {z₁, ⋯, zᵣ} generated in Algorithm 1 is uniformly ergodic Markov chains (u.e.M.c.). Then we present the definition of u.e.M.c. as follows: Let (A, S) be a measurable space, define Pₘᵣ(A|ξₖ) as a set of transition probability measures where A ∈ S, ξₖ ∈ A. Then a sequence of random variables {ξₖ}ₖ≥₁ is a Markov chain, if there holds Pₘᵣ(A|ξₖ) := P(ζₘᵣ+i ∈ A|ζₘᵣ,j < i, ζₘᵣ,j = ξₖ). Pₘᵣ(A|ξₖ) denotes the probability that the state ζₘᵣ+i will belong to the set A after m time steps, starting from the initial state ζₘᵣ,j at time i. The fact that the transition probability Pₘᵣ(A|ξₖ) does not depend on the values of ζₘᵣ,j prior to time i is the Markov property, that is Pₘᵣ(A|ξₖ) = P(ζₘᵣ+i ∈ A|ζₘᵣ,j = ξₖ). This is expressed in words as “given the present state, the future and past states are independent”. For two probabilities v₁, v₂ on the measure space (A, S), the total variation distance between the measures v₁, v₂ is defined as ||v₁ − v₂||ₜᵥ = supₐ∈S |v₁(A) − v₂(A)|. Thus the definition of u.e.M.c. can be stated as follows [30]-[32].

Definition 1: [33] Let Φ(−) be the stationary distribution of {ξₖ}ₖ≥₁, {ξₖ}ₖ≥₁ is said to be uniformly ergodic Markov chain for some 0 < α < ∞ and 0 < γ < 1, if there holds

||P(x) − Φ(−)||ₜᵥ ≤ αγᵗ, ∀t ≥ 1, τ ∈ N.

To estimate the generalization bounds of MKSVM-TSL, we should estimate the excess misclassification error R(sign(f₀) − R(f₀). Here f₀ is the Bayes rule, which is defined as f₀ = sign(f₀), f₀ is the regression function of ρ, i.e., f₀ = ʃₓ ydρ(y|x), x ∈ X. Zhang [2] established the following relation between excess misclassification error R(sign(f₀) − R(f₀)) and excess generalization error E(f₀) − E(f₀) for convex loss, that is,

R(sign(f₀)) − R(f₀) ≤ E(f₀) − E(f₀). (3)

To estimate the right hand side E(f₀) − E(f₀) of inequality (3), we also present the following some definitions and assumptions.

Definition 2: [34] For any ε > 0 and F is a subset of metric space. Let N(F, ε) is the covering number of the function set F which is defined as for any ℓ ∈ N, there exist the minimal ℓ disks in F with radius ε covering F. For some given R ∈ R, we define Bₙ = {f ∈ ℋ₉ : ||f||₉ ≤ R} which is in C(X) as a class of functions. Define N(ε) = N(B₁, ε) as the covering number of B₁, where ε > 0.

Assumption 1: [35] We say that RKHS has polynomial complexity exponent s > 0 if ln N(B₁, ε) ≤ Cₛ(1/ε)ˡ, ∀ɛ > 0.

The extensively study of the covering number N(ε) can be found in [36]-[37]. Assumption 1 has been proved by Zhou [34]P for the case that if the kernel is Cᵐₙ/ₙ on a subset X of Rⁿ. And the C∞ kernel is also valid for any s > 0.

Definition 3: [38] On the space of measurable functions f : X → R, we define the projection operator π as

π(f(x)) = {−T, f(x) < −T, f(x), if f(x) ∈ [−T, T], T, if f(x) > T.

Definition 4: [38] Denote fᵢ = arg min f ∈ ℋₙ sup R(∥f∥₉) and the regularization error D(λ) as D(λ) := E(fᵢ) − E(f₀) + λ∥fᵢ∥₉. We have that for some 0 < β ≤ 1 and a constant C₀ > 0, D(λ) ≤ C₀βλᵢ for any λ > k.

Assumption 2: [24] For each {Hₙ}, Hₙ is separable with respect to the norm RKHS and sup₁<λ≤C₁∥Kᵢₙ(x, y) ≤ 1.

Now our main results on the generalization bound of MKSVM-TSL can be stated as follows.

Theorem 1: Suppose ζ = [ζᵢ = (xᵢ, yᵢ)]ᵢₙ₌₁ are u.e.M.c. samples. We have that for any η ∈ (0, 1), R ≥ T, the learning rate holds true with probability at least 1 − η provided that n ≥ 224∥Γ₀∥² ln(1/η)₁+1/(C₁λᵢ),

E(π(fᵢ)) − E(π(f₀)) ≤ 2R(224∥C₁∥Γ₀₀² 1ₙ)n + 3D(λ) + 56√D(λ)/λ + T ln(2/η)∥Γ₀₀∥²/n.

The proof of Theorem 1 is given in Appendix A. By Theorem 1 and inequality (3), we also establish the following learning rate of MKSVM-TSL.

