Statistica Sinica Preprint No: SS-2021-0018	
Title	An Online Projection Estimator for Nonparametric Regression in Reproducing Kernel Hilbert Spaces
Manuscript ID	SS-2021-0018
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202021.0018
Complete List of Authors	Tianyu Zhang and Noah Simon
Corresponding Author	Tianyu Zhang
E-mail	zty@uw.edu

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

An Online Projection Estimator for Nonparametric Regression in Reproducing Kernel Hilbert Spaces

Tianyu Zhang and Noah Simon

University of Washington

Abstract: The goal of nonparametric regression is to recover an underlying regression function from noisy observations, under the assumption that the regression function belongs to a prespecified infinite-dimensional function space. In the online setting, in which the observations come in a stream, it is generally computationally infeasible to refit the whole model repeatedly. As yet, there are no methods that are both computationally efficient and statistically rate optimal. In this paper, we propose an estimator for online nonparametric regression. Notably, our estimator is an empirical risk minimizer in a deterministic linear space, which is quite different from existing methods that use random features and a functional stochastic gradient. Our theoretical analysis shows that this estimator obtains a rate-optimal generalization error when the regression function is known to live in a reproducing kernel Hilbert space. We also show, theoretically and empirically, that the computational cost of our estimator is much lower than that of other rate-optimal estimators proposed for this online setting.

Key words and phrases: nonparametric regression, online learning, reproducing kernel Hilbert space, Mercer expansion

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

1. Introduction

It is often of interest to estimate an underlying regression function, linking features to an outcome, from noisy observations. When the structure of this function is not known (e.g., when we do not want to assume a simple linear form), some form of nonparametric regression is employed. More formally, suppose we observe $(X_i, Y_i) \overset{i.i.d.}{\sim} \rho(X, Y)$, for $i = 1, 2, ..., n$, generated from the following statistical model:

$$Y_i = f_\rho(X_i) + \epsilon_i, \tag{1.1}$$

where, for each i, $X_i \overset{i.i.d.}{\sim} \rho_X$ (which take values in $\mathbb{R}^d$) are our features, $Y_i \in \mathbb{R}$ is our outcome, ϵ_i are independent and identically distributed (i.i.d.) mean zero noise variables. One can think of f_ρ as being implicitly defined by the joint distribution $\rho(X, Y)$. It is often of interest to estimate f_ρ, the regression function (e.g., in predictive modeling or inferential applications). Under mild conditions, the regression function f_ρ can also be characterized as the minimizer of

$$\min_{f \in \mathcal{F}} \mathbb{E}(Y - f(X))^2 \tag{1.2}$$

when $\mathcal{F} = L^2_{\rho_X}$, which is the best measurable function for predicting Y given X under a least squares loss.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

1.1 Nonparametric Regression in RKHS

In nonparametric regression, we often assume that f_ρ belongs to a specified infinite-dimensional function space $\mathcal{F}$. This is known as the *Hypothesis Space*. Commonly used $\mathcal{F}$ in statistics and computer science communities include the Holder ball, Sobolev space (Wahba, 1990), general reproducing kernel Hilbert space (RKHS) (Christmann and Steinwart, 2008), and Besov space (Härdle et al., 2012). Here, we focus on estimation when $\mathcal{F}$ is an RKHS. Briefly, an RKHS over $\mathcal{X}$ is a Hilbert space $(\mathcal{F}, \langle\cdot,\cdot\rangle_{\mathcal{F}})$ with the following reproducing property: for any $f \in \mathcal{F}$ and $x \in \mathcal{X}$,

$$f(x) = \langle f, K_x \rangle_{\mathcal{F}}\,, \tag{1.3}$$

where K_x is the so-called kernel function associated with $\mathcal{F}$ evaluated at x. This is discussed in more detail in Section 2.

In the classical nonstreaming setting of nonparametric regression, estimation in an RKHS $\mathcal{F}$ is a well-studied problem. In this case, the kernel ridge regression (KRR) estimator is the gold standard; see, for example (Wainwright, 2019). It is defined by

$$\hat{f}_n^{KRR} := \underset{f\in\mathcal{F}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \left(Y_i - f\left(X_i\right)\right)^2 + \lambda_n^{KRR}\|f\|_{\mathcal{F}}^2, \tag{1.4}$$

where λ_n^{KRR} is a hyperparameter that balances the mean squared error and the complexity of the estimate. Owing to the reproducing property (1.3),

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

$\hat{f}_n^{KRR}$ can be written as a finite linear combination of the kernel function evaluated at $(X_i)_{i=1}^n$ (Schölkopf et al., 2001).

In general, (1.4) requires solving an $n \times n$ linear system, and thus has a computational cost in the order of n^3. In an online setting, this is exacerbated by the need to refit for each new observation, resulting in n^4 computation being required to fit a sequence of n estimators. Although this penalized estimator has good statistical properties (rate optimal convergence and strong empirical performance), its high computational cost restricts its application in online settings. Substantial effort has been made to reduce the computational cost of KRR using, for example, "scalable kernel machines" based on a random Fourier feature (RFF) (Liu et al., 2020) or a Nyström projection (Gittens and Mahoney, 2016). This is discussed further in Section 2.1.

1.2 Parametric and Nonparametric Online Learning

Online learning has been studied thoroughly in the parametric setting: there, we assume f_ρ takes a parametric form, indexed by a finite-dimensional parameter $\beta \in \mathbb{R}^p$ (e.g., $f_\rho(X) = \beta^\top X$ for a linear model).

In this parametric online setting, it is useful to frame the regression

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

function as a population minimizer,

$$\min_{\beta \in \mathbb{R}^p} \mathbb{E}[(Y - f_\beta(X))^2]. \tag{1.5}$$

From here, it is popular to directly apply a stochastic gradient descent (SGD) to (1.5), using each sample in our "stream" to calculate one unbiased estimate of the gradient. Updating such an estimator with a new observation has a constant computational cost of $O(p)$. In addition, these estimators achieve an optimal parametric convergence rate of $O(1/n)$ under mild conditions (Kushner and Yin, 2003; Bach and Moulines, 2013; Frostig et al., 2015; Babichev and Bach, 2018).

