Learning With Selected Features

Shao-Bo Lin, Jian Fang, and Xiangyu Chang

***Abstract*—The coming *big data* era brings data of unprecedented size and launches an innovation of learning algorithms in statistical and machine-learning communities. The classical kernel-based regularized least-squares (RLS) algorithm is excluded in the innovation, due to its computational and storage bottlenecks. This article presents a scalable algorithm based on subsampling, called learning with selected features (LSF), to reduce the computational burden of RLS. Almost the optimal learning rate together with a sufficient condition on selecting kernels and centers to guarantee the optimality is derived. Our theoretical assertions are verified by numerical experiments, including toy simulations, UCI standard data experiments, and a real-world massive data application. The studies in this article show that LSF can reduce the computational burden of RLS without sacrificing its generalization ability very much.**

***Index Terms*—Learning theory, regularized least squares (RLS), selected features, subsampling, uniqueness set.**

I. Introduction

With the development of the technique in data acquisition, collecting hluge quantities of data becomes increasingly frequent. For example, thousands of high-resolution satellite images are analyzed for geographical investigations; hundreds of thousands of Internet URLs are inspected for the detection of pop-up junk messages; and millions of customer transactions are gathered for making marketing decisions. These massive data bring new opportunities for discovering subtle population patterns which are not shown by data of small size, and simultaneously, produce a series of scientific challenges, such as the storage bottleneck and algorithmic scalability [66]. Learning algorithms of high quality, especially of low computational complexity, are desired to conquer the massive data challenges in statistical and machine-learning communities.

Kernel methods [18], which focus on mapping data points from the input space to some feature space where a linear method is sufficient to find the estimator, have been widely used in computer vision, financial studies, and engineering [5], [49], [51]. Kernel-based regularized least squares (RLS) [18] is a typical kernel method and receives the impact first due to its extensive computational burden in both memory and time in the wave of algorithmic innovations for massive data. Several scalable variants of RLS were developed, including the localized RLS [27], distributed RLS [64], RLS with subsampling [19], and RLS with randomized sketches [60]. Localized RLS first divides the input space into ℓ disjoint partitions and then runs RLS on the samples, whose inputs are located on the partition containing the query point. Distributed RLS starts with partitioning the dataset into ℓ disjoint subsets, then assigns each data subset to a local machine to produce a local estimator by using RLS, and finally, synthesizes a global estimator by (weighted) averaging all local estimators. RLS with subsampling, containing the *Nyström regularization* [58] and learning with random features (LRF) [33], randomly selects centers of kernel with small size in a data-dependent (or independent) way. RLS with randomized sketches, such as sub-Gaussian sketches and randomized orthogonal system sketches, is a generalized subsampling strategy, which projects the kernel matrix to a lower dimensional space. The feasibility of these modifications has been verified in [27] and [50] for localized RLS; [13], [24], and [64] for distributed RLS; [36] and [37] for RLS with subsampling; and [60] for RLS with randomized sketches in terms of providing the same optimal learning rates as RLS.

This article aims at introducing a novel scalable variant of RLS based on subsampling to tackle massive data challenges. Given $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m \subset \mathcal{X} \times \mathcal{Y}$ with input space $\mathcal{X} \subseteq \mathbb{R}^d$ and output space $\mathcal{Y} \subseteq \mathbb{R}$, RLS is defined by

$$f_{\mathbf{z},\lambda} := \arg\min_{f\in\mathcal{H}_K} \left\{ \frac{1}{m}\sum_{i=1}^m (f(x_i)-y_i)^2 + \lambda\|f\|_K^2 \right\} \tag{1}$$

where $\lambda > 0$ is a regularization parameter and $(\mathcal{H}_K, \|\cdot\|_K)$, induced by a Mercer kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, is called the reproducing kernel Hilbert space (RKHS). Due to the representation theorem in the learning theory [11], the final estimator derived by (1) is in the sample-dependent hypothesis space $\mathcal{H}_{m,K} := \{f = \sum_{i=1}^m a_i K_{x_i}\}$, where $K_{x_i} = K(x_i, \cdot)$. Then, $f_{\mathbf{z},\lambda}$ in (1) can be solved by

$$f_{\mathbf{z},\lambda} := \arg\min_{f\in\mathcal{H}_{m,K}} \left\{ \frac{1}{m}\sum_{i=1}^m (f(x_i)-y_i)^2 + \lambda\|f\|_K^2 \right\}. \tag{2}$$

Algorithm[1] (2) and then algorithm (1) can be solved by a standard matrix inverse technique requiring complexities of

Manuscript received June 13, 2019; revised January 13, 2020; accepted March 25, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61876133, Grant 11771012, and Grant 61977038, in part by NIH under Grant R01GM109068, Grant R01MH104680, and Grant R01MH107354, and in part by NSF under Grant 1539067. This article was recommended by Associate Editor Y.-M. Cheung. *(Corresponding author: Jian Fang.)*

Shao-Bo Lin and Xiangyu Chang are with the Center of Intelligent Decision-Making and Machine Learning, School of Management, Xi'an Jiaotong University, Xi'an 710049, China.

Jian Fang is with the Department of Biomedical Engineering, Tulane University, New Orleans, LA 70118 USA (e-mail: jianfang86@gmail.com).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

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

Digital Object Identifier 10.1109/TCYB.2020.2987810

[1]If the solution to an optimization problem can be analytically derived and uniquely determined, we call the optimization problem an algorithm throughout this article.

$\mathcal{O}(m^3)$, $\mathcal{O}(m^2)$, and $\mathcal{O}(m)$ in training time, memory, and testing time, respectively. RLS with subsampling replaces $\mathcal{H}_{m,K}$ in algorithm (2) by an n-dimensional linear space $\widetilde{\mathcal{H}}_n := \{f = \sum_{i=1}^n a_i K_{\theta_i}, a_i \in \mathbb{R}\}$ with $n \in \mathbb{N}$, $n \ll m$, $K_{\theta_i}(\cdot) = K(\theta_i, \cdot)$ and the center set $\{\theta_i\}_{i=1}^n \subset \mathcal{X}$. It can be mathematically stated as

$$f_{\mathbf{z},n,\lambda} := \arg\min_{f\in\widetilde{\mathcal{H}}_n} \left\{ \frac{1}{m}\sum_{i=1}^m (f(x_i) - y_i)^2 + \lambda \|f\|_K^2 \right\}. \tag{3}$$

When $\{\theta_j\}_{j=1}^n$ are randomly drawn from $\{x_i\}_{i=1}^m$, algorithm (3) is the plain Nyström regularization [58]. When $\{\theta_j\}_{j=1}^n$ are randomly drawn in a data-independent way, algorithm (3) coincides with LRF [33]. We refer the readers to [19] for detailed strategies on selecting $\{\theta_j\}_{j=1}^n$ for algorithm (3). For fixed $n \ll m$ and $\lambda > 0$, algorithm (3) reduces the complexities of RLS from $\mathcal{O}(m^3)$, $\mathcal{O}(m^2)$, and $\mathcal{O}(m)$ to $\mathcal{O}(n^2 m)$, $\mathcal{O}(nm)$, and $\mathcal{O}(n)$ in training time, memory, and testing time.

Theoretical analysis of algorithm (3) has been carried out by [22], [36], and [37] in the framework of the statistical learning theory, yielding that algorithm (3) can achieve the optimal learning rate of algorithm (1) provided the features[2] are appropriately selected. Along the analysis in [22], [36], and [37], the number of features can be regarded as a *computational regularization* to reflect the computational complexity and generalization ability simultaneously. In other words, a small number of features usually leads to low computational complexity but large generalization error, while a large number of features requires a high computational burden. On the contrary, the *statistical regularization* of the penalty $\lambda\|f\|_K^2$ was introduced to conquer the overfitting only, which may increase the computational complexity when used together with computational regularization. Since the computational regularization and statistical regularization play the same role in generalization, if we can skip the statistical regularization in (1) and (3), the computational time in parameter tuning can be considerably saved and the potential of computational regularization can be fully explored. To this end, we study the learning performance of the algorithm

$$f_{\mathbf{z},n} := \arg\min_{f\in\widetilde{\mathcal{H}}_n} \left\{ \frac{1}{m}\sum_{i=1}^m (f(x_i) - y_i)^2 \right\}. \tag{4}$$

As the regularization term is removed, some additional restrictions to the features should be imposed to guarantee the learning performance. In this way, we call algorithm (4), equipped with appropriately selected features, as learning with selected features (LSF). Different from RLS with subsampling, LSF utilizes the idea of computational regularization to replace the statistical regularization to reduce the computational burden of RLS.

Borrowing an idea of the *uniqueness set* from the paper [26] in the approximation theory, we find that if the kernel is radial (see Definition 1 in Section II) with an analytic and nonpolynomial link function, and the center set contains a uniqueness set for some polynomials (see Definition 2 in Section II), then LSF can achieve the almost optimal learning rate of RLS as long as the number of features is properly tuned.

To facilitate the use of LSF, we also prove that the Lebesgue measure of the family of sets that are not uniqueness sets is 0, showing that the centers can be either randomly drawn according to some distribution continuous with the Lebesgue measure, or deterministically selected according to some discrepancy principle [30]. Our theoretical assertions are verified by numerical studies, including toy simulations, UCI standard data applications, and a music-prediction task containing over 500 000 samples. All the experiments show that LSF possesses the comparable generalization ability as the widely used algorithms, such as the Nyström regularization, LRF, learning with randomized sketches, and RLS with less computational costs.

Overall, the main contributions of this article can be summarized two-fold.

1) Theoretically, we introduce a sufficient condition on selecting kernels and centers, equipped with which, LSF can achieve near-optimal learning rates of RLS.
2) Computationally, we simplify the training of Nyström-type regularization by skipping the statistical regularization, which significantly reduces the training time.

The remainder of this article is organized as follows. In the next section, we introduce the uniqueness set and radial functions, and then propose the LSF algorithm in detail. Section III provides our main results, where almost optimal learning rates are derived in the framework of the statistical learning theory. In Section IV, we compare our theoretical results with some related work and give some further discussions. Section V presents the numerical verifications for our theoretical assertions. In Section VI, we draw a simple conclusion for this article.

II. Learning With Selected Features

In this section, we present a strategy to select features and propose a new algorithm. For this purpose, we need definitions of the radial kernel [42] and the uniqueness set [26].

Definition 1: We say that a function $K^\delta(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ is a radial kernel with parameter δ if there exists a link function $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ such that $K^\delta(x, y) = \phi(\delta\|x - y\|^2)$, where $\|x\|$ denotes the Euclidean norm for $x \in \mathbb{R}^d$.

The definition of the uniqueness set is a bit technical. It comes from a nice paper [26] in the approximation theory and depends on a delicate construction of orthonormal basis for polynomials, which will be given in Appendix A for the sake of completeness. Let $\mathbb{B}^d$ be the unit ball in $\mathbb{R}^d$ and $\mathcal{P}_s^d$ be the family of algebraic polynomials defined on $\mathbb{B}^d$ of degree at most s. Denote by

$$\left\{P_{k,j,i} : k = 0, \ldots, s, j = k, k-2, \ldots, \tau_k, i = 1, 2, \ldots, D_j^{d-1}\right\}$$

an orthonormal basis for $\mathcal{P}_s^d$, where $\tau_k := \begin{cases} 0, & k \text{ even} \\ 1, & k \text{ odd} \end{cases}$, $D_j^{d-1} \sim j^{d-2}$ and $A \sim B$ denotes that there exist absolute

[2]We call $K(\theta_j, x)$ as a feature along the nomination system in [37]. In particular, there exists a kernel $\widetilde{K}(x, x') = \int_{\mathcal{X}} K(t, x)K(t, x')d\vartheta(t)$ with $\vartheta(\cdot)$ a density (or weight) function on $\mathcal{X}$. Thus, $K(\theta_j, \cdot) : \mathcal{X} \to \mathbb{R}$ can be regarded as a feature mapping.

Algorithm 1 LSF

Input: Let $\{(x_i, y_i)\}_{i=1}^m$ be the sample set and $n \in \mathbb{N}$ be the number of selected features.

Step 1: Select a set of centers $\mathcal{A}_n := \{\alpha_j\}_{j=1}^n$ to contain a uniqueness set for $\mathcal{Q}_{2s}^d$ for some $s \in \mathbb{N}$.

Step 2: Select a radial kernel K^δ with parameter δ and link function ϕ, which possesses up to $s+1$ times bounded derivatives, and satisfies $\phi^{(\ell)}(0) \neq 0$ for all $\ell = 0, 1, \ldots, s$.