Theorem 2: Under the same assumptions of Theorem 1, we have that for any 0 < η < 1, with probability at least 1 − η, the following inequality holds

R(π(fᵢ)) − R(π(f₀)) ≤ C(1/n)², where C is a constant independent of n.

The proof of Theorem 2 is given in Appendix A. In order to understand Theorem 2 for better, we present the following remarks.

Remark 1: (i) From Theorem 2, we have that when β = 1 and s tend to 0, R(π(f₀)) − R(f₀) is arbitrarily close to O(n⁻¹), which is same as that obtained in [23], [24] and [39] for the case of randomly independent samples. This means that the learning rate obtained in Theorem 2 is optimal for the u.e.M.c. samples. In other words, Theorem 2 extended the previously known works on MKSVM algorithm (2) from randomly independent samples [23], [24], [39] to u.e.M.c. samples.

(ii) Compared Theorem 2 with the corresponding result obtained in [28], we can find that Theorem 2 also extended.
the corresponding result obtained in [28] from a single kernel to the case of multiple kernels. To our knowledge, Theorem 2 is the first work on this topic.

V. NUMERICAL STUDIES

In order to show the performance of MKSVM-TSL, we compare the proposed MKSVM-TSL algorithm with three MKL algorithms, the mean weighted MKSVM based on randomly independent samples [25], the multiple kernel least squares SVM (MKLSSVM) with semi-infinite programming (SIP) based on randomly independent samples [12] and the ratio weighted MKSVM (Rat-MKSVM) based on randomly independent samples [40].

A. DATASETS AND PARAMETERS SELECTION

Our experiments are based on 9 publicly datasets: Image, Census (Census-income), Seismic, Connect4, Isbi4 and TV-News,1 Cod-rnd, A9a and W8a.2 In Table 1, we state the information of these datasets. All experiments were run on Intel 2.80GHz E5-1603 v4 CPU with MATLAB 2018a.

All the experiments of this paper are based on the following 9 kernels, a linear kernel \( K(a, b) = a^\prime b \), three polynomial kernels \( K(a, b) = (1 + a^\prime b)^d \) with \( d \in \{2, 3, 4\} \) and five RBF kernels \( K(a, b) = \exp(-\|a - b\|^2/2\sigma) \), where \( \sigma \) is chosen from the set \( \{0.25, 0.5, 1, 2, 4\} \) and \( a^\prime \) is the transpose of \( a \). The parameter \( \lambda \) of MKSVM algorithm (2) is chosen by 5-fold cross-validation. Besides, all the datasets are normalized before training and testing.

B. EXPERIMENTAL RESULTS

We compare Algorithm 1 with three MKL algorithms, the classical MKSVM, MKLSSVM and Rat-MKSVM in the following three aspects, the accuracy, the total sampling and training time and the sparsity of classifiers. Our experimental procedure can be simply stated as follows: (i) We take randomly \( n \) training samples from the original training set \( D_{\text{train}} \). Then we obtain the corresponding learning model by the three MKL algorithms with these \( n \) training samples, respectively, and we test them on the test samples \( D_{\text{test}} \).

(ii) For Algorithm 1, we first obtain a learning model \( f_z \) according to Algorithm 1, and then we similarly test it on the same test samples \( D_{\text{test}} \).

(iii) The procedures (i) and (ii) are repeated for 50 times, and we record the corresponding average accuracy, the total sampling and training time, the average number of support vectors of each algorithm, respectively.

### TABLE 2. Average accuracy (%) for \( n = 8000 \).

| Dataset | MK | MKLS | R-MK | MK-TSL-1 | MK-TSL-3 |
|---------|----|------|------|----------|----------|
| Cod-rnd | 93.40±0.58 | 85.70±1.20 | 93.51±0.85 | 94.87±0.34 | 94.78±0.26 |
| Image   | 85.12±0.23 | 82.85±0.27 | 85.03±0.24 | 85.73±0.20 | 85.53±0.27 |
| Census  | 94.07±0.12 | 81.16±1.20 | 93.95±0.13 | 94.53±0.08 | 94.20±0.14 |
| Seismic | 81.93±0.41 | 59.86±3.77 | 81.80±0.40 | 82.79±0.31 | 81.97±0.34 |
| A9a     | 79.49±0.40 | 76.92±0.43 | 79.84±0.52 | 81.18±0.26 | 82.03±0.44 |
| Connect4| 75.35±0.49 | 68.18±0.46 | 75.20±0.45 | 76.47±0.43 | 76.11±0.65 |
| W8a     | 94.28±1.34 | 85.44±3.30 | 95.03±1.34 | 97.10±0.66 | 97.34±0.28 |
| Isbi4   | 83.06±0.26 | 80.25±0.33 | 83.08±0.24 | 83.44±0.21 | 83.09±0.32 |
| TV-News | 87.12±0.31 | 75.99±4.58 | 87.00±0.29 | 87.76±0.26 | 87.27±0.58 |