However, comparatively less attention has been given to online nonparametric regression. A few rate-optimal functional SGD algorithms have been proposed (Tarres and Yao, 2014; Dieuleveut and Bach, 2016), where the hypothesis function space $\mathcal{F}$ is assumed to be an RKHS. The RKHS structure makes it possible to take the gradient of the evaluation functional $L_x(f) := f(x)$. Although such estimators have been shown to be statistically rate optimal, updating them with a new observation (X_{n+1}, Y_{n+1}) usually involves evaluating n kernel functions at X_{n+1}, with a computational cost of $O(n)$. This is in contrast to the constant update cost of $O(p)$ in a parametric SGD. Thus, the computational cost of a nonparametric SGD will accumulate at order $O(n^2)$, which is not ideal for methods that

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

are nominally designed to deal with large data sets. Although there has been some effort devoted to transfer RFF- or Nystrom-based methods to online settings (See Section 2.1), the theoretical guarantees are usually not close to optimal, with strong restrictions on the noise variables.

Our contribution We propose a method for constructing online estimators in an RKHS by considering the Mercer expansion (eigendecomposition) of a kernel function. Existing methods usually take an iterative form, which can be interpreted as projecting a random function onto a random space with growing dimension (Koppel et al., 2019, Equation (15)). However, our estimator is the first one that can be treated as an empirical risk minimizer (ERM, or M-estimator of negative loss) in a deterministic linear space with growing dimension.

We analyze both the statistical and the computational properties of the estimator to show that i) it has an asymptotically optimal (up to a logarithm term) generalization error, ii) it has a significantly lower computational cost than those of other proposed rate-optimal nonparametric SGD estimators, and iii) it is robust against heavy-tailed noise. Interestingly, it only requires the $(1 + \Delta)$ moment of the noise to be finite for any $\Delta > 0$ to achieve consistency.

Note that in the theoretical analysis of our estimator, we do not require

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

the covariate X to be equally spaced or uniformly distributed, as in standard references (Tsybakov, 2008) (though such assumptions would significantly simplify the proof). In addition, we do not require it to be known for rate optimal convergence. We show that our estimator obtains rate optimal convergence if ρ_X is absolutely continuous with respect to the measure used to conduct the eigendecomposition of the kernel function (usually, the latter is taken as a uniform measure or a Gaussian distribution).

Notation: we use $a_n = \Theta(b_n)$ to indicate that two sequences increase/decrease at the same rate as $n \to \infty$. Formally,

$$0 < \liminf_{n\to\infty} \left|\frac{a_n}{b_n}\right| \leq \limsup_{n\to\infty} \left|\frac{a_n}{b_n}\right| < \infty. \tag{1.6}$$

For $a \in \mathbb{R}$, $\lfloor a \rfloor$ is the largest integer that is smaller than or equal to a. The $\|\cdot\|_2$-norm of a function is its $L^2_{\rho_X}$-norm, that is $\|f\|_2^2 = \int_{\mathcal{X}} f^2(z) d\rho_X(z)$. In this paper, when we say two functions f and g are orthogonal with respect to the measure P, we mean $\int f(x)g(x)dP(x) = 0$.

2. Preliminaries on RKHS

In this section, we provide background information on RKHS and existing methods, before introducing our estimation procedure.

First, we formally introduce the concept of a Mercer kernel and its corresponding RKHS. A symmetric bivariate function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

positive semi-definite (PSD) if, for any $n \geq 1$ and $(x_i)_{i=1}^n \subset \mathcal{X}$, the $n \times n$ kernel matrix $\mathbb{K}$ with elements $\mathbb{K}_{ij} := K(x_i, x_j)$ is always a PSD matrix. A continuous, bounded, PSD kernel function K is called a *Mercer kernel*. We have the following duality between a Mercer kernel and a Hilbert space.

Proposition 1. *For any Mercer Kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, let K_x denote the function $K_x(\cdot) := K(x, \cdot)$. There exists a unique Hilbert Space $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ of functions on $\mathcal{X}$ satisfying the following conditions:*

1. *For all $x \in \mathcal{X}$, $K_x \in \mathcal{H}$.*
2. *The linear span of $\{K_x \mid x \in \mathcal{X}\}$ is dense (w.r.t $\| \cdot \|_{\mathcal{H}}$) in $\mathcal{H}$.*
3. *(reproducing property) For all $f \in \mathcal{H}, x \in \mathcal{X}$,*

$$f(x) = \langle f, K_x \rangle_{\mathcal{H}}. \tag{2.1}$$

We call this Hilbert space the RKHS associated with kernel K, or the *native space* of K. For a more comprehensive discussion of the RKHS, see Cucker and Smale (2002), Wainwright (2019), and Fasshauer and McCourt (2015).

There is an equivalent definition of the RKHS, which we focus on here. Given any Mercer kernel K and any Borel measure ν, there exists a set of L^2_ν-orthonormal basis $(\phi_j)_{j=1}^\infty$ of $\bar{\mathcal{H}}$ (closure of $\mathcal{H}$ with respect to $\| \cdot \|_{L^2_\nu}$).

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

Additionally, each of the functions has a paired positive real number μ_j, sorted s.t. $\mu_j \geq \mu_{j+1} > 0$. We call the functions ϕ_j eigenfunctions and μ_j their corresponding eigenvalues. We state the following equivalent definition of the native space of K.

Proposition 2. *Define a Hilbert space*

$$\mathcal{H} = \left\{ f \in L^2_\nu \mid f = \sum_{k=1}^{\infty} \theta_j \phi_j \ \textit{with} \sum_{j=1}^{\infty} \left(\frac{\theta_j}{\sqrt{\mu_j}} \right)^2 < \infty \right\} \tag{2.2}$$

equipped with inner product:

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{\infty} \frac{a_j b_j}{\mu_j}, \tag{2.3}$$

for $f = \sum_{j=1}^{\infty} a_j \phi_j$ *and* $g = \sum_{j=1}^{\infty} b_j \phi_j$.

Then, $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ *is the reproducing Hilbert space of kernel* K.