Step 3: Write $\mathbf{y} := (y_1, \ldots, y_m)^T$ and $\mathbf{c} := (c_1, \ldots, c_n)^T$. Generate the kernel matrix $A_{m,n} := (K^\delta(x_i, \alpha_j))_{i,j=1}^{m,n}$. Solve the ordinary least squares

$$(c_1^*, \ldots, c_n^*)^T = \arg\min_{\mathbf{c}\in\mathbb{R}^n} \|A_{m,n}\mathbf{c} - \mathbf{y}\|^2. \tag{9}$$

Output: $f_{\mathbf{z},n,\delta} = \sum_{j=1}^n c_j^* K_{\alpha_j}^\delta$.

constants C_1 and C_2 such that $C_1 A \leq B \leq C_2 A$. Define

$$\mathcal{Q}_{2s}^d := \left\{ \sum_{(k,j,i)\in\mathbf{I}_s} a_{k,j,i} P_{k,j,i} : a_{k,j,i} \in \mathbb{R} \right\} \tag{5}$$

with

$$\mathbf{I}_s := \Big\{(k,j,i) : k + j \leq 2s, k = 0, \ldots, 2s \\ j = k, k-2, \ldots, \tau_k,\ i = 1, 2, \ldots, D_j^{d-1}\Big\}. \tag{6}$$

It is easy to see

$$\mathcal{P}_s^d \subseteq \mathcal{Q}_{2s}^d \subseteq \mathcal{P}_{2s}^d. \tag{7}$$

Thus, the dimension of $\mathcal{Q}_{2s}^d$, denoted by T_s, satisfies $\binom{s+d}{s} \leq T_s \leq \binom{2s+d}{2s}$ and $T_s \sim s^d$. We then present the definition of the uniqueness set for $\mathcal{Q}_{2s}^d$ [26].

Definition 2: For arbitrary $s \in \mathbb{N}$, we say that $\Theta_s := \{\theta_i \in \mathbb{B}^d : 1 \leq i \leq T_s\}$ is a uniqueness set for $\mathcal{Q}_{2s}^d$, if, for any two polynomials P_1 and P_2 from $\mathcal{Q}_{2s}^d$, $P_1(\theta_i) = P_2(\theta_i), i = 1, \ldots, T_s$, implies $P_1(\theta) = P_2(\theta)$ for all $\theta \in \mathbb{B}^d$.

According to Definition 2, Θ_s is a uniqueness set for $\mathcal{Q}_{2s}^d$ if and only if for any $P \in \mathcal{Q}_{2s}^d$

$$P(\theta_i) = 0 \to P \equiv 0, \quad i = 1, \ldots, T_s. \tag{8}$$

If Θ_s is not the zero set for arbitrary $P \in \mathcal{Q}_{2s}^d$, (8) obviously holds for any $P \in \mathcal{Q}_{2s}^d$. On the contrary, if Θ_s is a zero set for some polynomial $P \in \mathcal{Q}_{2s}^d$, then (8) does not hold for this polynomial, implying that Θ_s is not a uniqueness set for $\mathcal{Q}_{2s}^d$. Hence, Θ_s is a uniqueness set for $\mathcal{Q}_{2s}^d$ if and only if it is not the zero set for arbitrary $P \in \mathcal{Q}_{2s}^d$.

With the help of the above two definitions, we are in a position to develop the new learning algorithm, LSF, in Algorithm 1.

To facilitate the use of LSF, we should present some guidance. In step 1, it seems difficult to judge whether a set of points contains a uniqueness set for $\mathcal{Q}_{2s}^d$ for some $s \in \mathbb{N}$ at the first glance, since $\mathcal{Q}_{2s}^d$ itself needs a delicate construction. Thus, a method to judge whether a set of points is the uniqueness set for $\mathcal{Q}_{2s}^d$ is critical. We present a sufficient condition in Theorem 1, which will be proved in Appendix B.

Theorem 1: Let $s \in \mathbb{N}$ and T_s be the dimension of $\mathcal{Q}_{2s}^d$. Then, the Lebesgue measure of the set

$$\mathcal{B}_{T_s} := \left\{\Theta_s : \Theta_s \text{ is not a uniqueness set for } \mathcal{Q}_{2s}^{\mathrm{d}}\right\} \tag{10}$$

is zero.

Due to Theorem 1, arbitrary T_s points are almost surely the uniqueness set for $\mathcal{Q}_{2s}^d$ under the Lebesgue measure. In particular, arbitrary n centers satisfying $T_s \leq n < T_{s+1}$, and are not generated on lower dimensional manifold contain a uniqueness set for $\mathcal{Q}_{2s}^d$ almost surely. In practice, we recommend three strategies to select centers in step 1. The first one is to choose $\{\alpha_j\}_{j=1}^n$ randomly according to the uniform distribution from $\mathcal{X}$ since the uniform distribution is continuous with the Lebesgue measure. The second one is to generate $\{\alpha_j\}_{j=1}^n$ deterministically as the *Sobel sequences* or *Halton sequences* [30]. The third strategy is to select $\{\alpha_j\}_{j=1}^n$ randomly according to the uniform distribution from $\{x_i\}_{i=1}^m$, if the inputs of the sample are not generated on lower dimensional manifolds. In fact, if $\{x_i\}_{i=1}^m$ is sampled from a manifold, then the Lebesgue measure of an arbitrary subset of inputs is zero. Under this circumstance, Theorem 1 is not sufficient to guarantee the uniqueness. It would be interesting to derive the corresponding sufficient condition for this case.

In step 2, there are totally three restrictions on kernels: radial, smooth, and derivative values on zero. The condition on the link function ϕ involves a parameter s, which is difficult to determine. It is easy to check that arbitrary analytic link function ϕ with $\phi^{(\ell)}(0) \neq 0$, $\ell = 0, 1, \ldots,$ satisfies these three conditions for all $s \in \mathbb{N}$. Thus, we recommend to use link functions, such as the exponential function (making K^δ be the Gaussian kernel), multiquadratics, logistic function, and inverse multiquadratics [57]. The selection of kernel to satisfy these three conditions is crucial, since we will show in Section V that improper link functions will degrade the performance of LSF.

In step 3, the ordinary least squares is equivalent to the empirical risk minimization on the linear space

$$\overline{H}_{K,n,\delta} = \left\{ f = \sum_{j=1}^n c_j K_{\alpha_j}^\delta, c_j \in \mathbb{R} \right\} \tag{11}$$

that is

$$f_{\mathbf{z},n,\delta} = \arg\min_{f\in\overline{H}_{K,n,\delta}} \left\{ \frac{1}{m}\sum_{i=1}^m (f(x_i) - y_i)^2 \right\}. \tag{12}$$

Solving (12) or (9) is a classical and simple linear optimization problem. In particular, we can utilize the pseudoinverse technique [34] or the well-known QR decomposition [40] to get the final estimator.

In a nutshell, LSF can be regarded as a two-stage learning scheme. It selects appropriate features to build up the hypothesis space $\overline{H}_{K,n,\delta}$ in the first stage and solves a linear least-squares problem (12) in the second stage. Thus, it requires the complexities of $\mathcal{O}(mn^2)$, $\mathcal{O}(mn)$, and $\mathcal{O}(n)$ in training time, memory,

and testing time, respectively. There are totally two parameters n and δ in the learning process. A preferable approach to select them in practice is the well-known cross-validation [20]. The theoretical verifications of the feasibility of cross-validation were developed in [20, Ch. 8] in expectation and [8] in probability.[3] Numerically, motivated by [10], we will show in Section V that the optimal values of parameters are easy to be found via cross-validation in terms of plotting contours with good shape of available parameters.

III. Theoretical Behaviors

We analyze the learning performance of LSF in the standard framework of the learning theory [11]. Suppose that the sample $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m$ is assumed to be independently drawn according to an unknown distribution $\rho := \rho_X \rho(y|x)$ with ρ_X the marginal distribution and $\rho(y|x)$ the condition distribution on x. Without loss of generality, we assume $\mathcal{X} = \mathbb{B}^d$ and $\mathcal{Y} = [-M, M]$ for some $M > 0$. Suppose further $f : \mathcal{X} \to \mathcal{Y}$ is a function that one uses to model the correspondence between x and y. A natural measurement of the error is the generalization error, defined by

$$\mathcal{E}(f) := \int_{\mathcal{Z}} (f(x) - y)^2 d\rho$$

which is minimized by the regression function $f_\rho(x) := \int_{\mathcal{Y}} y d\rho(y|x)$. Let $L^2_{\rho_X}$ be the Hilbert space of ρ_X square integrable functions on $\mathcal{X}$ with norm $\|\cdot\|_\rho$. With the assumption that $y \in [-M, M]$, it is known [11] that $f_\rho \in L^2_{\rho_X}$ and

$$\mathcal{E}(f) - \mathcal{E}(f_\rho) = \|f - f_\rho\|_\rho^2 \quad \forall f \in L^2_{\rho_X}. \tag{13}$$

Before presenting the main result, we need the following assumption on the regression function f_ρ. Let $r = u + v$ for some $u \in \mathbb{N}_0 := \{0\} \cup \mathbb{N}$ and $0 < v \leq 1$. A function $f : \mathcal{X} \to \mathbb{R}$ is said to be (r, c_0)-smooth if for every $\alpha = (\alpha_1, \ldots, \alpha_d), \alpha_i \in N_0$, $\sum_{j=1}^d \alpha_j = u$, the partial derivatives $[(\partial^u f)/(\partial {x_1}^{\alpha_1} \cdots \partial {x_d}^{\alpha_d})]$ exist and satisfy

$$\left| \frac{\partial^u f}{\partial {x_1}^{\alpha_1} \cdots \partial {x_d}^{\alpha_d}}(x) - \frac{\partial^u f}{\partial {x_1}^{\alpha_1} \cdots \partial {x_d}^{\alpha_d}}(x') \right| \leq c_0 \|x - x'\|^v.$$

Denote by $\mathrm{Lip}^{(r,c_0)}$ the set of all (r, c_0)-smooth functions.

Assumption 1: $f_\rho \in \mathrm{Lip}^{(r,c_0)}$ for $r > 0$ and $0 < c_0 < \infty$.

Assumption 1 describes the smoothness of the regression function and is a standard assumption in the learning theory. It has been adopted in [17], [23], [25], [45], [46], and [65] to quantify learning rates for various algorithms.

Let $\pi_M t$ denote the clipped value of t at $\pm M$, that is, $\pi_M t := \min\{M, |t|\}\mathrm{sgnt}$. Then, it is obvious that

$$\mathcal{E}(\pi_M f_{\mathbf{z},n,\delta}) - \mathcal{E}(f_\rho) \leq \mathcal{E}(f_{\mathbf{z},n,\delta}) - \mathcal{E}(f_\rho).$$

The clipped operator [17], [20], [65] is a standard technique to improve the generalization ability for some algorithms. Based on these preliminaries, we present the following main result, whose proof will be given in Appendix C.

Theorem 2: Let $\varepsilon > 0$ and $f_{\mathbf{z},n,\delta}$ be the estimator defined in Algorithm 1 with $T_s \leq n < T_{s+1}$ for some $s \in \mathbb{N}_0$. If Assumption 1 holds and $s \sim \varepsilon^{-1/(2r)}$, then there exist a $\delta^* \in (0, 1/4)$ depending on ε, positive constants C_i, $i = 1, \ldots, 4$, depending only on M, r, c_0, and d, an absolute constant $L_0 > 0$, and $\varepsilon_m^-, \varepsilon_m^+$ satisfying

$$\mathrm{C}_1 m^{-2r/(2r+d)} \leq \varepsilon_m^- \leq \varepsilon_m^+ \leq \mathrm{C}_2 (m/\log m)^{-2r/(2r+d)} \tag{14}$$

such that for any $\varepsilon < \varepsilon_m^-$ and $\delta > 0$

$$\sup_{f_\rho \in \mathrm{Lip}^{(r,c_0)}} \mathrm{Prob}\left\{\mathbf{z} : \|f_\rho - \pi_M f_{\mathbf{z},n,\delta}\|_\rho^2 > \varepsilon\right\} \geq L_0 \tag{15}$$

and for any $\varepsilon \geq \varepsilon_m^+$ and $\delta \in (0, \delta^*]$

$$e^{-\mathrm{C}_3 m\varepsilon} \leq \sup_{f_\rho \in \mathrm{Lip}^{(r,c_0)}} \mathrm{Prob}\left\{\mathbf{z} : \|f_\rho - \pi_M f_{\mathbf{z},n,\delta}\|_\rho^2 > \varepsilon\right\} \leq e^{-\mathrm{C}_4 m\varepsilon}. \tag{16}$$

In real-world applications, we are frequently faced with an important problem that whether a sample of size m is sufficient to produce an estimator with accuracy at most ε, and required to deduce the probability of success. The probability obviously depends on m and ε. If m is small, it is impossible to construct an estimator with very small prediction accuracy. This fact is quantitatively verified by (15). More important, (16) reveals a quantitive relation between the probability of success and the prediction accuracy based on a sample of size m. It says in (16) that if the accuracy ε is relaxed to ε_m^+ or larger, then the probability of success of LSF is at least $1 - e^{-\mathrm{C}_4 m\varepsilon}$. The first inequality (lower bound) of (16) implies that the probability cannot be improved further, showing an optimal confidence estimate for LSF.