### TABLE 3. Average accuracy (%) for \( n = 10000 \).

| Dataset | MK | MKLS | R-MK | MK-TSL-1 | MK-TSL-3 |
|---------|----|------|------|----------|----------|
| Cod-rnd | 94.21±0.56 | 82.27±4.10 | 94.03±0.65 | 95.22±0.22 | 95.05±0.23 |
| Image   | 85.25±0.24 | 83.20±0.20 | 85.06±0.30 | 85.94±0.23 | 85.84±0.24 |
| Census  | 94.20±0.11 | 81.06±0.64 | 94.10±0.09 | 94.41±0.07 | 94.30±0.13 |
| Seismic | 81.98±0.36 | 60.65±2.92 | 81.57±0.48 | 82.94±0.21 | 82.23±0.45 |
| A9a     | 79.54±0.34 | 76.12±1.47 | 80.08±0.59 | 81.21±0.32 | 82.06±0.31 |
| Connect4| 75.59±0.45 | 67.90±0.62 | 75.45±0.49 | 76.75±0.35 | 76.65±0.46 |
| W8a     | 94.40±1.21 | 85.26±3.12 | 94.31±1.13 | 97.19±0.65 | 97.46±0.38 |
| Isbi4   | 83.19±0.22 | 80.34±0.27 | 83.26±0.28 | 83.72±0.17 | 83.41±0.32 |
| TV-News | 87.53±0.29 | 78.83±2.00 | 87.50±0.29 | 88.06±0.24 | 87.94±0.42 |

1) COMPARISON OF THE ACCURACY

In Tables 2-3, we use “MK”, “MKLS”, “R-MK”, “MK-TSL-1” and “MK-TSL-3” to denote the (average) accuracy of MKSVM, MKLSSVM, Rat-MKSVM and our method MKSVM-TSL with \( k = 1, 3 \) for \( n = 8000 \) and \( n = 10000 \), respectively.

From Tables 2-3, it is obvious that for \( n = 8000 \) (or \( n = 10000 \)), all the average accuracy of MKSVM-TSL with \( k = 1, 3 \) are better than that of MKSVM, MKLSSVM and Rat-MKSVM, and the standard deviations of MKSVM-TSL with \( k = 1, 3 \) are smaller than that of other methods except for Image with \( n = 10000 \). To present which algorithm is more better by the experimental results shown in Table 2, we take the Wilcoxon signed-rank test with the significance value of \( \alpha = 0.05 \) [41] in Table 4.

### TABLE 4. Wilcoxon tests for MKSVM-TSL with \( k=1,3 \). (MK-TSL-1, MK-TSL-3), MKSVM (MK), MKLSSVM (MKLS) and Rat-MKSVM(R-MK).

| Comparison | \( R^+ \) | \( R^- \) | Hypothesis (\( \alpha = 0.05 \)) | Selected |
|------------|----------|----------|-----------------------------|---------|
| MK-TSL-1 vs. MK | 45 | 0 | Rejected MK | MK-TSL-1 |
| MK-TSL-1 vs. MKLS | 45 | 0 | Rejected MKLS | MK-TSL-1 |
| MK-TSL-1 vs. R-MK | 45 | 0 | Rejected R-MK | MK-TSL-1 |
| MK-TSL-1 vs. MK-TSL-3 | 32 | 13 | Not Rejected | MK-TSL-1 |

1. http://archive.ics.uci.edu/ml/  
2. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/
From Table 4, we can find that our algorithm introduced in Algorithm 1 with $k = 1$ has better performance compared to the other three algorithms, and Algorithm 1 with $k = 1$ has better performance compared to Algorithm 1 with $k = 3$.

In order to show the performance of MKSVM-TSL with $k = 1$ for better, we present Figs. 1-6 to compare 50-times experiments of MKSVM-TSL based on $k = 1$ with MKSVM, MKLSSVM and Rat-MKSVM. Here “red hexagram”, “blue pentagram”, “magenta square” and “green circle” denote the experimental results of MKSVM-TSL with $k = 1$, MKSVM, MKLSSVM and Rat-MKSVM, respectively. The horizontal axis and the vertical axis of figures are the experimental times and the accuracy (%), respectively.

From Figs. 1-6, we can find that for the given size ($n = 8000$ or $n = 10000$), almost all of 50 times experimental results of MKSVM-TSL with $k = 1$ are better than that of the other three algorithms, and the accuracy of MKSVM-TSL with $k = 1$ becomes much better as $n$ increases.

2) COMPARISON OF TOTAL SAMPLING AND TRAINING TIME

We use “MK”, “MKLS”, “R-MK”, “MK-TSL-1” and “MK-TSL-3” to denote the total sampling and training time of MKSVM, MKLSSVM, Rat-MKSVM, MKSVM-TSL with $k = 1, 3$, respectively.