For a discussion of this definition and its relation to Proposition 1, see Cucker and Smale (2002). For many kernels, the analytical form of (μ_j, ϕ_j) are available for some specific choice of measure ν. This can be useful for our method. We require the eigen-system of the kernel with respect to some (relatively arbitrary) measure. This measure does *not* need to be the measure ρ_X, it merely needs to be absolutely continuous with respect to ρ_X. We assume such a convenient measure, denoted by $\bar{\rho}_X$, exists (for which the kernel has an accessible eigen-system and $\bar{\rho}_X \ll \rho_X$). We call it a

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

working measure, and use the notation (λ_j, ψ_j) instead of the generic (μ_j, ϕ_j) to denote such an eigen-system with respect to $L^2_{\bar{\rho}_X}$. As an example, the kernel $K(x, z) = \min\{x, z\}$ is the reproducing kernel of the Sobolev space

$$W_1^0([0,1]) = \left\{ f : [0,1] \to \mathbb{R} \mid f(0) = 0 \text{ and } \int_0^1 \left(f'(x)\right)^2 dx < \infty \right\}, \quad (2.4)$$

and its eigenfunctions and eigenvalues are (w.r.t. $\bar{\rho}_X = \text{Unif}([0,1])$)

$$\psi_j(x) = \sqrt{2}\sin\left(\frac{(2j-1)\pi x}{2}\right) \quad \lambda_j = \frac{4}{(2j-1)^2\pi^2}. \quad (2.5)$$

It is also possible to write the kernel as a *Mercer expansion* w.r.t (ψ_j, λ_j):

$$K(x, z) = \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j(z). \quad (2.6)$$

The functions $\{\sqrt{\lambda_j}\psi_j(x), j = 1, 2, ...\}$ are also called the feature maps of the kernel K. Note too that, by definition, ψ_j are orthogonal w.r.t. $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Twenty commonly used kernels' Mercer expansions are provided in (Fasshauer and McCourt, 2015, Appendix A).

If a function $f = \sum_{j=1}^{\infty} \theta_j \psi_j$ has a finite $\| \cdot \|_{\mathcal{H}}$ RKHS-norm, its general Fourier coefficients $(\theta_j)_{j \in \mathbb{N}}$ need to be at least $o(\lambda_j j^{-1/2})$ so that the norm series $\sum_{j=1}^{\infty} (\theta_j / \sqrt{\lambda_j})^2$ converges. This suggests that, for sufficiently large N, the truncation $f_N = \sum_{j=1}^{N} \theta_j \psi_j$ should be a good approximation to f. This basic idea motivates our work. By analyzing the spectrum of the kernel, we can identify what N should be.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

2.1 Existing Online Nonparametric Methods

In an RKHS, it is possible to take the functional gradient of the evaluation operator L_x, for any $x \in \mathcal{X}$. This allows methods using a functional SGD to solve the regression problem (1.2). Usually, *functional SGD* estimators after n steps, $\hat{f}_n^{SGD}$ of f_ρ, take the form of a weighted sum of n kernel functions K_{X_i}, for $i = 1, 2, ..., n$, (Tarres and Yao, 2014; Dieuleveut and Bach, 2016):

$$\hat{f}_n^{SGD} = \sum_{i=1}^{n} a_i K_{X_i}. \tag{2.7}$$

To update $\hat{f}_n^{SGD}$ with (X_{n+1}, Y_{n+1}), it is necessary to evaluate all n kernel basis functions $\{K_{X_i}, i = 1, 2, ...n\}$ at X_{n+1}. Thus, the computational cost of the update is $O(n)$. Several works have attempted to improve this computational cost. In Si et al. (2018), Lu et al. (2016), and Koppel et al. (2019), the authors choose a subset of features $(K_{X_i})_{i=1}^n$ with cardinality smaller than n. In Dai et al. (2014) and Lu et al. (2016), kernel-agnostic random Fourier features are used: typically, $O(\sqrt{n})$ basis functions are required in this setting; see Rudi and Rosasco (2017). Although computationally more efficient than a vanilla functional SGD (2.7), the theoretical aspects of these scalable methods are not fully satisfying: 1) noise variables are required to have extremely light tails to provably guarantee convergence; 2) verified convergence rates are not minimax-optimal; and 3) the target parameter is,

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

in general, not even f_ρ but, instead, a penalized population risk-minimizer.

Compared with the linear space spanned by random features or kernel functions, the space spanned by eigenfunctions has a minimal approximation error in the sense of minimizing the Kolmogorov N-width (Santin and Schaback, 2016, Section 3). This inspired us to use them as basis functions to construct our estimator. Briefly, this means that projecting onto the N-dimensional linear space spanned by the eigenfunctions has the minimal residual among all the N-dimension linear sub-spaces of $L^2_{\tilde{\rho}_X}$. More technically,

$$\sup_{\|f\|_{\mathcal{H}}=1} \left\| f - \Pi_{L^2_{\tilde{\rho}_X},\, \mathcal{F}_N} f \right\|_{L^2_{\tilde{\rho}_X}} = \inf_{V_N \subset L^2_{\tilde{\rho}_X}} \sup_{\|f\|_{\mathcal{H}}=1} \left\| f - \Pi_{L^2_{\tilde{\rho}_X},\, V_N} f \right\|_{L^2_{\tilde{\rho}_X}} = \sqrt{\lambda_{N+1}}, \tag{2.8}$$

where $\mathcal{F}_N$ is the linear space spanned by the first N eigenfunctions $(\psi_j)_{j=1}^N$, $\Pi_{A,B}$ is the projection operator onto space B using the inner product of A, and V_N is a generic N-dimensional linear space in $L^2_{\tilde{\rho}_X}$. This is important for statistical estimation, because there is a bias/variance tradeoff in this estimation problem (more basis functions decreases the bias, but increases the variance). By using a basis that can more compactly represent our function, we can find a more favorable tradeoff and asymptotically decrease our estimation error.

We propose a method with favorable statistical guarantees (minimax

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

rate-optimality) and a lower computational cost. The basis functions used should be kernel-sensitive, and the convergence rate should be sensitive to the decay rate of the eigenvalues λ_j. In addition, we give provable theoretical guarantees in a heavy-tail noise setting.

3. A Computationally Efficient Online Estimator

In this section, we present the proposed online regression estimator. We first discuss the well-known projection estimator in the batch learning setting, then shift to the online setting, where we naively refit the model with each observation. Lastly, we give our proposed modification to make this process computationally efficient. In what follows, we use N to denote the number of basis functions used to construct each projection estimator, though it should more formally be written as $N(n)$, because it is a nondecreasing function of n.