The values ε_m^- and ε_m^+ are critical for indicating learning rates of LSF. In order to have a tight learning rate estimate, we naturally wish to make the interval $[\varepsilon_m^-, \varepsilon_m^+]$ as narrow as possible. Theorem 2 shows that $\varepsilon_m^- \geq \mathrm{C}_1 m^{-2r/(2r+d)}$ and $\varepsilon_m^+ \leq \mathrm{C}_2 (m/\log m)^{-2r/(2r+d)}$, implying that the interval $[\varepsilon_m^-, \varepsilon_m^+]$ is almost the narrowest in the sense that up to a logarithmic factor, the upper and lower bounds are asymptotically identical. The most important discovery in Theorem 2 is that there is a sharp phase-transition phenomenon of LSF in terms that the probability of success changes dramatically within the critical interval $[\varepsilon_m^-, \varepsilon_m^+]$. It drops from a constant L_0 to an exponentially small quantity. We might call $[\varepsilon_m^-, \varepsilon_m^+]$ the interval of phase transition for LSF.

Theorem 2 potentially exhibits another phase-transition phenomenon with respect to the prediction accuracy and the number of features. Since the center set of LSF must contain a uniqueness set for $\mathcal{Q}_{2s}^d$ in our analysis, the learning performance of LSF with $n \in [T_s, T_{s+1})$ does not change very much for arbitrary $s \in \mathbb{N}$. However, when n varies from $T_{s+1} - 1$ to T_{s+1}, the learning performance changes dramatically since in the former case, $\mathcal{A}_n$ contains a uniqueness set for $\mathcal{Q}_{2s}^d$, while in the latter case, $\mathcal{A}_n$ includes a uniqueness set for $\mathcal{Q}_{2s+2}^d$. We will show in Section V, the mentioned two phase-transition phenomena and numerically verify the rationality of introducing the uniqueness set in LSF.

Based on Theorem 2, we can derive the following corollary, which reveals the tradeoff between bias and variance reflected by the number of selected features.

[3] We can derive the feasibility of cross-validation by using the same method in [8].

Corollary 1: Let $0 < \delta < 1$ and $f_{\mathbf{z},n,\delta}$ be the estimator defined in Algorithm 1 with $T_s \le n < T_{s+1}$ for some $s \in \mathbb{N}_0$. If Assumption 1 holds, then there exists a $\delta^* \in (0, 1/4)$ depending on m such that for any $\delta \in (0, \delta^*]$, there holds

$$\left\|\pi_M f_{\mathbf{z},n,\delta} - f_\rho\right\|_\rho^2 \le \tilde{C}\left(\frac{n\log m}{m} + n^{-\frac{2r}{d}}\right)\log\frac{3}{\delta} \tag{17}$$

where $\tilde{C}$ is a constant depending only on M, d, c_0, and r.

Corollary 1 shows that, like the statistical regularization [17], the computational regularization can be used to avoid overfitting via selecting $n \sim s^d \sim m^{\lfloor d/(2r+d)\rfloor}$ features. Let $\mathcal{M}(\Theta)$ be the class of all Borel measures ρ on Z such that $f_\rho \in \Theta$. Recall that we do not know ρ so that the best we can say about it is that it lies in $\mathcal{M}(\Theta)$. We enter into a competition over all estimators $\mathbb{E}_m : \mathbf{z} \to f_{\mathbf{z}}$ and define

$$e_m(\Theta) := \inf_{\mathbb{E}_m} \sup_{\rho\in\mathcal{M}(\Theta)} E\left(\left\|f_\rho - f_{\mathbf{z}}\right\|_\rho^2\right).$$

It is easy to see that $e_m(\Theta)$ quantitively measures the quality of $f_{\mathbf{z}}$. If $\Psi = \{f \in \mathrm{Lip}^{(r,c_0)} : \|f\|_\infty \le M\}$, then it can be found in [20, Ch. 3] that

$$e_m(\Psi) \ge \mathrm{C}_5 m^{-\frac{2r}{2r+d}},\ m = 1, 2, \ldots \tag{18}$$

where C_5 is a constant depending only on M and d. In this way, we can derive the following almost optimal learning rates for LSF in expectation.

Corollary 2: Let $f_{\mathbf{z},n,\delta}$ be the estimator defined in Algorithm 1 with $T_s \le n < T_{s+1}$ for some $s \in \mathbb{N}_0$. If Assumption 1 holds and $n \sim m^{d/(2r+d)}$, then there exists a $\delta^* \in (0, 1/4)$ depending on m such that for any $\delta \in (0, \delta^*]$, there holds

$$\begin{aligned}\mathrm{C}_5 m^{-2r/(2r+d)} &\le \sup_{f_\rho\in\mathrm{Lip}^{(r,c_0)}} E\left\{\left\|f_\rho - \pi_M f_{\mathbf{z},n,\delta}\right\|^2\right\} \\ &\le \mathrm{C}_6(m/\log m)^{-2r/(2r+d)}\end{aligned} \tag{19}$$

with constants C_5 and C_6 depending only on M, d, c_0, and r.

Combining (18) and (19), we find that if $f_\rho \in \mathcal{F}^{(r,c_0)}$, then LSF achieves almost optimal learning rates for the best algorithms. Noting from [24], [31], [36], and [59] RLS, RLS with subsampling and other random sketch schemes can also achieve these optimal learning rates, we thus theoretically verify that LSF does not degenerate the learning performance of RLS.

IV. Related Work and Discussion

Studying the learning performance of RLS and its variants, such as distributed RLS, localized RLS, RLS with subsampling, and RLS with randomized sketches is a hot topic in the learning theory. Optimal learning rates of RLS have been verified in [7], [24], and [48] under some capacity assumptions on the kernel and regularity assumption on the regression function. The regularity assumption is described via an integral operator $L_K(f) := \int_{\mathcal{X}} K_x f(x) d\rho_X$ from $L^2_{\rho_X}$ to $L^2_{\rho_X}$ depending on the marginal distribution ρ_X, making these results not distribution free. Almost optimal learning rates of RLS under the distribution-free setting were derived in [65] for polynomial kernels and [17] for Gaussian kernels. Optimal learning rates for distributed RLS were originally presented in [64] by a matrix decomposition approach under certain eigenfunction assumptions, which were removed in [24] by a novel integral operator approach. The approach was then extended to a wider class of spectral regularization in [29]. Optimal learning rates for localized learning were first provided in [27] for Gaussian kernels and then established for general kernels in [50] under some eigenfunction assumptions. Optimal learning rates for the Nyström regularization were initially proved in [2] in a fixed design setting and then derived in [36] in the framework of the learning theory, provided the size of subsampling is not very small. Recently, a capacity-independent optimal learning rates for the Nyström regularization were derived in [22] under a more general source condition than [36]. Optimal learning rates for LRF were given in [37] when the number of features is not very small, under some regularity assumption associated with the kernel generated by random features. Leverage score-based sampling was introduced in [38], where the efficiency of LRF was improved. The stochastic gradient method was studied in [9], which can further reduce the memory requirement of LRF. Optimal learning rates for learning with randomized sketches were given in [60] with certain assumptions on the sketch matrix. By using the circulant matrix, a more scalable sketch RLS method was developed in [61], which keeps the optimal convergence rate. Computationally, the above methods can reduce the time complexity of RLS considerably from $\mathcal{O}(m^3)$ to $\mathcal{O}(n^2 m)$ or $\mathcal{O}(m^3/k^2)$, where n is the number of features for RLS with subsampling, and k is the number of partitions for distributed RLS and localized RLS. Fixed-budget kernel learning is another direction to reduce the computational cost of RLS. The main idea is to perform kernel learning in an online fashion, where the data were fed in sequentially. By keeping a fixed but small number of informative samples, the memory and time cost can be affordably controlled. Several novel methods were accordingly developed for classification and regression with different training and pruning strategies [14], [15], [53], [55]. It was also applied for subspace tracking by incorporating low-rank approximations [44]. Nevertheless, to the best of our knowledge, the optimal learning rate for the fixed-budget kernel regression remains an open problem. Besides the applications to RLS, optimal subsampling has been studied in the ordinary logistic regression [56] and generalized linear models [1]. An efficient variant of SVM for massive data was also developed in [62]. The method is based on a selected feature mapping on polynomial kernels. An alternating direction method of the multipliers algorithm and the optimal learning rate were derived.

Theoretically, we study optimal learning rates for LSF in the distribution-free setting, which is different from the results in [7], [24], [27], [36], [37], and [64]. However, Corollary 2 involves an additional logarithmic term making the learning rate be almost optimal and the LSF estimator needs to be clipped. Our technique novelties include a tight approximation error estimate based on Taylor's formula and an oracle inequality for ERM. The results and analysis in this article are in the flavor of [25] and [23], which focused on establishing almost optimal learning rates for ERM on neural networks (NNs) and radial basis function networks (RBFNs), respectively. Since

the sets of NNs and RBFNs studied in [23] and [25] are nonlinear, it is difficult to numerically deduce estimators achieving these optimal learning rates, which inevitably hinders their use in practice. Fortunately, LSF stated in Algorithm 1 only requires a linear least-squares problem that can be solved via the pseudoinverse technique.

We then illustrate the novelty of LSF by explaining its working mechanism. Distributed learning in [64] and localized learning in [27] can boil down to a divide-and-conquer strategy in tackling massive data. The difference is that distributed learning divides the training data into different local processors while localized learning divides the data into different partitions of the input space. Differently, RLS with subsampling drives a direction from low-rank approximations of kernel matrices without introducing any divide-and-conquer strategy. In particular, for algorithm (3), it was shown in the literature that the centers can be independently drawn from the inputs of the sample according to the uniform distribution [58], randomly drawn from the inputs of the sample with a judiciously chosen nonuniform importance sampling distribution [63], drawn from the data according to some approximate leverage scores [16], or drawn according to the uniform distribution independent of the data [33]. In all these approaches, there are totally three tunable parameters in algorithm (3): 1) kernel parameter; 2) regularization parameter; and 3) the number of features.[4] The kernel parameter depicts the capacity of the unit ball of the corresponding RKHS. The regularization parameter balances the bias (approximation error) and variance (sample error). The number of features reflects the computational complexity and generalization ability, simultaneously. Algorithm 1 removes the regularization parameter by noting its same role as the number of features in generalization and parameterizes n to control the capacity of $\overline{\mathcal{H}}_{K,n,\delta}$. From (3) to (12), the regularization parameter λ is removed but additional requirements of the features are imposed. Comparing with (1), (12) transforms a continuous parameter λ to a discrete parameter n. The main advantage of this transformation is that the computational complexity of (12), $\mathcal{O}(n^2m)$, varies with the parameter n, while that of (1), $\mathcal{O}(m^3)$, is independent of λ, implying fewer computations for LSF since n is usually much smaller than m.

Furthermore, it is notable that the estimator $f_{\mathbf{z},\lambda}$ defined by (1) can be viewed as the maximum *a-posteriori* (MAP) function based on the Gaussian process [35, Sec. 6.2.3]. To reduce the computational burden of RLS for massive data, a large literature [12], [41], [43], [47], [52] assumed that the Gaussian process regression models are *sparse*, which mean that the estimator is constructed in terms of $n \ll m$ latent variables (also called *inducing variables*). The aforementioned inducing variables method leverages a subset of the training inputs or auxiliary pseudopoints [32], [47] as inducing variables, and then reduces the computational complexities of RLS to $\mathcal{O}(n^2m)$. The advantage of inducing variable methods is that it can be generally applied to other Gaussian process models (such as the latent Gaussian process model [6] and deep Gaussian process models [39]). The proposed LSF theory in this article can be seen as a novel inducing variable method to solve the Gaussian process regression model and provide a sufficient condition on how to generate inducing variables with a theoretical guarantee. The main novelty of the LSF theory is generalization ability verifications in theory and fewer parameters in practice.

To facilitate the use of LSF, it is necessary to discuss the nonsingularity of matrix $A_{m,n}$. Theoretically speaking, there is not any positive-definite requirement for K^δ in Algorithm 1. However, the solvability of (12) in practice needs the nonsingularity of matrix $A_{m,n}$. With the positive-definite assumption, the nonsingularity can be verified by using the standard technique in the approximation theory [57, Ch. 12]. In particular, for some specified ϕ, such as the exponential function, multiquadratics, and inverse multiquadratics, the nonsingularity of matrix $A_{m,n}$ can be easily derived from [57, Ch. 12]. As we consider arbitrary ϕ and arbitrary set of centers satisfying the restrictions presented in Algorithm 1, it is difficult to develop a uniform framework to analyze the nonsingularity of $A_{m,n}$. Facing with the singular matrix, we suggest to use a regularization version of (12) with fixed but very small regularization parameter $\lambda_0 = 10^{-10}$. That is, we suggest to consider the algorithm

$$f^r_{\mathbf{z},n,\delta} = \arg\min_{f\in\overline{\mathcal{H}}_{K,n,\delta}} \left\{ \frac{1}{m}\sum_{i=1}^{m}(f(x_i)-y_i)^2 + \lambda_0 \sum_{j=1}^{n} c_j^2 \right\} \quad (20)$$

for non positive-definite kernels, where c_j is given in (11). It should be highlighted that the difference of (3) and (20) is that (3) treats λ as a parameter but (20) sets $\lambda_0 = 10^{-10}$. There are not additional computations by adding such a fixed regularizer. Since λ_0 is extremely small, the almost optimal learning rates for (20) can be guaranteed by using the same methods presented in Appendix C. Of course, adding a regularizer like (20) is not appealing, a preferable way is to search some sufficient conditions for kernels and centers to guarantee the nonsingularity of the matrix $(K(x_i,\alpha_j))_{i=1,j=1}^{m,n}$ and to satisfy the conditions in Algorithm 1. But it is quite challenging. We will keep on studying this topic in future works.