Tables 5-6 show that the total sampling and training time of MKSVM-TSL with $k = 3$ is shorter than that of the other three algorithms for $n = 8000$ (or $n = 10000$) and the total sampling and training time of MKSVM-TSL with $k = 3$ is much shorter than that of MKSVM-TSL with $k = 1$. In particular, the sum of total sampling and training time for all training data of our proposed MKSVM-TSL algorithm with $k = 3$ is less than one-twelfth of that of MKLSSVM.
for \( n = 8000 \), and even less than one-fifteenth of that of MKLSSVM when \( n = 10000 \).

3) COMPARISON OF SUPPORT VECTOR NUMBERS
The support vector in SVM is the nonzero vector of dual variables. We call the obtained classifier is more sparse when the number of support vectors of the obtained classifier is smaller. We now compare the number of support vectors for MKSVM-TSL based on \( k = 1, 3 \) with that of other two algorithms, MKSVM and Rat-MKSVM.

| Dataset   | MK  | MKLS | R-MK | MK-TSL-1 | MK-TSL-3 |
|-----------|-----|------|------|-----------|-----------|
| Cod-md    | 651.0798 | 3400.9782 | 716.8158 | 330.7095 | **178.7169** |
| Image     | 655.9195 | 2789.1104 | 715.8290 | 340.1592 | **177.2504** |
| Census    | 698.5393 | 2846.7299 | 766.9621 | 359.5199 | **188.5408** |
| Seismic   | 726.2406 | 2856.0527 | 767.3075 | 365.8104 | **189.2263** |
| A9a       | 817.5787 | 5876.7253 | 851.1736 | 425.5344 | **234.5182** |
| Connect4  | 1010.531 | 2861.7465 | 1066.7789 | 456.0607 | **241.9805** |
| W8a       | 760.4565 | 6701.4881 | 1029.3617 | 398.7423 | **221.4682** |
| Isbi4     | 1030.3469 | 3713.4987 | 922.8793 | 625.5773 | **374.4113** |
| TV-News   | 9167.6994 | 5652.9914 | 9429.9839 | 1836.9294 | **1045.6350** |
| Sum of Time | 15518.4138 | 36699.3212 | 16267.0918 | 5139.0431 | **2851.7476** |

| Dataset   | MK  | MKLS | R-MK | MK-TSL-1 | MK-TSL-3 |
|-----------|-----|------|------|-----------|-----------|
| Cod-md    | 1023.8697 | 8405.5630 | 1130.1108 | 534.7952 | **278.8645** |
| Image     | 1001.6060 | 4961.4809 | 1119.8026 | 518.6723 | **274.6208** |
| Census    | 1094.5409 | 5243.8346 | 1212.2559 | 557.2939 | **292.4772** |
| Seismic   | 1094.6536 | 5061.1422 | 1206.0447 | 564.6051 | **291.2473** |
| A9a       | 1411.0516 | 10632.4582 | 1447.4082 | 667.2108 | **357.7101** |
| Connect4  | 1723.5436 | 5043.7989 | 1815.9318 | 737.9582 | **380.9100** |
| W8a       | 1198.7428 | 11990.6545 | 5356.4955 | 622.8071 | **339.5612** |
| Isbi4     | 1739.2814 | 5261.8564 | 1600.0260 | 910.4770 | **558.9182** |
| TV-News   | 13144.3696 | 10066.6790 | 13018.2784 | 3283.4996 | **1575.7541** |
| Sum of Time | 23431.6592 | 66667.4677 | 27906.3539 | 8397.3192 | **4350.0634** |
to signify the average number of support vectors of the corresponding algorithms for 50-times experimental results. Since the classifier obtained by MKLSSVM does not have the support vector, we did not present the number of support vectors of MKLSSVM in Tables 7-8.

**TABLE 7.** The average number of support vectors for \(n = 8000\).

| Dataset | MK | MKLS | R-MK | MK-TSL-1 | MK-TSL-3 |
|---------|----|------|------|----------|----------|
| Cod-rnd | 976.9 | 976 | 342.72 | 223.04 |
| Image   | 3499 | 3525.94 | 757.54 | 426.46 |
| Census  | 1203.58 | 1220.9 | 426.28 | 211.76 |
| Seismic | 3084.74 | 3042.22 | 720.42 | 389.36 |
| A9a     | 3442.5 | 3511.2 | 1695.48 | 1064.92 |
| Connect4 | 2848.04 | 2845.74 | 1331.02 | 829.06 |
| W8a     | 539.6 | 544.12 | 305.96 | 169.54 |
| Iabi4   | 2741.12 | 2737.34 | 946.1 | 584.3 |
| TV-News | 2119.06 | 2136.56 | 862.4 | 297.68 |

**TABLE 8.** The average number of support vectors for \(n = 10000\).

| Dataset | MK | MKLS | R-MK | MK-TSL-1 | MK-TSL-3 |
|---------|----|------|------|----------|----------|
| Cod-rnd | 1297.18 | 1285.42 | 484.4 | 324.96 |
| Image   | 4333.88 | 4379.18 | 961.68 | 497.56 |
| Census  | 1489.46 | 1482.18 | 519.7 | 258.42 |
| Seismic | 3781.3 | 3770.48 | 870.1 | 463.22 |
| A9a     | 4132.64 | 4293.32 | 1989.72 | 1190.68 |
| Connect4 | 3354.62 | 3350.68 | 1572.1 | 976.06 |
| W8a     | 643.9 | 630.88 | 356.4 | 197.84 |
| Iabi4   | 3360.54 | 3353.26 | 1155.96 | 703.54 |
| TV-News | 2566.08 | 2574.9 | 837.56 | 369.92 |