3.1 Projection Estimator in Batch Learning

Suppose we have n samples $(X_i, Y_i)_{i=1}^n$, and let $\mathcal{F}_N = \text{span}(\psi_1, ..., \psi_N)$ be the N-dimensional linear space spanned by the N eigenfunctions with the largest eigenvalues. The function $\hat{f}_{n,N}$ that minimizes the empirical mean squared error over $\mathcal{F}_N$ is a very attractive candidate for estimating $f_\rho \in \mathcal{H}$,

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

which we use for the online setting.

Formally, define $\boldsymbol{\theta} = (\theta_1, ..., \theta_N)^\top$ and $\boldsymbol{\psi}^N(X_i) = (\psi_1(X_i), ..., \psi_N(X_i))^\top$. Consider the following least squares problem (in the Euclidean space):

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^N} \sum_{i=1}^{n} (Y_i - \boldsymbol{\theta}^\top \boldsymbol{\psi}^N(X_i))^2. \tag{3.1}$$

The solution can be written in matrix form as

$$\hat{\boldsymbol{\theta}} := (\hat{\theta}_1, ..., \hat{\theta}_N)^\top = (\Psi_n^\top \Psi_n)^{-1} \Psi_n^\top \boldsymbol{Y}_n, \tag{3.2}$$

if $\Psi_n^\top \Psi_n$ is invertible. Here, $\boldsymbol{Y}_n = (Y_1, ..., Y_n)^\top$ is the observed response, and Ψ_n is the design matrix with elements $\Psi_{ij} = \psi_j(x_i)$. Then, the estimator

$$\hat{f}_{n,N} = \sum_{j=1}^{N} \hat{\theta}_j \psi_j \tag{3.3}$$

is the empirical risk minimizer (ERM) in $\mathcal{F}_N$. Estimators that take this form are called nonparametric *projection estimators* (of f_ρ, with level N) (Tsybakov, 2008).

The optimal number of basis functions to use depends on both the sample size n and how fast the eigenvalues λ_j in (2.6) decay. As stated formally in Theorem 3, the optimal choice is $N = \Theta(n^{\frac{d}{2\alpha+d}})$ when $\lambda_j = \Theta(j^{-2\alpha/d})$, with $\alpha > \frac{d}{2}$. Note that the condition $\alpha > \frac{d}{2}$ ensures that the considered RKHS can be embedded into the space of continuous functions (as a result of the Sobolev inequality, cf. Theorem 12.55 (Leoni, 2017)).

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

With this choice for N, the convergence of $\hat{f}_{n,N}$ achieves the minimax rate over functions with a bounded RKHS norm. Similar results for projection estimators have been shown when $(\psi_j)_{j=1}^{\infty}$ is the trigonometric basis, and x_i are deterministic and evenly spaced (Tsybakov, 2008) or ρ_X is the uniform distribution (Belloni et al., 2014). Our analysis shows that the optimality of the projection estimator holds for general ψ_j, and does not require them to be orthonormal with respect to the empirical measure or ρ_X.

3.2 Naive Online Projection Estimator

The most direct way of extending the projection estimator (3.3) to the online setting is simply to refit the whole model whenever a new pair of data (X_i, Y_i) comes in. In Algorithm 1, we provide this naive updating rule for our reader to better understand the proposed method. Our modified proposal in Section 3.4 greatly improves upon this in terms of computational cost, while giving the same estimates $\hat{f}_{n,N}$.

In this algorithm, $\boldsymbol{Y}_n = (Y_1, ..., Y_n)^\top$ is the vector of outcomes, Ψ_n is the $n \times N$ design matrix at step n, and Φ_n denotes the $N \times N$ matrix $(\Psi_n^\top \Psi_n)^{-1}$ (inversion of Gram matrix).

Whenever new data come in, the algorithm augments the design matrix by adding one new row to Ψ_{n-1} based on the new observation X_n. The new

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

Algorithm 1 Naive rule for updating $\hat{\boldsymbol{\theta}}$ with a new observation (X_n, Y_n).

Input: $(X_i)_{i=1}^n, \boldsymbol{Y}_n, \Phi_{n-1}, \Psi_{n-1}, \alpha, N$

function UpdateCurrent(X_n, N, Φ_n, Ψ_n)

$\boldsymbol{\psi}_n \leftarrow [\psi_1(X_n), \psi_2(X_n), ..., \psi_N(X_n)]^\top$

$\Psi_n \leftarrow \begin{bmatrix} \Psi_n \\ \boldsymbol{\psi}_n^\top \end{bmatrix} \quad \Phi_n \leftarrow \left(\Psi_n^\top \Psi_n\right)^{-1}$

return (Φ_n, Ψ_n)

function AddBasis $((X_i)_{i=1}^n, N, \Phi_n, \Psi_n)$

$\boldsymbol{\psi}^{N+1} \leftarrow [\psi_{N+1}(X_1), ..., \psi_{N+1}(X_n)]^\top$

$\Psi_n \leftarrow \begin{bmatrix} \Psi_n & \boldsymbol{\psi}^{N+1} \end{bmatrix} \quad \Phi_n \leftarrow \left(\Psi_n^\top \Psi_n\right)^{-1}$

return (Φ_n, Ψ_n)

if $n = \text{Floor}((N+1)^{2\alpha+1})$ **then**

$(\Phi_n, \Psi_n) \leftarrow$ UpdateCurrent$(X_n, N, \Phi_{n-1}, \Psi_{n-1})$

$(\Phi_n, \Psi_n) \leftarrow$ AddBasis$((X_i)_{i=1}^n, N, \Phi_n, \Psi_n)$

$N \leftarrow N + 1$

else

$(\Phi_n, \Psi_n) \leftarrow$ UpdateCurrent$(X_n, N, \Phi_{n-1}, \Psi_{n-1})$

end if

$\hat{\boldsymbol{\theta}} \leftarrow \Phi_n \Psi_n^\top \boldsymbol{Y}_n$

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

row $[\psi_1(X_n), \psi_2(X_n), ..., \psi_N(X_n)]$ can be understood as the embedding of X_n into the feature space spanned by $(\psi_j)_{j=1}^N$.