In our theoretical results, the learning performance of LSF is independent of the center generating mechanism, provided the regression function is smooth, the embedding space is $\mathbb{B}^d$ (or any compact sets topologically homeomorphic to $\mathbb{B}^d$), and the center set contains a uniqueness set for $\mathcal{Q}_{2s}^d$ with some $s \in \mathbb{N}_0$. Our simulation studies in Section V-A verify this assertion. However, in real data applications, the situation that f_ρ is not smooth or $\mathcal{X}$ is a lower dimensional manifold embedded into $\mathbb{R}^d$ occurs frequently, just like the real data experiments shown in Section V-C. Thus, it would be interesting to select a feasible and efficient center generating mechanism to tackle real data, when assumptions in this article are not satisfied.

To finalize the discussion, we should mention that we only adopt the generalization and computational burden to measure the performance of learning algorithms. With this premise, we develop some sufficient conditions on selecting features such that the regularization term in RLS with subsampling (3) can

[4] Although the results presented in [36] and [37] hold for a number of n, the lower bound of n depends on the regularity assumption and confidence level, making it be a potential parameter in practice.

be removed. This does not mean that the regularization term is useless in kernel methods. In fact, although different regularizers may reach the same optimal learning rates [23], [48] under Assumption 1, it is well known that different regularizers lead to different properties of the deduced estimators, including the sparseness, interpretability, robustness, and so on.

V. Numerical Studies

In this section, we present toy simulations, UCI standard data verifications, and a Million Song data application to assess the performance of LSF. All experiments were conducted in MATLAB R2017b on a workstation with Intel/Xeon E5-2697 v3 2.6-GHz CPU and 128-GB RAM.[5] Without special declarations, the kernel adopted in this section is the Gaussian kernel $K^\delta(x, x') = \exp\{-\delta\|x - x'\|^2\}$.

A. Toy Simulations

This series of simulations were designed to support the correctness of the theoretical results in Section III and to demonstrate the learning performance of LSF. For this purpose, the regression function f_ρ is supposed to be known and given by

$$f_\rho(x) = \left(1 - \|x\|_2^2\right)_+^6 \left(35\|x\|_2^2 + 18\|x\|_2^2 + 3\right),\ x \in [-1, 1]^2$$

where $a_+ = \max\{a, 0\}$. It is easy to check that $f_\rho \in \mathrm{Lip}^{(4,c_0)}$ for some $0 < c_0 < \infty$ and $f_\rho \notin \mathrm{Lip}^{(5,c_1)}$ for all $0 < c_1 < \infty$. We generated the training sample $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^m$, where $\{x_i\}_{i=1}^m$ are drawn independently according to the uniform distribution from $[-1, 1]^2$, $y_i = f_\rho(x_i) + \epsilon_i$, and $\varepsilon_i \sim N(0, 0.1)$ is the Gaussian noise. The generalization abilities of an algorithm were tested by applying the resultant estimator to the test sample $\mathbf{z}_{test} = \{(x_i^{(t)}, y_i^{(t)})\}_{i=1}^{m_1}$, which was generated similarly to $\mathbf{z}$ but with a promise that $y_i^{(t)} = f_\rho(x_i^{(t)})$.

The simulations are done for six purposes. In the first simulation, we devote to justifying the phase-transition phenomenon with respect to the size of the sample and tolerance of the test error. The second simulation is to numerically exhibit that there is a phase-transition phenomenon concerning the number of features and tolerance. In the third simulation, we study the parameter selection of LSF. The fourth one aims at revealing the influence of the center generating mechanism in LSF. In the fifth one, we compare LSF with LRF, Nyström regularization, sub-Gaussian sketches (G-sketch), and RLS to show the pros and cons of LSF. The last simulation concerns the role of the kernel via comparing LSF with the aforementioned algorithms.

In the first simulation, let $\{\alpha_i\}_{i=1}^n$ be a set of points in $[-1, 1]^2$ generated by linearly mapping the Sobol sequences to $[-1, 1]^2$. The generalization error is measured by the root-mean-square error (RMSE) with RMSE:= $\sqrt{(1/1000)\sum_{i=1}^{1000}(f_{\mathbf{z}}(x_i^{(t)}) - y_i^{(t)})^2}$ for some estimator $f_{\mathbf{z}}$. Given a certain pair of tolerance and sample size, we repeat LSF 50 times, and a run is labeled as a success if the RMSE is smaller than the tolerance. We plot 2-D figures to reflect the probability of success, where the x-axis and the y-axis represent the sample size and tolerance, respectively. The color from red to blue represents the success rate from 0 to 1. Theorem 2 implies a sharp phase-transition phenomenon of LSF in the sense that the probability of success changes dramatically when the tolerance is in the critical interval $[\varepsilon_m^-, \varepsilon_m^+]$. Fig. 1 shows that such a phenomenon numerically exists. In the lower part of each figure of Fig. 1, the color is red, indicating that the probability when RMSE is smaller than the tolerance is approximately 0. Thus, if the number of samples is small, LSF cannot yield an estimator with very small tolerance, verifying (15) in Theorem 2. In the upper area, the probability of success is approximately 1, which verifies (16) in Theorem 2. Between these two areas, there exists a band, in which the color varies from red to blue dramatically, exhibiting the phase-transition phenomenon. It is shown that the phase-transition interval is extremely narrow and its width monotonously decreases with m, which verifies (14). All these results coincide with our theoretical assertions in Theorem 2.

The second simulation aims at numerically exhibiting the phase-transition phenomenon on the tolerance and the number of features (or kernel size). To be detailed, since the center set is required to contain a uniqueness set for $\mathcal{Q}_{2s}^d$ with some $s \geq 0$ and $T_s \sim s^d$, the learning performance of LSF changes dramatically from $n < T_{s_\varepsilon}$ to $n = T_{s_\varepsilon}$, where s_ε is the smallest s to achieve the accuracy ε. Furthermore, once $n \geq T_{s_\varepsilon}$, the learning performance cannot be essentially improved. Under this circumstance, there is a sharp phase-transition phenomenon of LSF concerning the tolerance and kernel size. That is, given a tolerance ε, the probability of success changes dramatically when the kernel size varies from $T_{s_\varepsilon} - 1$ to T_{s_ε}. For this purpose, we employed three simulations on different sample sizes at $m = 100, 250, 400$. The simulation results were shown in Fig. 2. In the left lower part of each figure in Fig. 2, the colors of all points are red, while in the right upper area, the color changes dramatically to blue when n varies in a very narrow range, exhibiting the phase-transition phenomenon. It should be noted that in the below area of all three figures, the colors are red. The reason is that we only use $m = 100, 250, 400$. We believe that the lower part will turn to blue when sufficiently many samples are given [see the bottom low of Fig. 4(g) for example]. The simulation results experimentally illustrate the rationality of introducing the uniqueness set to describe the centers of LSF.

In Algorithm 1, there are totally two parameters in LSF, the kernel parameter δ and the number of features n. In other words, we remove the regularization parameter λ in (3) to reduce the computational burden. An intuitive question is that whether it will bring additional difficulty in the parameter selection for LSF. This requires us to illustrate that cross-validation can effectively select the appropriate parameters in LSF. For this purpose, we record RMSE of LSF on each parameter pair (δ, n) via 50 times trials to study the role of parameters. In Fig. 3(a), we employ the approach in [10, Fig. 3] to study the shape of contours. The colors from yellow to blue denote the testing error from bad to good, and the best combinations are highlighted by red color. Fig. 3(a) shows that there exists a large connected range of good parameters

[5]Code is available at https://sites.google.com/site/jianfang86/lsf.

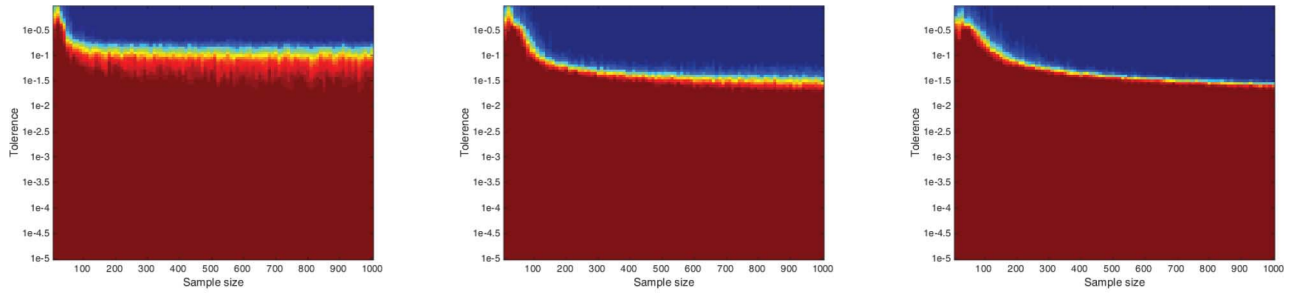


Fig. 1. Phase-transition phenomenon with respect to tolerance ε and sample size m given kernel size. (a) $n = 20$. (b) $n = 40$. (c) $n = 60$.

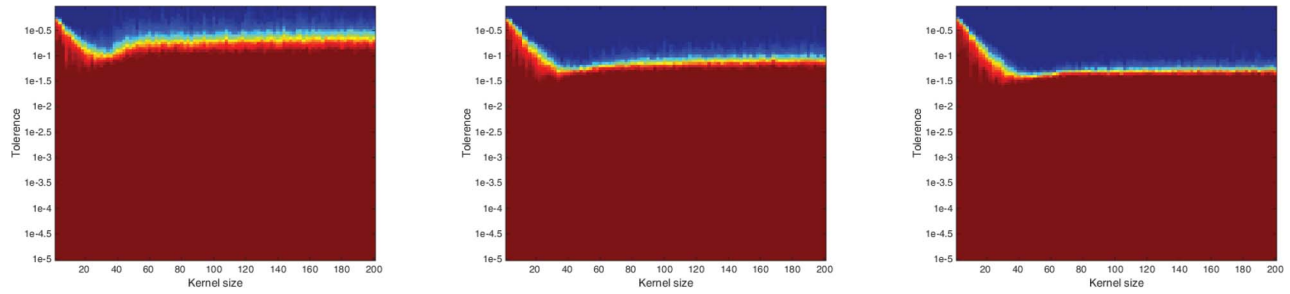


Fig. 2. Phase-transition phenomenon with respect to tolerance ε and kernel size n given sample size. (a) $m = 100$. (b) $m = 250$. (c) $m = 400$.

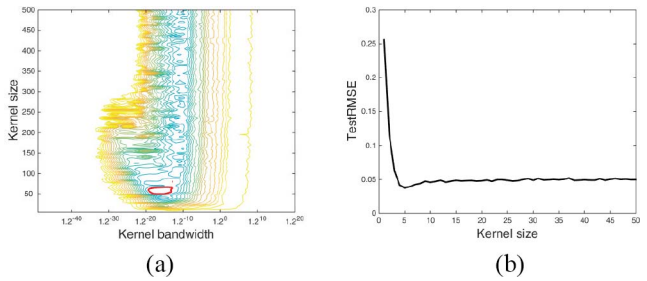


Fig. 3. (a) Contour plot of the test error when $m = 500$ with respect to the number of features and kernel parameters. (b) Testing error with respect to the number of features when $m = 500$.

for LSF, which is similar to the phenomenon revealed for statistical regularization [10, Fig. 3]. This means that the optimal parameters can be easily selected by cross-validation according to the analysis in [10]. Since we use the number of features to balance the bias and variance, we also study the relationship between the generalization error and the number of features. In Fig. 3(b), we record the testing error as a function of the number of features to illustrate the reasonability of introducing the computational regularization. Here, the kernel parameter is selected according to the least testing error such that the role of number features can be fully revealed. The trends exhibiting in Fig. 3(b) verify Corollary 1 by showing a tradeoff between bias and variance and exhibiting an optimum value of the number of features. It behaves similarly as the statistical regularization [10] and thus verifies the reasonability of using a single computational regularization in the learning process.

In the fourth simulation, we compare the performance of LSF with three different center generating mechanisms. In the first case, the centers are generated by the Sobol sequences. In the second case, the centers are drawn independently according to the uniform distribution from $[-1, 1]^2$. In the last case, the centers are drawn independently according to the uniform distribution from $\{x_i\}_{i=1}^m$. We try three groups of simulations by fixing one of ε, m, and n, and varying the other two. The simulation results are reported in Fig. 4, showing that LSF with aforementioned center generating mechanisms possesses almost the same testing errors. These results numerically verify our assertions in Theorem 2 and Corollary 2 that optimal learning rates hold for the arbitrary set of centers, provided it contains a uniqueness set when f_ρ is smooth.