By Tables 7-8, we can find that all the number of support vectors for MKSVM-TSL with \(k = 1, 3\) are obviously smaller than that of MKSVM and Rat-MKSVM, which implies that the classifier obtained by Algorithm 1 is sparser than that of the classical MKSVM and Rat-MKSVM. And the number of support vectors for MKSVM-TSL with \(k = 3\) are much smaller than that of MKSVM-TSL with \(k = 1\).

**C. DISCUSSIONS**

From the above experiment results, we can find that although our algorithm introduced in Algorithm 1 with \(k = 1, 3\) has much better performance than the other three MKL algorithms, the different value of \(k\) is much more important in Algorithm 1. Therefore, we take \(n = 18000\) training samples to explore the performance of MKSVM-TSL with \(k = 1, 3, 5, 8\).

1) **COMPARISON OF THE ACCURACY**

In Table 9, we use “MK-TSL-1”, “MK-TSL-3”, “MK-TSL-5”, and “MK-TSL-8” to denote the (average) accuracy of MKSVM-TSL with \(k = 1, 3, 5, 8\) for \(n = 18000\), respectively.

Table 9 shows that there is little difference between the average accuracy of MKSVM-TSL with \(k = 1, 3, 5, 8\) and the average accuracy of MKSVM-TSL with \(k = 8\) are obvious smaller than that of MKSVM-TSL with \(k = 1, 3, 5\). In order to find out which algorithm is more better by the experimental results shown in Table 9, we take the Wilcoxon signed-rank test with the significance value of \(\alpha = 0.05\) [41] in Table 10.

**TABLE 9.** Average accuracy (%) for \(n = 18000\).

| Dataset | MK-TSL-1 | MK-TSL-3 | MK-TSL-5 | MK-TSL-8 |
|---------|----------|----------|----------|----------|
| Cod-rnd | 95.35±0.19 | 95.37±0.14 | 95.25±0.18 | 95.03±0.21 |
| Image   | 86.41±0.14 | 86.57±0.16 | 86.12±0.21 | 85.49±0.27 |
| Census  | 94.56±0.06 | 94.52±0.09 | 94.42±0.08 | 94.26±0.11 |
| Seismic | 83.21±0.22 | 82.69±0.24 | 82.29±0.31 | 81.66±0.33 |
| A9a     | 80.51±0.29 | 81.95±0.32 | 82.16±0.31 | 82.09±0.35 |
| Connect4 | 77.32±0.37 | 77.67±0.43 | 77.03±0.47 | 76.26±0.47 |
| W8a     | 96.31±0.20 | 97.53±0.46 | 97.64±0.37 | 97.33±0.41 |
| Iabi4   | 84.44±0.24 | 84.14±0.23 | 83.63±0.31 | 83.07±0.31 |
| TV-News | 88.74±0.21 | 88.74±0.24 | 88.38±0.26 | 87.74±0.43 |

From Table 10, we can find that our proposed MKSVM-TSL algorithm with \(k = 3\) has the best performance compared to MKSVM-TSL with \(k = 1, 5, 8\), and MKSVM-TSL with \(k = 8\) has the worst performance compared to MKSVM-TSL with \(k = 1, 3, 5\).

2) **COMPARISON OF TOTAL SAMPLING AND TRAINING TIME**

We use “MK-TSL-1”, “MK-TSL-3”, “MK-TSL-5”, and “MK-TSL-8” to denote the total sampling and training time of MKSVM-TSL with \(k = 1, 3, 5, 8\) for \(n = 18000\), respectively.

**TABLE 10.** Wilcoxon tests for MKSVM-TSL with \(k = 1, 3, 5, 8\) (MK-TSL-1, MK-TSL-3, MK-TSL-5 and MK-TSL-8).

| Comparison | \(R^+\) | \(R^-\) | Hypothesis (\(\alpha = 0.05\)) | Selected |
|------------|--------|--------|-------------------------------|----------|
| MK-TSL-1 vs MK-TSL-3 | 15.5 | 29.5 | Not Rejected | MK-TSL-3 |
| MK-TSL-1 vs MK-TSL-5 | 28 | 17 | Not Rejected | MK-TSL-1 |
| MK-TSL-1 vs MK-TSL-8 | 31 | 14 | Not Rejected | MK-TSL-1 |
| MK-TSL-3 vs MK-TSL-5 | 39 | 6 | Not Rejected | MK-TSL-3 |
| MK-TSL-3 vs MK-TSL-8 | 44 | 1 | Rejected MK-TSL-8 | MK-TSL-3 |
| MK-TSL-5 vs MK-TSL-8 | 45 | 0 | Rejected MK-TSL-8 | MK-TSL-5 |

Table 11 shows that the total sampling and training time of MKSVM-TSL with \(k = 3\) is much shorter than that of MKSVM-TSL with \(k = 1\). And almost all the total sampling and training time of MKSVM-TSL with \(k = 8\) is the shortest than that of MKSVM-TSL with \(k = 1, 3, 5\) except for...
In this section, we give the proof of the main results presented in Section IV.