When $n = \lfloor (N+1)^{\frac{2\alpha+d}{d}} \rfloor$, this algorithm additionally adds a new column to the design matrix Ψ_n (increasing the dimension of the basis function we project upon by one). This new column is just the evaluation of ψ_{N+1} at $(X_i)_{i=1}^n$. Recall that ψ_{N+1} is the $(N+1)$th eigenfunction in the Mercer expansion (2.6). It is straightforward to show that this criterion of adding new basis functions ensures $N = \Theta(n^{\frac{d}{2\alpha+d}})$.

The computational cost of each update using Algorithm 1 is $\sim n^{\frac{2\alpha+3d}{2\alpha+d}}$. In particular, calculating $\Psi_n^\top \Psi_n$ takes $\sim nN^2 \sim n^{\frac{2\alpha+3d}{2\alpha+d}}$ computations. Although this algorithm gives a statistically rate-optimal estimator and is straightforward to implement, it is rather computationally expensive. In particular, the functional SGD algorithm has a comparatively smaller computational cost of $\sim n$ per update.

3.3 Efficient Online Projection Estimator

In this section, we explicitly give our proposed method (the details of which are given in Algorithm 2). By using some common block/rank-one updating tools from linear algebra, we are able to substantially improve Algorithm 1. In particular, it is expensive to repeatedly calculate $(\Psi_n^\top \Psi_n)^{-1}$

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

directly. However, the matrix Ψ_n has only one more row and (sometimes) one more column than Ψ_{n-1}. It is possible to calculate $(\Psi_n^\top \Psi_n)^{-1}$ by updating $(\Psi_{n-1}^\top \Psi_{n-1})^{-1}$. The latter will already have been calculated when observing (X_{n-1}, Y_{n-1}).

When Ψ_n has one more row than Ψ_{n-1},

$$\Psi_n = \begin{bmatrix} \Psi_{n-1} \\ \boldsymbol{\psi}_n^\top \end{bmatrix}, \tag{3.4}$$

where $\boldsymbol{\psi}_n = \left[\psi_1\left(X_n\right), \psi_2\left(X_n\right), \ldots, \psi_N\left(X_n\right)\right]^\top$. We can write $\Psi_n^\top \Psi_n$ in the form

$$\Psi_n^\top \Psi_n = \Psi_{n-1}^\top \Psi_{n-1} + \boldsymbol{\psi}_n \boldsymbol{\psi}_n^\top. \tag{3.5}$$

Thus, $\left(\Psi_n^\top \Psi_n\right)^{-1}$ can be calculated from $\left(\Psi_{n-1}^\top \Psi_{n-1}\right)^{-1}$ and $\boldsymbol{\psi}_n$ using the Sherman–Morrison formula (Sherman and Morrison, 1950).

When Ψ_n has one more column than Ψ_{n-1},

$$\Psi_n = \begin{bmatrix} \Psi_{n-1} & \boldsymbol{\psi}^{N+1} \end{bmatrix}. \tag{3.6}$$

We can write $\Psi_n^\top \Psi_n$ in the form

$$\Psi_n^\top \Psi_n = \begin{bmatrix} \Psi_{n-1}^\top \Psi_{n-1} & \Psi_{n-1}^\top \boldsymbol{\psi}^{N+1} \\ \left(\boldsymbol{\psi}^{N+1}\right)^\top \Psi_{n-1} & \left(\boldsymbol{\psi}^{N+1}\right)^\top \boldsymbol{\psi}^{N+1} \end{bmatrix}. \tag{3.7}$$

Therefore, $\left(\Psi_n^\top \Psi_n\right)^{-1}$ is related to $\left(\Psi_{n-1}^\top \Psi_{n-1}\right)^{-1}$ by the block matrix inversion formula (Petersen and Petersen, 2008).

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

The detailed updating rule of the proposed method is given explicitly in Algorithm 2. The basic structure of this algorithm is identical to that of Algorithm 1. However, the updating rules discussed above are used to avoid recalculating some quantities from scratch. We also establish a recursive relationship between $\hat{\boldsymbol{\theta}}_{n+1}$ and $\hat{\boldsymbol{\theta}}_n$. Curiously, the recursive formula has a form very similar to that of the pre-conditioned SGD estimator (with the inverse of the Gram matrix as the pre-conditioner). When $n \neq \lfloor (N+1)^{\frac{2\alpha+d}{d}} \rfloor$, the recursion is

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + \Phi_n \boldsymbol{\psi}_n \left[Y_n - \hat{f}_{n-1,N}(X_n) \right]. \tag{3.8}$$

Note that for the SGD, the updating rule replaces Φ_n by I, the identity matrix, thus omitting the correlation of ψ_j w.r.t. the empirical measure. When features are added, there is still a geometrical interpretation; see the Supplementary Material, S3.

3.4 Computational Cost of Algorithm 2

We now show that the computational cost of the updating rule in Algorithm 2 is, on average, $O(n^{\frac{2d}{2\alpha+d}})$.

When $n \neq \lfloor (N+1)^{\frac{2\alpha+d}{d}} \rfloor$, we do not add a new feature ψ_{N+1}, but only update the Φ_{n-1} matrix with the current N features. The most expensive step is the inner product of Φ_{n-1} and $\boldsymbol{\psi_n}$, which is an $N \times N$ matrix multi-

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

Algorithm 2 Rule for updating $\hat{\boldsymbol{\theta}}$ with a new observation (X_n, Y_n) efficiently. At step $*$, the value of $\Psi_{n-1}^{\top}\boldsymbol{Y}_{n-1}$ stored in memory needs to be used to avoid repeating calculation.