In the fifth simulation, we compare the learning performances of LSF with the benchmark learning strategy RLS. To illustrate the rationality of removing the regularization term, we also compare LSF with the Nyström regularization and LRF. Finally, sub-Gaussian sketches without statistical regularization are included to test the differences of subsampling strategies. In this simulation, MSF denotes the mean number of selected features via 50 times trials and it reflects the testing time of estimators. TrainMT denotes the mean training time. Furthermore, LSF(S), LSF(R), and LSF (D) denote LSF with centers generated by Sobol sequences, randomly drawn from $\mathcal{X}$, and randomly drawn from the data $\{x_i\}_{i=1}^m$. LRF(S), LRF(R), and LRF(D) denote algorithm (3) with centers generated by the Sobol sequences, randomly drawn from $\mathcal{X}$, and randomly drawn from the data $\{x_i\}_{i=1}^m$. It is easy to see that LRF(R) coincides with LRF and LRF(D) is the plain Nyström regularization. For the sake of simplicity, we will use LRF to denote both LRF and Nyström regularization in the rest of this article. All parameters are selected via ten-fold cross-validation. In implementing LSF and G-Sketch, we set n and

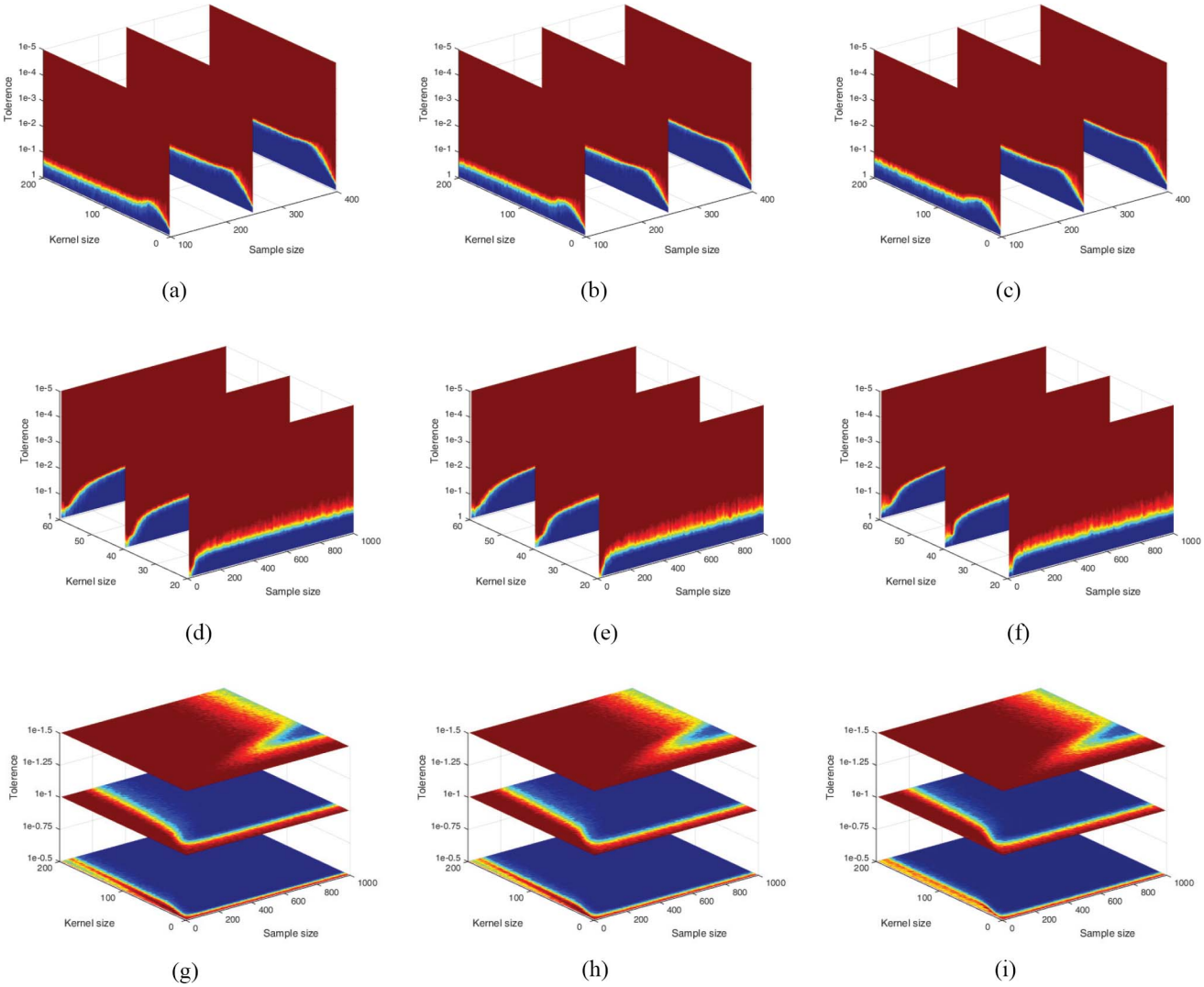


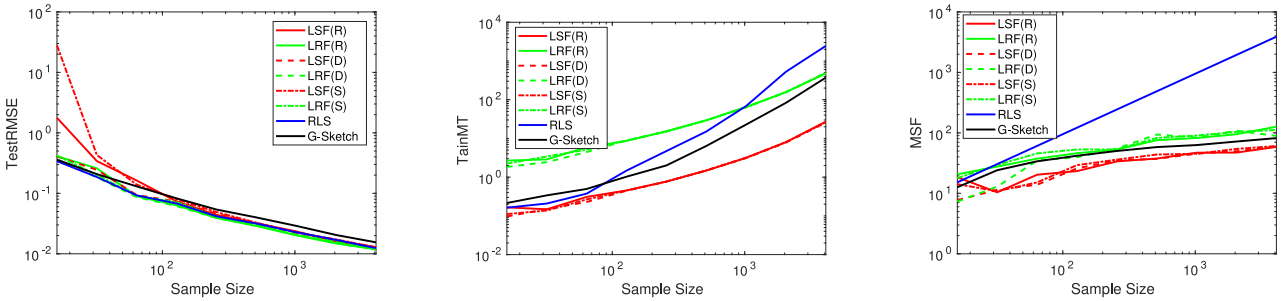
Fig. 4. LSF with different center generation mechanisms. (a), (d), and (g) LSF with Sobol centers; (b), (e), and (h) LSF with random centers (independent of samples); and (c), (f), and (i) LSF with random centers (depending on samples).

δ as two parameters. In implementing algorithm (3), we set n, δ, and λ as three parameters, and the theoretical results in [36] and [37] showed that optimal learning rates hold for not very small n. We highlight that we pursue the minimal RMSE for all mentioned algorithms to verify whether the learning performance of RLS is degraded. We report the simulation results in Fig. 5.

It can be found in Fig. 5 that if the size of samples is small, LSF performs worse than LRF and RLS, which means that removing the penalty brings a risk of degrading the generalization capability. However, if the size of the sample is large enough, which is the main purpose of our study in this article, the test errors of almost all the mentioned algorithms are the same, except that G-Sketch is slightly worse. But the training time of LSF is smaller than the other algorithms. It is also worthy to note that LSF can always lead to less testing time than LRF (reflected by the number of selected features). This phenomenon may imply that statistical regularization and computational regularization do play partly overlapping roles, which hinders to fully exploit the potentials of computational regularization in reducing the complexity of the predictors.

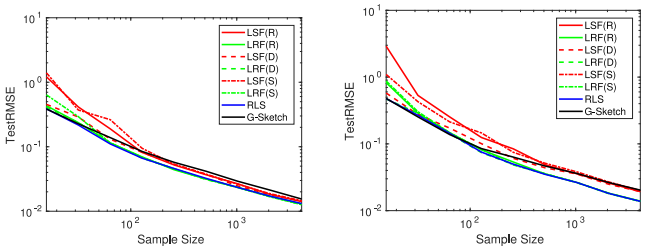
In the last simulation, we aim at studying the role of kernels, since the previous simulations only study the performance of LSF with the Gaussian kernel. In Algorithm 1, we impose three conditions on the kernel: 1) radial; 2) analytic; and 3) the values of derivatives on 0 are not vanished. It should be illustrated that they are somewhat necessary, at least not too loose. To this end, we adopt two additional kernels into comparison: 1) the inverse multiquadratic kernel, $K^{\delta}_{im}(x, x') = [1/(\sqrt{\delta\|x - x\|^2 + 1})]$, with link function $\phi_{im}(t) := [1/(\sqrt{1 + t^2})]$ and 2) the thin-plate spline kernel, $K^{\delta}_{ts}(x, x') = \delta\|x - y\|^2 \log \delta\|x - y\|^2$, with link function $\phi_{ts}(t) = t^2 \log t^2$. It is easy to check that K^{δ}_{im} satisfies the aforementioned restrictions on the kernel but K^{δ}_{ts} fails since $\phi_{ts}(0) = 0$. The simulation results are exhibited in Figs. 6 and 7.

It can be found in Fig. 6(a) that the testing errors of LSF with the inverse multiquadratic kernel perform similar as LSF with the Gaussian kernel in the sense that they achieve similar RMSE as RLS and LRF, provided the size of data is large enough. However, for the thin-plate spline kernel, the performance of LSF is always worse than the others. This



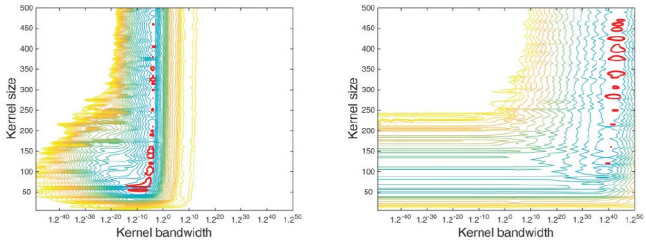
(a) (b) (c)

Fig. 5. Comparisons of LSF with RLS, Nyström regularization, and randomized sketches on different sizes of samples in terms of (a) test error, (b) training time, and (c) number of features.



(a) (b)

Fig. 6. Comparisons of LSF with RLS, Nyström regularization, and randomized sketches on different sizes of samples in terms of test error for (a) inverse multiquadratic kernel and (b) thin-plate spline kernel.



(a) (b)

Fig. 7. Contour plot of the test error when $m = 500$ with respect to the number of features and kernel parameters for (a) inverse multiquadratic kernel and (b) thin-plate spline kernel.

verifies the necessity of selecting kernels (or features). In addition, Fig. 7 presents the reason why the degeneration happens. Without the aforementioned assumptions, it is difficult to derive a similar optimal approximation error estimate (with respect to the linear width theory) like Theorem 4. Thus, it requires many more features (larger n) to achieve the small approximation error, just as Fig. 6(b) purposes to show.

A large number of features inevitable lead to large variance and consequently bad generalization ability according to the bias–variance tradeoff principle [11]. All these results show that LSF is a feasible and efficient learning scheme for tackling massive data.

B. UCI Standard Data

We further apply LSF(S), LSF(R), LSF(D), LRF(S), LRF(R), LRF(D), G-Sketch, and RLS on a family of UCI standard datasets. The datasets consist of five real-world benchmark problems covering different fields from the UCI machine-learning repository,[6] with the training and testing samples drawn as in Table I.

TABLE I
SPECIFICATION OF REAL-WORLD BENCHMARK DATASETS

Data sets	Train Number	Test Number	#Attributes
Stock	760	190	9
Abalone	3341	836	8
Bank8FM	3599	900	8
Delta_ailerons	5703	1426	5
Delta_Elevators	7613	1904	6

In these experiments, we use ten-fold cross-validation to select parameters involved in each algorithm. Then, we randomly select the training samples and evaluate the TestRMSE, TrainMT, and MSF of each estimator for 50 times. The UCI data results are listed in Table II, in which the standard deviation for each measurement is in parentheses. As far as the testing error is concerned, we find that LSF, LRF, and G-Sketch perform slightly worse than RLS on the Stock data and Abalone data (relatively small size) and are comparable with RLS on the Bank8FM data, Delta_ailerons data, and Delta_Elevators data (relatively large size). Furthermore, LSF performs similarly as LRF and G-Sketch on almost all the mentioned data. It should be noted that the center generating mechanisms does not affect the learning performance of LRF and LSF very much, which verifies our theory on the uniqueness set. All these results demonstrate the generalization ability of LSF presented in Section III, saying that LSF can reach the almost optimal learning rates of RLS given enough samples. However, concerning the computational burden, including the training time and testing time, LSF is much better than RLS, G-Sketch, and LRF on all datasets. This verifies our theoretical assertions and shows the power of LSF in regression problems.