**Proposition 1:** Let \( f_x \) be defined by (2), the following inequality holds
\[
\mathcal{R}(\text{sign}(\pi(f_x))) - \mathcal{R}(f_x) \leq S_1 + S_2 + S_3,
\]
where
\[
S_1 = \mathcal{E}(\pi(f_x)) - \mathcal{E}(f_x) - \mathcal{E}(\pi(f_x)) - \mathcal{E}(f_x),
\]
\[
S_2 = \mathcal{E}(f_x) - \mathcal{E}(f_x) - \mathcal{E}(f_x),
\]
\[
S_3 = \mathcal{E}(f_x) - \mathcal{E}(f_x) + \lambda \| f_x \|_K^2.
\]

**Proof:** By inequality (3), we can get
\[
\mathcal{R}(\text{sign}(\pi(f_x))) - \mathcal{R}(f_x) \leq \mathcal{E}(\pi(f_x)) - \mathcal{E}(f_x) + \lambda \| f_x \|_K^2,
\]
and \( \mathcal{E}(\pi(f_x)) - \mathcal{E}(f_x) + \lambda \| f_x \|_K^2 \leq S_1 + S_2 + S_3 + S_4, \)

Recall \( f_x \) which has been defined in (2), there holds
\[
\mathcal{E}(\pi(f_x)) + \lambda \| f_x \|_K^2 \leq \mathcal{E}(f_x) + \lambda \| f_x \|_K^2.
\]

By the definition of \( \pi \) and for any function \( f \) on \( Z \), \( |\pi(f(x)) - y| \leq |f(x) - y| \) is always true. Using these two inequalities above, the above proposition is gotten.

According to Definition 4, \( S_3 \) is bounded by \( O(\lambda^2) \) for \( 0 < \beta < 1 \). Then we have to estimate \( S_1 \) and \( S_2 \). Thus we introduce the following lemmas.

**Lemma 1:** [28] Let \( G \) be a countable class of bounded measurable functions and \( \xi_1, \xi_2, \cdots, \xi_n \) be u.e.M.c. samples. For any \( g \in G \) and \( \xi_i, i = 1, 2, \cdots, n \), assume that \( 0 \leq g(\xi_i) \leq B \) where \( B \) is a constant. Then for any \( \epsilon > 0 \) and a positive constant \( ||\Gamma_0|| \), there holds
\[
P\left\{ \frac{1}{n} \sum_{i=1}^{n} g(\xi_i) - E(g) \geq \epsilon \right\} \leq 2 \exp \left\{ \frac{-n\epsilon^2}{56B||\Gamma_0||^2E(g)} \right\}.
\]

By Lemma 1, we have the following relative uniform convergence for u.e.M.c. samples. The detailed proof can be found in [28].

**Lemma 2:** Under the same conditions of Lemma 1, for any \( \epsilon > 0 \), we have
\[
P\left\{ \sup_{g \in G} \left( \frac{1}{n} \sum_{i=1}^{n} g(\xi_i) - E(g) \right) - E(g) + \epsilon \geq 4\sqrt{\epsilon} \right\}
\]
\[
\leq N(G, \epsilon) \exp \left\{ \frac{-n\epsilon^2}{56B||\Gamma_0||^2} \right\}.
\]

**Lemma 3:** [42] Let \( c_1, c_2 > 0 \), and \( p_1 > p_2 > 0 \). The equation \( x^{p_1} - c_1x^{p_2} - c_2 = 0 \) has a unique positive zero \( x^* \).

In addition, \( x^* \leq \min\{2(c_1)^{1/(p_1-p_2)}, (2c_2)^{1/p_1}\} \).

**Proposition 2:** Let \( \mathbf{z} = [z_i = (x_i, y_i)]_{i=1}^{n} \) be u.e.M.c. samples, for any \( 0 < \eta < 1 \), \( R \geq T \) and \( n \geq 224||\Gamma_0||^2(\ln(1/\eta))^{1/2}C_{1/2}^2 \), with probability at least \( 1 - \eta/2 \), there holds
\[
S_1 \leq \frac{1}{2} \left( E(\pi(f_x)) - E(f_x) + \left( \frac{224C_1R^{s+1}||\Gamma_0||^2}{n} \right)^{1/2} \right).
\]