Input: $(X_i)_{i=1}^{n}, \boldsymbol{Y}_n, N, \Phi_{n-1}, \Psi_{n-1}, a, \Psi_{n-1}^{\top}\boldsymbol{Y}_{n-1}$

function UpdateCurrent $(X_n, N, \Phi_{n-1}, \Psi_{n-1})$ **output** (Φ_n, Ψ_n)

$\boldsymbol{\psi}_n \leftarrow [\psi_1(X_n), \psi_2(X_n), ..., \psi_N(X_n)]^{\top}$

$\Psi_n \leftarrow [\Psi_{n-1}^{\top}\ \boldsymbol{\psi}_n]^{\top}$, $\Phi_n \leftarrow \Phi_{n-1} - \frac{\Phi_{n-1}\ \boldsymbol{\psi_n}\ \boldsymbol{\psi_n}^T\ \Phi_{n-1}}{1 + \boldsymbol{\psi_n}^T\ \Phi_{n-1}\ \boldsymbol{\psi_n}}$

function AddBasis $((X_i)_{i=1}^{n}, N, \Phi_n, \Psi_n)$ **output** (Φ_n, Ψ_n)

$\boldsymbol{\psi}^{N+1} \leftarrow [\psi_{N+1}(X_1), \psi_{N+1}(X_2), ..., \psi_{N+1}(X_n)]^{\top}$

$c \leftarrow \left(\boldsymbol{\psi}^{N+1}\right)^{\top}\boldsymbol{\psi}^{N+1} \quad \boldsymbol{b} \leftarrow \Psi_n^{\top}\boldsymbol{\psi}^{N+1} \quad k \leftarrow c - \boldsymbol{b}^{\top}\Phi_n\,\boldsymbol{b}$

$$\Psi_n \leftarrow \begin{bmatrix} \Psi_n & \boldsymbol{\psi}^{N+1} \end{bmatrix}, \ \Phi_n \leftarrow \begin{bmatrix} \Phi_n + \frac{1}{k}\Phi_n\,\boldsymbol{b}\boldsymbol{b}^T\,\Phi_n & -\frac{1}{k}\Phi_n\,\boldsymbol{b} \\ -\frac{1}{k}\boldsymbol{b}^T\,\Phi_n & \frac{1}{k} \end{bmatrix}$$

$(\Phi_n, \Psi_n) \leftarrow \text{UpdateCurrent}(X_n, N, \Phi_{n-1}, \Psi_{n-1})$

if $n = \text{Floor}((N+1)^{2a+1})$ **then**

$(\Phi_n, \Psi_n) \leftarrow \text{AddBasis}((X_i)_{i=1}^{n}, N, \Phi_n, \Psi_n)$

$N \leftarrow N + 1$

end if

$\hat{\boldsymbol{\theta}} \leftarrow \Phi_n\Psi_n^{\top}\boldsymbol{Y}_n \quad *$

plied by an $N \times 1$ vector. Because the $N = \Theta(n^{\frac{d}{2\alpha+d}})$ at step n, the update is of order $n^{\frac{2d}{2\alpha+d}}$.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

When $n = \lfloor (N+1)^{\frac{2\alpha+d}{d}} \rfloor$, we add both a column and a row to the design matrix Ψ_{n-1}. The most expensive step is calculating the vector $\boldsymbol{b}$, which gives the pair-wise inner product between ψ_{N+1} and $(\psi_j)_{j=1}^N$ with respect to the empirical measure. In this step, an $N \times (n-1)$ matrix is multiplied by an $(n-1) \times 1$ vector, which requires a computation of order $n^{\frac{2\alpha+2d}{2\alpha+d}}$. However, the algorithm adds new features less frequently as n increases. Thus, in calculating the average computational cost, we amortize this expense over all updates after including new basis functions.

Let

$$n = (N)^{\frac{2\alpha+d}{d}}$$

$$n^+ = (N+1)^{\frac{2\alpha+d}{d}}.$$

That is, n is the first step when there are more than N features included, and n^+ is the first step when there are more than $N+1$ features. Then, the length of the interval between the two "basis adding" steps is

$$\begin{aligned} n^+ - n &= (N+1)^{\frac{2\alpha+d}{d}} - (N)^{\frac{2\alpha+d}{d}} \\ &= \Theta(N^{2\alpha/d}) = \Theta(n^{\frac{2\alpha}{2\alpha+d}}). \end{aligned}$$

Thus, an $O(n^{\frac{2\alpha+2d}{2\alpha+d}})$ computation is performed per $n^{\frac{2\alpha}{2\alpha+d}}$ steps, which is, on average, $O(n^{\frac{2d}{2\alpha+d}})$ per step. Thus, the average computational cost of a *single update* using Algorithm 2 is of order $n^{\frac{2d}{2\alpha+d}}$.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

4. Theoretical Analysis of the Online Projection Estimator

In this section, we formally show that the proposed online estimator achieves the optimal statistical convergence rate when the true regression function belongs to the hypothesized RKHS. In previous theoretical analyses of (batch) projection estimators (Tsybakov, 2008), the proof is shown when ψ_j are orthogonal to each other w.r.t. the empirical measure of the covariates. This event has probability zero if X has a continuous density. In this section, we show it is possible to get a rate-optimal bound on the generalization error of $\hat{f}_{n,N}$, even if ψ_j (the eigenfunctions of the kernel w.r.t. our "convenient" working distribution) are quite correlated w.r.t. the empirical measure of X.

Recall that $\mathcal{F}_N = \text{span}(\psi_1, ..., \psi_N)$ is the linear space spanned by the first N eigenfunctions. Define the *population* minimizer f_N over $\mathcal{F}_N$ as

$$f_N := \arg\min_{f \in \mathcal{F}_N} \mathbb{E}[(f(X) - f_\rho(X))^2]. \tag{4.1}$$

Here, recall that $\hat{f}_{n,N} \in \mathcal{F}_N$ is the estimator, f_N is the population risk minimizer over $\mathcal{F}_N$, and $f_\rho \in \mathcal{H}$ is the target function to be estimated. To establish the result that $\|\hat{f}_{n,N} - f_\rho\|_2 \to 0$ as $n \to \infty$, we first bound the rate at which $\|\hat{f}_{n,N} - f_N\|_2$ goes to zero as N grows (sufficiently slowly); then, we bound the rate at which $\|f_N - f_\rho\|_2 \to 0$ as $N \to \infty$. With the

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

correct choice of $N = \Theta(n^{\frac{d}{2\alpha+d}})$, we can balance the rate of the above two terms converging to zero. Before we state the result, we give assumptions necessary for the proof.

(A1) The joint distribution of i.i.d. (X_i, Y_i) has support $\mathcal{X} \times \mathbb{R} \subset \mathbb{R}^d \times \mathbb{R}$ and $\mathcal{X}$ is compact. The i.i.d. zero-mean noise random variables $\epsilon_i = Y_i - f_\rho(X_i)$ satisfy the following:

$$\|\epsilon_i\|_{m,1} := \int_0^\infty \mathbb{P}(|\epsilon_i| > t)^{1/m} dt < \infty, \quad \text{for some } m > 1. \tag{4.2}$$

Note. If for some $\delta > 0$ and $m > 1$, we have that the $m + \delta$ moment of ϵ_i exists, then (A1) is satisfied for that value of m. This is *slightly* stronger than the existence of the mth moment; see Ledoux and Talagrand (2013), Chapter 10.