C. Massive Data Experiments

In this part, we focus on the Million Song data [4] that consist of 463 715 training examples and 51 630 testing examples. The Million Song data describe a learning task of predicting the year in which a song is released. Each example is a song released between 1922 and 2011, and the

[6] http://archive.ics.uci.edu/ml/datasets.html

TABLE II
COMPARISONS ON UCI DATASETS

	LSF(S)	LRF(S)	LSF(R)	LRF(R)	LSF(D)	LRF(D)	RLS	G-Sketch
Stock data								
TestRMSE	0.030(0.003)	0.031(0.002)	0.028(0.002)	0.030(0.002)	0.030(0.002)	0.031(0.002)	0.027(0.002)	0.028(0.003)
TrainMT	4.2	75.2	3.5	72.3	3.8	75.5	132.7	14.0
MSF	206.1(43.3)	241.2(22.2)	215.0(36.9)	239.2(23.5)	209.2(43.5)	240.0(21.2)	760(0)	243.3(18.7)
Abalone data								
TestRMSE	0.078(0.007)	0.076(0.003)	0.076(0.003)	0.076(0.003)	0.078(0.004)	0.076(0.003)	0.075(0.003)	0.076(0.004)
TrainMT	42.6	833.9	38.2	766.0	42.0	835.0	7420.1	267.8
MSF	50.6(15.6)	331.0(112.1)	69.2(21.0)	432.4(90.1)	49.5(18.2)	353.4(131.3)	3340(0)	71.5(17.3)
Bank8FM data								
TestRMSE	0.043(0.003)	0.042(0.001)	0.043(0.001)	0.044(0.001)	0.044(0.003)	0.042(0.001)	0.042(0.001)	0.043(0.001)
TrainMT	45.4	912.2	45.9	928.7	45.3	907.9	11019.6	397.6
MSF	175.73(31.61)	501.22(39.3)	248.84(41.65)	501.22(39.3)	179.04(36.05)	509.61(29.05)	3590(0)	299.3(59.2)
Delta ailerons data								
TestRMSE	0.039(0.001)	0.038(0.001)	0.039(0.001)	0.039(0.001)	0.039(0.001)	0.038(0.001)	0.039(0.001)	0.039(0.001)
TrainMT	103.3	2074.3	59.8	1198.2	104.7	2082.5	14154.7	647.34
MSF	67.0(16.4)	399.2(162.4)	102.5(59.5)	543.4(129.8)	65.3(18.3)	399.8(174.1)	5700(0)	73.7(24.5)
Delta elevators data								
TestRMSE	0.053(0.001)	0.053(0.001)	0.053(0.001)	0.053(0.001)	0.053(0.001)	0.053(0.001)	0.053(0.001)	0.053(0.001)
TrainMT	173.3	3439.7	110.5	2193.1	173.0	3433.7	46073.8	1303.4
MSF	81.3(16.8)	501.5(167.9)	88.9(40.2)	685.4(99.1)	84.6(15.8)	535.7(161.5)	7610(0)	78.4(17.6)

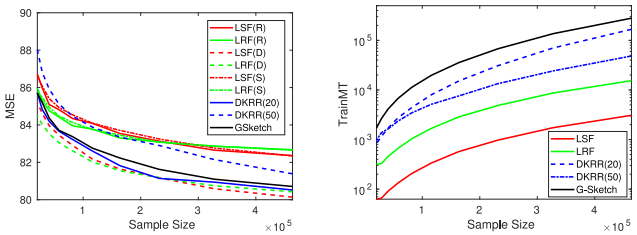
song is represented as a vector of timbre information computed about the song. Each sample point consists of a pair $(x_i, y_i) \in [0,1]^d \times [1922, 2011]$ with $d = 90$. Since RLS requires $\mathcal{O}(m^3)$ floating computations and cannot tackle this dataset, we employ the distributed RLS (DRLS) due to its excellent performance to tackle this dataset [64].

Similar as [64], we give a feature weight vector $W = (w_1, \ldots, w_d)^\top$ for setting $x_{ij} := w_j x_{ij}$ and choose $w_j = 1$ if $j \le 12$ and $w_j = 0.2$ if $12 < j \le 90$. Besides full data, we also compare the performance by using subsets drawn uniformly at random from all the available training instances. We use ten-fold cross-validation to select the parameters for LSF, LRF, G-Sketch, and DRLS. Then, we implement each algorithm independently ten times and record the TestMSE, TrainMT, and MSF.

We compare LSF equipped with three center generating mechanisms with LRF, G-Sketch, and DRLS running on 20 and 50 local processors, respectively. The experimental results are reported in Table III using the full training data and Fig. 8 using subsets. It can be found that in this experiment, the center generating mechanism plays a crucial role in LSF learning as well as LRF. We find that LSF and LRF with centers randomly drawn from the data are the best. The reason might be that the input space of this task is in a lower dimensional manifold embodied into $[0,1]^{90}$ and LSF with centers generated in a data-dependent way can reflect this property. We thus recommend the use of data-dependent centers for LSF to tackle the high-dimensional learning tasks. It is shown in Table III and Fig. 8 that LSF performs similar as LRF and G-Sketch in the testing error but is better in training time and testing time. Compared with DRLS, we find that LSF can yield slightly smaller testing error with much less selected features. In short, with appropriately selected centers, LSF can handle the massive dataset successfully.

TABLE III
COMPARISONS ON MILLION SONG DATA

Method	TestRMSE	TrainMT	MSF
LSF(S)	83.36	3.1e3	2076
LRF(S)	83.37	1.5e4	3296
LSF(R)	84.19	3.1e3	2476
LRF(R)	84.20	1.5e4	3296
LSF(D)	80.62	3.1e3	3296
LRF(D)	80.81	1.5e4	3296
DRLS(20)	81.52	1.7e5	463,715
DRLS(50)	82.54	4.8e4	463,715
G-Sketch	80.73	2.7e5	3296



(a) (b)

Fig. 8. Comparisons of LSF with LRF, DRLS, and G-Sketch on subsets of Million Song data in terms of (a) test error and (b) training time.

VI. CONCLUSION

In this article, we developed a new learning scheme based on subsampling, called the learning with selected features (LSF), to reduce the computational burden for the classical kernel-based RLS algorithm. The contributions can be summarized as the following four aspects.

1) The uniqueness set was borrowed from [26] to describe the center generating mechanism of LSF.
2) The regularization term in RLS with subsampling was removed by imposing additional restrictions to the kernels and features to reduce the computational burden.
3) Almost optimal learning rates for LSF were derived in a distribution-free setting.
4) Numerical experiments were conducted to verify our theoretical assertions.

We concluded this article with the sober note: there is still much room for improvement, such as the nonsingularity

of matrix $(K(x_i, \alpha_j))_{i=1,j=1}^{m,n}$ and the center generating mechanism discussed in Section IV. We will keep studying these interesting topics and report the progress in a future study.

APPENDIX A
CONSTRUCTION OF ORTHONORMAL BASIS

Since the uniqueness set plays crucial in our analysis and its definition depends on an orthonormal basis for $\mathcal{P}_s^d$, we present in this part, the detailed construction of the orthonormal basis for polynomials. Let $G_s^\nu(t)$ be the Gegenbauer polynomial [54] with index ν. It is known that the family of polynomials $\{G_s^\nu\}_{s=0}^\infty$ is a complete orthogonal system in the weighted space $L^2(I, w_\nu)$ with $I := [-1, 1]$ and $w_\nu(t) := (1-t^2)^{\nu-(1/2)}$, that is

$$\int_I G_{s'}^\nu(t) G_s^\nu(t) w_\nu(t) dt = \begin{cases} 0, & s' \neq s \\ h_{s,\nu}, & s' = s \end{cases}$$

where

$$h_{s,\nu} = \frac{\pi^{1/2}(2\nu)_s \Gamma\left(\nu + \frac{1}{2}\right)}{(s+\nu)s!\Gamma(\nu)}$$

$$(a)_0 := 0, \ (a)_s := a(a+1)\cdots(a+s-1) = \frac{\Gamma(a+s)}{\Gamma(a)}.$$

Define

$$U_s := \left(h_{s,d/2}\right)^{-1/2} G_s^{d/2}, \ s = 0, 1, \ldots \tag{21}$$

Then, $\{U_s\}_{s=0}^\infty$ is a complete orthonormal system for the weighted L^2 space $L^2(I, w)$ with $w(t) := (1-t^2)^{[(d-1)/2]}$.

Let $\mathbb{S}^{d-1}$ be the unit sphere in $\mathbb{R}^d$. The class of all spherical harmonics (homogenous harmonic polynomials defined on $\mathbb{S}^{d-1}$) of degree k is denoted by $\mathbb{H}_k^{d-1}$, and the class of all spherical polynomials with total degrees $k \leq s$ is denoted by Π_s^{d-1}. It can be found in [54] that $\Pi_s^{d-1} = \bigoplus_{k=0}^s \mathbb{H}_k^{d-1}$. Since the dimension of $\mathbb{H}_k^{d-1}$ is given by

$$D_k^{d-1} := \dim \mathbb{H}_k^{d-1} = \begin{cases} \frac{2k+d-2}{k+d-2}\binom{k+d-2}{k}, & k \geq 1 \\ 1, & k = 0 \end{cases}$$

the dimension of Π_s^{d-1} is $\sum_{k=0}^s D_k^{d-1} = D_s^d \sim s^{d-1}$.

Let $\{Y_{k,l} : l = 1, \ldots, D_k^{d-1}\}$ be an arbitrary orthonormal system of $\mathbb{H}_k^{d-1}$. Define

$$P_{k,j,i}(x) = v_k \int_{\mathbb{S}^{d-1}} Y_{j,i}(\xi) U_k(x \cdot \xi) d\omega_{d-1}(\xi) \tag{22}$$

where $x \in \mathbb{B}^d$, $v_k := ([(k+1)_{d-1}]/[2(2\pi)^{d-1}])^{(1/2)}$ and $d\omega_{d-1}$ denotes the aero element of $\mathbb{S}^{d-1}$. Then, it follows from [26] that:

$$\left\{P_{k,j,i} : k = 0, \ldots, s, j = k, k-2, \ldots, \tau_k, i = 1, 2, \ldots, D_j^{d-1}\right\}$$

is an orthonormal basis for $\mathcal{P}_s^d$.

APPENDIX B
PROOF OF THEOREM 1

Define

$$V_{\ell,j,i}(\theta) = \int_{\mathbb{S}^{d-1}} (\theta \cdot \xi)^\ell Y_{j,i}(\xi) d\omega_{d-1}(\xi)$$

$$u_{\beta,k} = \int_{[-1,1]} t^\beta U_k(t) w(t) dt$$

$$\mathcal{F}_s = \left\{ f_{s',k,j,i}(\cdot) = \sum_{\ell=0}^{2s'} \binom{2s'}{\ell} u_{2s'-\ell,k} V_{\ell,j,i}(\cdot) : \right.$$
$$\left. s' = 0, 1, \ldots, s, (k,j,i) \in \mathbf{I}_s \right\}.$$

The following definition describes some singularity for Θ_s.

Definition 3: For arbitrary $s \in \mathbb{N}$, we call $\Theta_s = \{\theta_\ell \in \mathbb{B}^d : 1 \leq \ell \leq T_s\}$ as the nonsingular set for $\mathbf{I}_s$ if the matrix $(\sum_{s'=0}^s a_{s'} f_{s',k,j,i}(\theta_\ell))_{(k,j,i)\in\mathbf{I}_s, 1\leq\ell\leq T_s}$ is invertible for $a_{s'} = (\int_{\mathbb{S}^{d-1}} (\vec{e}\cdot\xi)^{2s'} d\omega_{d-1}(\xi))^{-1}$ and $\vec{e} = (0, \ldots, 0, 1)^T \in \mathbb{R}^d$.

To prove Theorem 1, we need three lemmas, which can be found in [26, Proposition 5.3], Lemma 5.1 and Lemma 5.2 of [26, Lemmas 5.1 and 5.2] and [3, Lemma 3.1], respectively.

Lemma 1: Θ_s is the uniqueness set for $\mathcal{Q}_{2s}^d$ if and only if it is the nonsingular set for $\mathbf{I}_s$.

Lemma 2: The elements in $\{g_{k,j,i}(\cdot) : (k,j,i) \in \mathbf{I}_s\}$ are linear independent.

Lemma 3: Let $P \in \mathcal{P}_s^d$. Then, its zero set $\mathcal{Z}(P) := \{x \in \mathbb{B}^d : P(x) = 0\}$ has Lebesgue measure 0.