**Proof:** Denote \( \xi_1 = l(\pi(f_x), z) - l(f_x, z) \), that is, \( S_1 = E\xi_1 - (1/n) \sum_{i=1}^{n} \xi_i(z_i) \). Let \( \mathcal{F}_R = \{ l(\pi(f), z) - l(f, z) : f \in \mathcal{B}_R \} \) and \( g = l(\pi(f), z) - l(f, z) \), so \( \mathcal{E}_g = E(\pi(f)) - E(f) \), \( (1/n) \sum_{i=1}^{n} g(z_i) = \mathcal{E}_g \). According to Assumption 2, we have \( ||f||_\infty \leq \kappa ||f||_K \leq R \) and \( |g(z)| \leq R + T := B \) for any \( f \in \mathcal{B}_R \). By Lemma 2, we have that for any \( \epsilon > 0 \),
\[
P\left\{ \sup_{f \in \mathcal{F}_R} \left( \frac{E(\pi(f)) - E(f_x) - \mathcal{E}(\pi(f)) - \mathcal{E}(f_x)}{\sqrt{E(\pi(f)) - E(f_x)}} + \epsilon \right) \geq \sqrt{\epsilon} \right\}
\]
\[
= P\left\{ \sup_{g \in \mathcal{F}_R} \left( \frac{E(g) - \frac{1}{n} \sum_{i=1}^{n} g(z_i)}{\sqrt{E(g) + \epsilon}} \right) \geq \sqrt{\epsilon} \right\}
\]
\[
\leq N(\mathcal{F}_R, \epsilon) \exp \left\{ \frac{-n\epsilon^2}{56B||\Gamma_0||^2} \right\}.
\]
Moreover, we have \( |g_1 - g_2| \leq \| f_1 - f_2 \|_\infty \) for any \( g_1, g_2 \in \mathcal{F}_R \) and by Definition 2 and Assumption 1, \( N(\mathcal{F}_R, \epsilon) \leq N(\mathcal{B}_R, \epsilon) \leq N(\mathcal{B}_1, \epsilon/R) \), the following
inequality holds
\[
P \left\{ \sup_{f \in B_R} \frac{\mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho) - (\mathcal{E}_f(\pi(f)) - \mathcal{E}_f(f_\rho))}{\sqrt{\mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho) + \epsilon}} \geq \sqrt{c} \right\} \\
\leq \mathcal{N} \left( \frac{\epsilon}{\sqrt{R}} \right) \exp \left\{ -\frac{n \epsilon}{56\|\gamma_0\|^2(R + T)} \right\}.
\]
Recall that \( f_\rho \) is the minimization of the regularized empirical error on \( B_R \):
\[
P \left\{ \frac{\mathcal{E}(\pi(f_\rho)) - \mathcal{E}(f_\rho) - (\mathcal{E}_f(\pi(f_\rho)) - \mathcal{E}_f(f_\rho))}{\sqrt{\mathcal{E}(\pi(f_\rho)) - \mathcal{E}(f_\rho) + \epsilon}} \geq \sqrt{c} \right\} \\
\leq \mathcal{N} \left( \frac{\epsilon}{\sqrt{R}} \right) \exp \left\{ -\frac{n \epsilon}{56\|\gamma_0\|^2(R + T)} \right\} \\
\leq \exp \left\{ C_3 \frac{R}{\epsilon} - \frac{n \epsilon}{56\|\gamma_0\|^2(R + T)} \right\}.
\]
Set \( \eta \) is equal to the right-hand side of the above inequality, that is
\[
\eta = \exp \left\{ C_3 \frac{R}{\epsilon} - \frac{n \epsilon}{56\|\gamma_0\|^2(R + T)} \right\}.
\]
Then, we get
\[
\epsilon^{s+1} - \frac{56(R + T)\ln(1/\eta)\|\gamma_0\|^2}{n} \leq \frac{56C_\rho(R + T)\epsilon^s\|\gamma_0\|^2}{n} = 0.
\]
Applying Lemma 3, the solution of above equation is given by \( \epsilon = \epsilon^s(n, \eta) \leq \max \{ n_1, n_2 \} \), where
\[
n_1 = \frac{112(R + T)\ln(1/\eta)\|\gamma_0\|^2}{n},
\]
\[
n_2 = \left( \frac{112C_\rho(R + T)\|\gamma_0\|^2}{n} \right)^{\frac{1}{1+s}}.
\]
Let \( R \geq T \) and \( n \geq 224\|\gamma_0\|^2(\ln(1/\eta))^{1+1/(C_3^{1/4})} \), we have that for any \( 0 < \eta < 1 \), the following inequality is valid with probability at least \( 1 - \eta/2 \):
\[
S_1 = \mathcal{E}(\pi(f_\rho)) - \mathcal{E}(f_\rho) - (\mathcal{E}_f(\pi(f_\rho)) - \mathcal{E}_f(f_\rho)) \\
\leq \sqrt{\mathcal{E}^2(\pi(f_\rho)) - \mathcal{E}(f_\rho) + \epsilon} + \epsilon \\
\leq \frac{1}{2} \left\{ \mathcal{E}(\pi(f_\rho)) - \mathcal{E}(f_\rho) + \frac{224C_\rho\|\gamma_0\|^2}{n} \right\}^{\frac{1}{1+s}}.
\]
Proposition 3: For any \( 0 < \eta < 1 \), we have that the inequality
\[