Our noise assumption is substantially weaker than the typical sub-Gaussian noise assumptions (sub-Gaussian random variables have all moments bounded). In the light-tail noise setting, the level of the noise only influences the convergence speed by at most a constant. However, as shown in Theorem 3, if the eigenvalues decrease too fast (the RKHS is too small) and the noise has too few moments, the convergence *rate* will depend on the noise level. Our analysis characterizes the interplay between the size of the RKHS space and the noise level using a sharp multiplier inequality (Han et al., 2019, Theorem 1). There are currently no other methodologies,

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

to the best of our knowledge, that are both computationally tractable and have provable convergence guarantees with heavy-tailed noise in the online nonparametric regression setting.

(A2) The true regression function f_ρ belongs to the known RKHS $\mathcal{H}$; that is, the RKHS-norm $\|f_\rho\|_{\mathcal{H}}$ is finite.

(A3) The kernel function has Mercer expansion $K(x,z) = \sum_{j=1}^{\infty} \lambda_j \psi_j(x)\psi_j(z)$, where $(\psi_j)_{j=1}^{\infty}$ are orthonormal with respect to some specified *working distribution* $\bar{\rho}_X$, and $\lambda_j = \Theta(j^{-2\alpha/d})$ with $\alpha > d/2$.

(A4) The distribution of X, ρ_X, is absolutely continuous w.r.t. $\bar{\rho}_X$. Let $p_X = d\rho_X/d\bar{\rho}_X$ denote its Radon–Nikodym derivative. We assume, for some $D < \infty$,

$$p_X(x) \le D \quad \text{for all } x \in \mathcal{X}.$$

Note. In the (very common) case that both of these have densities with respect to the Lebesgue measure, this is equivalent to the ratio of their densities being bounded.

Theorem 3 (Optimal convergence rate)**.** *Assume (A1–A4), let $\hat{f}_{n,N}$ be the projection estimator* (3.3)*. Assume that $\|\hat{f}_{n,N}\|_\infty \le M$, for some $M < \infty$. Choosing $N = \Theta(n^{\frac{d}{2\alpha+d}})$, we have*

$$\|\hat{f}_{n,N} - f_\rho\|_2 = O_P\left(n^{-\frac{\alpha}{2\alpha+d}}\sqrt{\log n} \vee n^{-\frac{1}{2}+\frac{1}{2m}}\sqrt{\log n}\right). \tag{4.3}$$

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

If $m \geq 2$ in (A1), the above bound holds in expectation:

$$\mathbb{E}[\|\hat{f}_{n,N} - f_\rho\|_2] = O\left(n^{-\frac{\alpha}{2\alpha+d}}\sqrt{\log n} \vee n^{-\frac{1}{2}+\frac{1}{2m}}\sqrt{\log n}\right). \tag{4.4}$$

Note that as long as all the moments of ϵ_i exist (e.g., when ϵ_i are sub-exponential), the convergence rate depends only on the size of the RKHS. One merit of our method is that even if the noise does not have a finite variance, that is, $m < 2$ in (A1), our method still has convergence guarantees. To the best of our knowledge, existing works on nonparametric SGD do not give convergence guarantees with such heavy-tailed noise.

As we compare the two components on the RHS of the bound presented in (4.4), we can see that when $m > \frac{2\alpha}{d} + 1$, that is, when we have a relatively light-tailed noise, our bound is dominated by the size of the RKHS. However, when $m < \frac{2\alpha}{d} + 1$, it is the noise that dominates our bound. Furthermore, note that as d increases, fewer moments on ϵ are required for our bound to match the classical nonparametric minimax rate in our RKHS.

The following lower bound demonstrates that this rate of convergence is indeed optimal (up to a logarithm term) among all estimators. For $\lambda_j = \Theta(j^{-2\zeta})$ (to compare with Theorem 3, take $\zeta = \alpha/d$), let $B_R = \{f \in \mathcal{H} \mid \|f\|_{\mathcal{H}} \leq R\}$ be the R-ball in the RKHS $\mathcal{H}$. Then, we have the minimax

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

bound

$$\liminf_{n\to\infty} \inf_{\hat{f}} \sup_{f_\rho \in B_R} \mathbb{E}\left[n^{\frac{\zeta}{2\zeta+1}} \|\hat{f} - f_\rho\|_2 \right] \geq C, \tag{4.5}$$

where the infimum ranges over all possible functions $\hat{f}$ that are measurable of the data. For a derivation of the lower bound, see Wainwright (2019, Chap. 15).

Upper bounds similar to our results in Theorem 3 have been shown in Tarres and Yao (2014) and Dieuleveut and Bach (2016) for SGD-type nonparametric online methods. However, the proposed estimators there use n basis functions, and therefore have an unacceptable $\Theta(n^2)$ total computational cost. There are methods that aim to improve the computational aspect by using random features or other acceleration methods (see Section 2.1). However, the theoretical guarantees on the statistical convergence rates in those works are, in general, quite weak (generally giving upper bounds of $n^{-1/4}$ in the RMSE, which is far from the minimax rate) and insensitive to the decay rate of the eigenvalues.

Many existing online nonparametric estimators aim to find a function $f \in \mathcal{F}$ that minimizes an expected convex loss $\mathbb{E}[l(f(X), Y)]$, which is a more general setting than this study. However, the majority of previous works on this topic assume that the loss function $l(\cdot, \cdot)$ is Lipschitz w.r.t. the first argument; see Dai et al. (2014), Si et al. (2018), Koppel et al.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

(2019), and Lu et al. (2016). Specializing to the regression problem (with squared-error-loss), this is essentially assuming that the outcomes Y_i (therefore the noise ϵ_i) are uniformly bounded, because $l(f(x), y) - l(f(z), y) = (f(x)-y)^2-(f(z)-y)^2 = (f(x)-f(z))(f(x)+f(z)-2y)$. If we require $l(\cdot, \cdot)$ to be Lipschitz, we basically require $f(x), f(z), y$ to be uniformly bounded. Although we still only consider bounded f in this work, we relax the constraint on the noise variables: we require only finite moments of ϵ_i, and show the (in)sensitivity of our bound.