Proof of Theorem 1: Due to Lemma 1, Θ_s is a uniqueness set for $\mathcal{Q}_{2s}^d$ if and only if it is a nonsingular set for $\mathbf{I}_s$. Then, it suffices to prove $\mathrm{Det}(\mathcal{M}_N) \neq 0$, where $\mathcal{M}_N := (g_{k,j,i}(\theta_\ell))_{(k,j,i)\in\mathbf{I}_s, 1\leq\ell\leq T_s}$ with $N := T_s$ and $\mathrm{Det}(\mathcal{M}_N)$ denotes the determinant of $\mathcal{M}_N$. Recalling (10), we have

$$\mathcal{B}_N = \left\{\Theta_s = (\theta_\ell)_{\ell=1}^N \subset \mathbb{B}^d : \mathrm{Det}(\mathcal{M}_N) = 0\right\}.$$

We then prove $\Omega(\mathcal{B}_N) = 0$ by induction, where $\Omega(A)$ denotes the Lebesgue measure of a set A. When $N = 1$ or $s = 0$, it follows from Lemmas 2 and 3 directly that $\Omega(\mathcal{B}_1) = 0$. Assume $\mathcal{M}_k$ $(1 \leq k \leq N-1)$ is invertible almost surely, that is, $\Omega(\mathcal{B}_k) = 0$. Denote $\mathcal{M}_{k+1} = (a_{i,j})_{i,j=1}^{k+1}$. Let $\mathbf{a}_\ell := (a_{\ell,1}, \ldots, a_{\ell,k})$, $1 \leq \ell \leq k$, be the ℓth row of $\mathcal{M}_k$ and $\mathbf{a}_{k+1} = (a_{k+1,1}, \ldots, a_{k+1,k})$. Since $\mathcal{M}_k$ is invertible almost surely and $\mathbf{a}_{k+1}$ is a k-dimensional vector, there exists a unique nonzero vector $\mathbf{b} := (b_i)_{i=1}^k$ such that

$$\mathbf{a}_{k+1} = b_1\mathbf{a}_1 + b_2\mathbf{a}_2 + \cdots + b_k\mathbf{a}_k$$

almost surely. By looking at the $(k+1)$th column of $\mathcal{M}_{k+1}$, we find that $\mathcal{M}_{k+1}$ is invertible if and only if

$$a_{k+1,k+1} \neq b_1 a_{1,k+1} + b_2 a_{2,k+1} + \cdots + b_k a_{k,k+1}. \tag{23}$$

It is obvious that every element in $\{a_{\beta,k+1}\}_{\beta=1}^k$ corresponds a $g_{l,j,i}(\theta_{k+1})$ with some $(l,j,i) \in \mathbf{I}_s$. Denoting by $\mathbf{J}_k$ the collection of all these (l,j,i)s, it follows from (23) that $\mathcal{M}_{k+1}$ is invertible if θ_{k+1} is not in the set:

$$\mathrm{C}_k := \left\{ x \in \mathbb{B}^d : g_{l_0,j_0,i_0}(x) = \sum_{(l,j,i)\in\mathbf{J}_k} b_{l,j,i} g_{l,j,i}(x) \right.$$
$$\left. \text{for every } (l_0, j_0, i_0) \notin \mathbf{J}_k \right\}.$$

Due to Lemma 2, elements in $(g_{l,j,i}(\cdot))_{(l,j,i)\in\mathbf{J}_k}$ are linear independent for $k \leq N-1$. We have g_{l_0,j_0,i_0} is not the zero polynomial. Noting that $g_{l,j,i}$ is a polynomial of degree at most

$2s$, we get that C_k is the zero set for some algebraic polynomial of degree at most $2s$. Hence, it follows from Lemma 3 that $\Omega(\mathrm{C}_k) = 0$. Noting further that $\mathcal{B}_{k+1}$ is contained in the set

$$\Big\{(x_1, \ldots, x_{k+1}) \in (\mathbb{B}^d)^{k+1} : x_{k+1} \in \mathrm{C}_k\Big\}.$$

Fubini's theorem shows that

$$\begin{aligned}\Omega(\mathcal{B}_{k+1}) &= \int_{(\mathbb{B}^d)^k} \left(\int_{\mathbb{B}^d} \chi_{\mathcal{B}_{k+1}} dx_{k+1}\right) dx_1 \cdots dx_k \\ &\le \int_{(\mathbb{B}^d)^k} \Omega(\mathrm{C}_k) dx_1 \cdots dx_k = 0\end{aligned}$$

where χ_A denotes the indicator function for some set A. Thus, $\Omega(\mathcal{B}_k) = 0$ for all $k = 1, 2, \ldots, N$. This completes the proof of Theorem 1. ∎

Appendix C
Proof of Results in Section III

To prove Theorem 2, we need an oracle inequality concerning the pseudodimension [21]. For any $t \in \mathbb{R}$, define $\mathrm{sgn}(t) \coloneqq \begin{cases} 1, & \text{if } t \ge 0, \\ -1, & \text{if } t < 0. \end{cases}$ If a vector $\mathbf{t} = (t_1, \ldots, t_n)$ belongs to $\mathbb{R}^n$, then we denote by $\mathrm{sgn}(\mathbf{t})$ the vector $(\mathrm{sgn}(t_1), \ldots, \mathrm{sgn}(t_n))$. The VC dimension [20] of a set $\mathcal{V}$ over $\mathcal{X}$, denoted by $\mathrm{VCdim}(\mathcal{V})$, is defined by the maximal natural number l such that there exists a collection $(\xi_1, \ldots, \xi_l)$ in $\mathcal{X}$ such that the cardinality of the sgn-vectors set

$$S = \{(\mathrm{sgn}(v(\xi_1)), \ldots, \mathrm{sgn}(v(\xi_l))) : v \in \mathcal{V}\}$$

equals 2^l, that is, the set S coincides with the set of all vertices of unit cube in $\mathbb{R}^l$. The quantity

$$Pdim(\mathcal{V}) \coloneqq \max_g \mathrm{VCdim}(\mathcal{V} + g)$$

is called the pseudodimension of set $\mathcal{V}$ over $\mathcal{X}$, where g runs all functions defined on $\mathcal{X}$ and $\mathcal{V} + g = \{v + g : v \in \mathcal{V}\}$. The following theorem presents an oracle inequality on empirical risk minimization on a set of functions with finite pseudodimension.

Theorem 3: Let $\mathcal{H}$ be a class of functions with pseudodimension $Pdim(\mathcal{H})$. Define

$$f_{\mathbf{z},\mathcal{H}} \coloneqq \arg\min_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} (f(x_i) - y_i)^2. \tag{24}$$

Then, for arbitrary $h \in \mathcal{H}$, there holds

$$\begin{aligned}&\mathrm{Prob}\Big\{\|\pi_M f_{\mathbf{z},\mathcal{H}} - f_\rho\|_\rho^2 > 2\|h - f_\rho\|_\rho^2 + \varepsilon\Big\} \\ &\le \exp\left\{c' Pdim(\mathcal{H}) \log\frac{16M^2}{\varepsilon} - \frac{3m\varepsilon}{32M^2}\right\} \\ &\quad + \exp\left\{-\frac{m\varepsilon^2}{32(3M + \|h\|_\infty)^2\Big(\|h - f_\rho\|_\rho^2 + \frac{1}{12}\varepsilon\Big)}\right\}\end{aligned}$$

where c' is an absolute positive constant.

The proof of Theorem 3 depends on two standard concentration inequalities in [65] and a relation between the pseudodimension and ε-covering number proved in [28]. We will present it in the supplementary material.

Based on Theorem 3, to prove Theorem 2, it also needs to quantify $\|h - f_\rho\|_\rho$ for $h \in \overline{H}_{K,n,\delta}$. Thus, we need the following approximation error estimates.

Theorem 4: Let Θ_{T_s} be a *uniqueness set* for $\mathcal{Q}_{2s}^d$ with some $s \ge 0$ and ϕ possess up to $s+1$ times bounded derivatives and satisfy $\phi^{(v)}(0) \ne 0$ for all $0 \le v \le s$. If $f \in \mathrm{Lip}^{(r,c_0)}$, then there exists a $\delta^* \in (0, 1/4)$ depending on s and $c_i \in \mathbb{R}$, $i = 1, \ldots, T_s$ such that for arbitrary $\delta \in (0, \delta^*]$ there holds

$$\left\| f - \sum_{i=1}^{T_s} c_i \phi\Big(\delta\|x - \theta_i\|^2\Big)\right\|_\infty \le \mathrm{C}_7 s^{-r} \tag{25}$$

where C_7 is a constant depending only on r, c_0, and d.

The proof of Theorem 4 depends on Taylor's formula [23] and thus requires the smoothness of the link function of the kernel. It also depends heavily on the radial function representation for polynomials located on the uniqueness set [26]. Therefore, it requires that the center sets contain a uniqueness set and the kernel to be radial. We give the detailed proof of Theorem 4 in the supplementary material to shorten the length of this article. Theorem 4 implies the following corollary directly.

Corollary 3: Under the condition of Theorem 4, there exists a $\delta^* \in (0, 1/4)$ depending on s and an $f_0 \in \overline{\mathcal{H}}_{K,n,\delta}$ such that for arbitrary $\delta \in (0, \delta^*]$ there holds

$$\|f - f_0\|_\infty \le \mathrm{C}_7 s^{-r}.$$

The following lemma, which was proved in [21, Th. 4], stating that the pseudodimension of arbitrary linear space is equal to its dimension.

Lemma 4: Let G be a k-dimensional vector space of functions from a set $\mathbb{B}^d$ into $\mathbb{R}$. Then, $Pdim(G) = k$.

With these helps, we proceed the proof of Theorem 2 as follows.

Proof of Theorem 2: The proof of (15) and the lower bound of (16) can be found in [25], it suffices to prove the upper bound for (16). Since LSF is a two-stage learning strategy whose hypothesis space is a fixed n-dimensional linear space in the first stage and ERM is implemented in the second stage, it follows from Lemma 4 and Theorem 3 that for arbitrary $h \in \overline{\mathcal{H}}_{K,n,\delta}$

$$\begin{aligned}&\mathrm{Prob}\Big\{\|\pi_M f_{\mathbf{z},n,\delta} - f_\rho\|_\rho^2 > 2\|h - f_\rho\|_\rho^2 + \varepsilon\Big\} \\ &\le \exp\left\{c' n \log\frac{16M^2}{\varepsilon} - \frac{3m\varepsilon}{32M^2}\right\} \\ &\quad + \exp\left\{-\frac{m\varepsilon^2}{32(3M + \|h\|_\infty)^2\Big(\|h - f_\rho\|_\rho^2 + \frac{1}{12}\varepsilon\Big)}\right\}.\end{aligned} \tag{26}$$

Under Assumption 1, it follows from Corollary 3 and $|f_\rho(x)| \le M$ that:

$$\|f_\rho - f_0\|_\rho^2 \le \|f_\rho - f_0\|_\infty^2 \le \mathrm{C}_7^2 s^{-2r} \tag{27}$$

and

$$\|f_0\|_\infty \le \|f_\rho\|_\infty + \|f_0 - f_\rho\|_\infty \le M + \mathrm{C}_7. \tag{28}$$

Setting $h = f_0$ and noting $n \sim s^d \sim \varepsilon^{-d/(2r)}$ with $\varepsilon \ge \mathrm{C}_2(m/\log m)^{-2r/(2r+d)} \ge \varepsilon_m^+$, we obtain $\|f_0 - f_\rho\|_\rho^2 \le \mathrm{C}_8\varepsilon$

and

$$\begin{aligned} n &\le \mathrm{C}_9\varepsilon^{-d/(2r)} \le \mathrm{C}_9\varepsilon\varepsilon^{-(2r+d)/(2r)} \\ &\le \mathrm{C}_9\mathrm{C}_2^{-(2r+d)/(2r)} m\varepsilon \log^{-1} m \end{aligned}$$

where C_9 and C_8 are constants depending only on C_7, M, and d. Plugging the above estimates into (26), we have

$$\begin{aligned} &\mathrm{Prob}\Big\{\big\|\pi_M f_{\mathbf{z},n,\delta} - f_\rho\big\|_\rho^2 > (2\mathrm{C}_8+1)\varepsilon\Big\} \\ &\le \mathrm{Prob}\Big\{\big\|\pi_M f_{\mathbf{z},n,\delta} - f_\rho\big\|_\rho^2 > 2\big\|h - f_\rho\big\|_\rho^2 + \varepsilon\Big\} \\ &\le \exp\left\{\frac{c'\mathrm{C}_9\mathrm{C}_2^{-(2r+d)/(2r)} m\varepsilon}{\log m}\log\frac{16M^2 m}{\mathrm{C}_2} - \frac{3m\varepsilon}{32M^2}\right\} \\ &\quad + \exp\left\{-\frac{m\varepsilon}{32\Big((4M+2\mathrm{C}_7)^2\mathrm{C}_8 + \frac{1}{12}(4M+2\mathrm{C}_7)^2\Big)}\right\}. \end{aligned}$$

Set

$$\mathrm{C}_2 > \max\left\{16M^2, \left(\frac{32M^2C'\mathrm{C}_9}{3}\right)^{2r/(2r+d)}\right\}.$$

Then, scaling $(\mathrm{C}_8+1)\varepsilon$ to ε, we have

$$\mathrm{Prob}\Big\{\big\|\pi_M f_{\mathbf{z},n,\delta} - f_\rho\big\|_\rho^2 > \varepsilon\Big\} \le \exp\{-\mathrm{C}_4 m\varepsilon\}$$

where C_4 is a constant depending only on C_8, C_9, C_2, C_7, and M. The proof of Theorem 2 is completed. ■