S_2 \leq \frac{1}{2} D(\lambda) + \frac{28(\sqrt{D(\lambda)/\lambda} + T)\ln(2/\eta)\|\gamma_0\|^2}{n}
\]
holds true with probability at least \( 1 - \eta/2 \).
Proof: From the definition of \( f_\rho \) and Definition 4, we have
\[
\lambda \|f_\rho\|^2 \leq \mathcal{E}_f(f_\rho) - \mathcal{E}(f_\rho) + \lambda \|f_\rho\|^2 = D(\lambda).
\]
Set \( \xi_2 = I(f_\rho, z) - I(f_\rho, z) \), then we have
\[
S_2 = \mathcal{E}(\pi(f_\rho)) - \mathcal{E}(f_\rho) = D(\lambda).
\]
Because \( |f_\rho| \leq T \), we have
\[
|\xi_2| \leq \sqrt{D(\lambda)/\lambda} + T := C_1.
\]
By Lemma 1, take \( \eta = \exp \{-nc^2/56C_1\|\gamma_0\|^2E(\xi_2)\} \). Then
\[
\epsilon = \sqrt{\frac{56C_1\|\gamma_0\|^2E(\xi_2)\ln(1/\eta)}{n}}.
\]
Hence, with probability at least \( 1 - \eta/2 \), we have
\[
\frac{1}{n} \sum_{i=1}^{n} \xi_2(z_i) - E\xi_2 \\
\leq \frac{1}{n} \left( E\xi_2 + \frac{56C_1\ln(2/\eta)\|\gamma_0\|^2}{n} \right) \\
\leq \frac{1}{2} D(\lambda) + \frac{28(\sqrt{D(\lambda)/\lambda} + T)\ln(2/\eta)\|\gamma_0\|^2}{n}.
\]
Then we complete the proof of Proposition 3.
Proof of Theorem 1: Combining the upper bounds of \( S_1, S_2, S_3 \), we have that for any \( 0 < \eta < 1 \), the following inequality
\[
\mathcal{E}(\pi(f_\rho)) - \mathcal{E}(f_\rho) \leq 2R \frac{224C_\rho\|\gamma_0\|^2}{n}^{\frac{1}{1+s}} + 3D(\lambda) \\
+ \frac{56(\sqrt{D(\lambda)/\lambda} + T)\ln(2/\eta)\|\gamma_0\|^2}{n}
\]
is valid with probability at least \( 1 - \eta \). Then The proof of Theorem 1 is finished.
Proof of Theorem 2: Inspired by the idea from [43], let \( R = \sqrt{D(\lambda)/\lambda} \) and for any \( 0 < \eta < 1 \), with probability at least \( 1 - \eta \), the following inequality holds
\[
\mathcal{R}(\text{sign}(\pi(f_\rho))) - \mathcal{R}(f_\rho) \\
\leq 3D(\lambda) + 2(\sqrt{D(\lambda)/\lambda} + T)\ln(2/\eta)\|\gamma_0\|^2 \\
+ \frac{56(\sqrt{D(\lambda)/\lambda} + T)\ln(2/\eta)\|\gamma_0\|^2}{n}
\]
\[
\leq 3D(\lambda) + \frac{C_1}{n} + \frac{C_2}{n} \lambda^{-\beta_1} + \frac{C_3}{n} \lambda^{-\beta_1} \\
= 2T(224C_\rho\|\gamma_0\|^2)\ln(2/\eta)\|\gamma_0\|^2
\]
Since
\[
\frac{1}{n} \lambda^{-\beta_1} \leq \lambda^\beta, \left( \frac{1}{n} \right)^{\frac{\beta_1}{1+s}} \leq \lambda^\beta,
\]
then for any \( 0 < \eta < 1 \), with probability at least \( 1 - \eta \), the following inequality holds
\[
\mathcal{R}(\text{sign}(\pi(f_\rho))) - \mathcal{R}(f_\rho) \leq \bar{C} \left( \frac{2^\beta}{n^{\beta_1}} \right)^{\frac{1}{1+s}},
\]
where \( \bar{C} \) is a constant independent of \( n \). Then the proof of Theorem 2 is finished.
YUZE DUAN received the B.E. degree in statistics from the Shanxi University of Finance and Economics, China, in 2017. He is currently pursuing the M.S. degree with the Faculty of Mathematics and Statistics, Hubei University, China.

JIE XU is currently a Professor with the Faculty of Computer Science and Information Engineering, Hubei University, Wuhan, China. Her research interests include machine learning and pattern recognition.

QINGXIN LUO is currently pursuing the B.S. degree with the Faculty of Mathematics and Statistics, Hubei University, China.

YAN YANG received the B.E. degree from the Faculty of Mathematics and Computer Science, Hubei University, China, in 2010. She is currently an Associate Professor and the Head of the Software Engineering Department, Faculty of Computer Science, Wuhan Qingchuan University.