Proof of Corollary 1: If $\varepsilon \ge (1/m)$, it follows from (26) with $h = f_0$, (27), and (28) that:

$$\begin{aligned} &\mathrm{Prob}\Big\{\big\|\pi_M f_{\mathbf{z},n,\delta} - f_\rho\big\|_\rho^2 > \varepsilon + 2\big\|h - f_\rho\big\|_\rho^2\Big\} \\ &\le \exp\left\{c' n\log(16M^2 m) - \frac{3m\varepsilon}{32M^2}\right\} \\ &\quad + \exp\left\{-\frac{3m\varepsilon}{56(4M+\mathrm{C}_7)^2}\right\} \\ &\le \Big\{\exp\{c' n\log(16M^2 m)\} + 1\Big\}\exp\left\{-\frac{3m\varepsilon}{56(4M+\mathrm{C}_7)^2}\right\}. \end{aligned} \tag{29}$$

Setting

$$\Big\{\exp\Big\{c' n\log(16M^2 m)\Big\} + 1\Big\}\exp\left\{-\frac{3m\varepsilon}{56(4M+\mathrm{C}_7)^2}\right\} = \frac{\delta}{3}$$

we get

$$\varepsilon = \frac{56(4M+\mathrm{C}_7)^2}{3m}\left(2(c'+1)n\log(16M^2 m) + \log\frac{3}{\delta}\right) \ge \frac{1}{m}.$$

Then, it follows from (29), (27), and $n \sim s^d$ that with confidence at least $1-\delta$, there holds:

$$\big\|\pi_M f_{\mathbf{z},n,\delta} - f_\rho\big\|_\rho^2 \le \tilde{C}\left(\frac{n\log m}{m} + n^{-\frac{2r}{d}}\right)\log\frac{3}{\delta}.$$

This completes the proof of Corollary 1. ■

Proof of Corollary 2: The lower bound of (19) can be found in [20, Th. 3.2]. It suffices to prove the upper bound of (19). We apply the formula

$$E[\xi] = \int_0^\infty \mathrm{Prob}[\xi > t]dt \tag{30}$$

with $\xi = \|\pi_M f_{\mathbf{z},n,\delta} - f_\rho\|_\rho^2$ to prove Corollary 2. Based on Theorem 2, we have

$$\begin{aligned} E\Big[\big\|\pi_M f_{\mathbf{z},n,\delta} - f_\rho\big\|_\rho^2\Big] &\le \varepsilon_m^+ + \int_0^\infty e^{-\mathrm{C}_4 m\varepsilon}d\varepsilon \\ &\le \mathrm{C}_6(m/\log m)^{-2r/(2r+d)} \end{aligned}$$

where $\mathrm{C}_6 = \mathrm{C}_2 + \mathrm{C}_4^{-1}$. This completes the proof. ■

ACKNOWLEDGMENT

The authors would like to thank the Action Editor and three anonymous referees for their constructive suggestions.

REFERENCES

[1] M. Ai, J. Yu, H. Zhang, and H. Wang, "Optimal subsampling algorithms for big data generalized linear models," 2018. [Online]. Available: arXiv:1806.06761.
[2] F. R. Bach, "Sharp analysis of low-rank kernel matrix approximations," in *Proc. Conf. Learn. Theory*, 2013, pp. 185–209.
[3] R. F. Bass and K. Gröchenig, "Random sampling of multivariate trigonometric polynomials," *SIAM J. Math. Anal.*, vol. 36, no. 3, pp. 773–795, 2005.
[4] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, vol. 2, 2011, p. 10.
[5] R. Blundell and A. Duncan, "Kernel regression in empirical microeconomics," *J. Human Resources*, vol. 33, no. 1, pp. 62–87, 1998.
[6] E. V. Bonilla, K. Krauth, and A. Dezfouli, "Generic inference in latent Gaussian process models," 2016. [Online]. Available: arXiv:1609.00577.
[7] A. Caponnetto and E. De Vito, "Optimal rates for the regularized least-squares algorithm," *Found. Comput. Math.*, vol. 7, no. 3, pp. 331–368, 2007.
[8] A. Caponnetto and Y. Yao, "Cross-validation based adaptation for regularization operators in learning theory," *Anal. Appl.*, vol. 8, no. 2, pp. 161–183, 2010.
[9] L. Carratino, A. Rudi, and L. Rosasco, "Learning with SGD and random features," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10192–10203.
[10] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.
[11] F. Cucker and D. X. Zhou, *Learning Theory: An Approximation Theory Viewpoint*, vol. 24. Cambridge, U.K: Cambridge Univ. Press, 2007.
[12] L. Csató and M. Opper, "Sparse on-line Gaussian processes," *Neural Comput.*, vol. 14, no. 3, pp. 641–668, 2002.
[13] X. Chang, S. B. Lin, and D. X. Zhou, "Distributed semi-supervised learning with kernel ridge regression," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1493–1514, 2017.
[14] O. Dekel, S. Shalev-Shwartz, and Y. Singer, "The Forgetron: A kernel-based perceptron on a fixed budget," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 259–266.
[15] O. Dekel and Y. Singer, "Support vector machines on a budget," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 345–352.
[16] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff, "Fast approximation of matrix coherence and statistical leverage," *J. Mach. Learn. Res.*, vol. 13, pp. 3475–3506, Dec. 2012.
[17] M. Eberts and I. Steinwart, "Optimal regression rates for SVMs using Gaussian kernels," *Electron. J. Stat.*, vol. 7, pp. 1–42, Jan. 2013.
[18] T. Evgeniou, M. Pontil, and T. A. Poggio, "Regularization networks and support vector machines," *Adv Comput. Math.*, vol. 13, no. 1, pp. 1–50, 2000.
[19] A. Gittens and M. W. Mahoney, "Revisiting the Nyström method for improved large-scale machine learning," *J. Mach. Learn. Res*, vol. 28, no. 3, pp. 567–575, 2013.
[20] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk, *A Distribution-Free Theory of Nonparametric Regression*. New York, NY, USA: Springer, 2006.
[21] D. Haussler, "Decision theoretic generalizations of the PAC model for neural net and other learning applications," *Inf. Comput.*, vol. 100, no. 1, pp. 78–150, 1992.
[22] G. Kriukova, S. Pereverzyev, Jr., and P. Tkachenko, "Nyström type subsampling analyzed as a regularized projection," *Inverse Probl.*, vol. 33, no. 7, 2017, Art. no. 074001.

[23] S. Lin, X. Liu, Y. Rong, and Z. Xu, "Almost optimal estimates for approximation and learning by radial basis function networks," *Mach. Learn.*, vol. 95, no. 2, pp. 147–164, 2014.

[24] S. B. Lin, X. Guo, and D. X. Zhou, "Distributed learning with regularized least squares," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 3202–3232, 2018.

[25] V. Maiorov, "Approximation by neural networks and learning theory," *J. Complexity*, vol. 22, no. 1, pp. 102–117, 2006.

[26] V. Maiorov, "Representation of polynomials by linear combinations of radial basis functions," *Construct. Approx.*, vol. 37, no. 2, pp. 283–293, 2013.

[27] M. Meister and I. Steinwart, "Optimal learning rates for localized SVMs," *J. Mach. Learn. Res.*, vol. 17, no. 194, pp. 1–44, 2016.

[28] S. Mendelson and R. Vershynin, "Entropy and the combinatorial dimension," *Inventiones Mathematicae*, vol. 152, no. 1, pp. 37–55, 2003.

[29] N. Mucke and G. Blanchard, "Parallelizing spectrally regularized kernel algorithms," *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 1069–1097, 2018.

[30] H. Niederreiter, "Low-discrepancy and low-dispersion sequences," *J. Number Theory*, vol. 30, no. 1, pp. 51–70, 1988.

[31] A. Pensia, J. Varun, and L. Po-Ling, "Generalization error bounds for noisy, iterative algorithms," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2018, pp. 546–550.

[32] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939–1959, Dec. 2005.

[33] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1177–1184.

[34] C. R. Rao, *Generalized Inverse of Matrices and Its Applications*. New York, NY, USA: Wiley, 1971.

[35] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, US: MIT Press, 2006.

[36] A. Rudi, R. Camoriano, and L. Rosasco, "Less is more: Nyström computational regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1657–1665.

[37] A. Rudi, R. Camoriano, and L. Rosasco, "Generalization properties of learning with random features," 2016. [Online]. Available: arXiv:1602.04474.

[38] A. Rudi, D. Calandriello, L. Carratino, and L. Rosasco, "On fast leverage score sampling and optimal learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5672–5682.

[39] H. Salimbeni and M. Deisenroth, "Doubly stochastic variational inference for deep Gaussian processes," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4588–4599.

[40] T. Sauer, *Numerical Analysis*. London, U.K.: Addison-Wesley, 2006.

[41] A. Schwaighofer and V. Tresp, "Transductive and inductive methods for approximate Gaussian process regression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 977–984.

[42] C. Scovel, D. Hush, I. Steinwart, and J. Theiler, "Radial kernels and their reproducing kernel Hilbert spaces," *J. Complexity*, vol. 26, no. 6, pp. 641–660, 2010.

[43] M. Seeger, C. Williams, and N. Lawrence, "Fast forward selection to speed up sparse Gaussian process regression," in *Proc. 9th Int. Workshop Artif. Intell. Statist. (AISTATS)*, 2003.

[44] F. Sheikholeslami, D. Berberidis, and G. B. Giannakis, "Large-scale kernel-based feature extraction via low-rank subspace tracking on a budget," *IEEE Trans. Signal Process.*, vol. 66, no. 8, pp. 1967–1981, Apr. 2018.

[45] L. Shi, Y. L. Feng, and D. X. Zhou, "Concentration estimates for learning with L^1-regularizer and data dependent hypothesis spaces," *Appl. Comput. Harmon. Anal.*, vol. 31, no. 2, pp. 286–302, 2011.

[46] L. Shi, "Learning theory estimates for coefficient-based regularized regression," *Appl. Comput. Harmon. Anal.*, vol. 34, no. 2, pp. 252–265, 2013.

[47] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1257–1264.

[48] I. Steinwart, D. R. Hush, and C. Scovel, "Optimal rates for regularized least squares regression." in *Proc. Annu. Conf. Comput. Learn. Theory (COLT)*, 2009, pp. 1–10.

[49] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

[50] R. Tandon, S. Si, P. Ravikumar, and I. Dhillon, "Kernel ridge regression via partitioning," 2016. [Online]. Available: arXiv:1608.01976.

[51] S. Tang and H. Gao, "Traffic-incident detection-algorithm based on nonparametric regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 38–42, Mar. 2005.

[52] V. Tresp, "A Bayesian committee machine," *Neural Comput.*, vol. 12, no. 11, pp. 2719–2741, 2000.

[53] S. Van Vaerenbergh, I. Santamaría, W. Liu, and J. C. Príncipe, "Fixed-budget kernel recursive least-squares," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2010, pp. 1882–1885.

[54] K. Wang and L. Li, *Harmonic Analysis and Approximation on the Unit Sphere*, vol. 1. Beijing, China: Science, 2000.

[55] Z. Wang, K. Crammer, and S. Vucetic, "Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale SVM training," *J. Mach. Learn. Res.*, vol. 13, no. 100, pp. 3103–3131, 2012.

[56] H. Wang, "More efficient estimation for logistic regression with optimal subsamples," *J. Mach. Learn. Res.*, vol. 20, no. 132, pp. 1–59, 2019.

[57] H. Wendland, *Scattered Data Approximation*, vol. 17. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[58] C. K. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 682–688.

[59] A. Xu and R. Maxim, "Information-theoretic analysis of generalization capability of learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2524–2533.

[60] Y. Yang, M. Pilanci, and M. J. Wainwright, "Randomized sketches for kernels: Fast and optimal nonparametric regression," *Ann. Stat.*, vol. 45, no. 3, pp. 991–1023, 2017.

[61] R. Yin, Y. Liu, W. Wang, and D. meng, "Sketch kernel ridge regression using circulant matrix: Algorithm and theory," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 29, 2019, doi: 10.1109/TNNLS.2019.2944959.

[62] J. Zeng, M. Wu, S. B. Lin, and D. X. Zhou, "Fast polynomial kernel classification for massive data," 2019. [Online]. Available: arXiv:1911.10558.

[63] K. Zhang and J. T. Kwok, "Density-weighted Nyström method for computing large kernel eigensystems," *Neural Comput.*, vol. 21, no. 1, pp. 121–146, 2009.

[64] Y. Zhang, J. Duchi, and M. Wainwright, "Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates," *J. Mach. Learn. Res.*, vol. 16, no. 106, pp. 3299–3340, 2015.

[65] D. X. Zhou and K. Jetter, "Approximation with polynomial kernels and SVM classifiers," *Adv. Comput. Math.*, vol. 25, no. 1, pp. 323–344, 2006.

[66] Z. H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum]," *IEEE Comput. Intell. Mag.*, vol. 9, no. 4, pp. 62–74, Nov. 2014.