5. Multivariate Regression Problems

In most applications, the covariates X_i take values in $\mathbb{R}^d$s where $d > 1$. If the kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ has a known Mercer expansion (2.6), then the proposed method can be applied directly. If the kernel function takes a tensor product form (e.g. the Gaussian kernel), or is constructed from a one-dimensional kernel using a tensor product (e.g., $K(x, z) = \prod_{k=1}^{d} \min\{x^{(k)}, z^{(k)}\}$, where $x^{(k)}$ is the kth entry of $x \in \mathbb{R}^d$), the eigenvalues and eigenfunctions are just the tensor product of the one-dimensional kernels (Michel, 2012, Section 3.5), (Xiu, 2010, Section 5.2). However, as presented in Section 4, the minimax rate of estimating in a d-dimensional α-order Sobolev space is $\Theta(n^{-\frac{\alpha}{2\alpha+d}})$, which becomes quite slow

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

when d is large (unless, at the same time, a large α is assumed).

A popular low-dimensional structure is the nonparametric additive model (Hastie et al., 2009; Yuan and Zhou, 2016), which is thought to effectively balance model flexibility and interpretability. For $x \in \mathbb{R}^d$, we might consider imposing an additive structure on our model (1.1):

$$f_\rho(x) = \sum_{k=1}^{d} f_{\rho,k}\left(x^{(k)}\right), \tag{5.1}$$

where the component functions $f_{\rho,k}$ belong to an RKHS $\mathcal{H}$ (in general, they can belong to different spaces). For a fixed d, the minimax rate for estimating an additive model is identical (up to a multiplicative constant d) to the minimax rate in the analogous one-dimensional nonparametric regression problem that works with the same hypothesis space $\mathcal{H}$ (Raskutti et al., 2009). The proposed online method can be directly generalized to this setting. For further discussion and the empirical performance, see the Supplementary Material, S4.

6. Simulation Study

In this section, we illustrate the computational and statistical efficiency of the online projection estimator in both one-dimensional regression and additive model settings.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

6.1 Generalization Error of the Online Projection Estimator is Rate Optimal

In this section, we use simulated data to illustrate that the generalization error of our estimator reaches the minimix-optimal rate. For each sample, X_i is generated from ρ_X, which has density function is $p_X(x)$; Y_i is generated by $Y_i = f_\rho(X_i) + \epsilon_i$. The details of the parameters are listed in Table 1. In example 1, we purposely select ρ_X such that $\int_0^1 \psi_i(x)\psi_j(x)p_X(x)dx = \delta_{ij}$, together with bounded noise. In example 2, the basis functions are no longer orthogonal w.r.t. ρ_X, and a low signal-noise ratio is applied. In both simple and more realistic scenarios, the online projection estimator achieves rate-optimal statistical convergence.

The f_ρ in example 1 is taken from Dieuleveut and Bach (2016), where they used it to illustrate the performance of the functional SGD estimator; the regression function in example 2 is also used in a study of wavelet neural networks (Alexandridis and Zapranis, 2013).

In example 1, the hypothesis space is the second-order spline on the circle

$$W_2^0(per) = \left\{ f \in L^2([0,1]) \mid \int_0^1 f(u)du = 0 \right.$$
$$\left. f(0) = f(1), f'(0) = f'(1), \int_0^1 \left(f''(u)\right)^2 du < \infty \right\}.$$

In example 2, we use the Sobolev space $W_1^0([0,1])$ defined in (2.4). Because

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

Table 1: Settings of simulation studies. $^*B_4(x) = x^4 - 2x^3 + x^2 - \frac{1}{30}$ is the fourth Bernoulli polynomial, and $\{x\}$ means taking the fractional part of x.

	Example 1	Example 2
Kernel $K(s,t)$	$\frac{-1}{24}B_4(\{s-t\})^*$	$\min\{s,t\}$
Eigenvalue λ_j	$\frac{2}{(2\pi j)^4} = O(j^{-4})$	$\frac{4}{(2j-1)^2\pi^2} = O(j^{-2})$
Basis function $\psi_j(x)$	$\sin(2\pi jx), \cos(2\pi jx)$	$\sqrt{2}\sin(\frac{(2j-1)\pi x}{2})$
$p_X(x)$	$1_{[0,1]}(x)$	$(x+0.5)1_{[0,1]}(x)$
Noise ϵ	Unif([-0.02,0.02])	Normal(0,5)
True regression function f_ρ	$B_4(x)$ $+\cos^2(12x-6)$	$(6x-3)\sin(12x-6)$

the eigenvalues decrease faster in example 1, we observe a convergence rate of $\sim n^{-4/5}$, which is faster than that in example 2, $\sim n^{-2/3}$.

We use $\|\hat{f}_{n,N} - f_\rho\|_2^2$ as a measure of goodness of fit (Figure 1). The proposed method is compared with an online nonparametric SGD estimator (Dieuleveut and Bach, 2016) and the KRR estimator (1.4). Although the KRR might have a better generalization capacity (the rates should be the same, but there might be an improvement in the constant), it is computa-

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

tionally prohibitive to apply it in an online learning setting; thus, we include this method as a reference only. The hyperparameters for each method are chosen to optimize performance (oracle hyperparameters). For our method, this is the constant in front of the timing of adding new basis functions. In Figure 2, we present several typical realizations of $\hat{f}_{n,N}$ for both examples, together with data points.

6.2 CPU Time

Figure 3 shows the CPU time used to calculate the online estimators for up to n samples when solving example 2 for the online projection estimator and the nonparametric SGD estimator. Experiments were run on a computer with one Intel Core M3 processor, 1.2 GHz, with 8 GB of RAM. For the projection estimator, new basis functions are added when $n = \lfloor N^{2\alpha+1} \rfloor$, for $N = 1, 2, ...$ First, for all $\alpha \in \{1, 2, 3\}$, the online projection estimators are all significantly faster to compute than is the nonparametric SGD estimator after $n > 10^4$, because the latter requires evaluating n basis functions for the $(n+1)$st update, which will accumulate very fast. In addition, for larger α, the total computational cost for the online projection estimator becomes nearly linear in n. There are also some "jumps" in the CPU time for the online projection estimator. These correspond to steps in which new basis

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

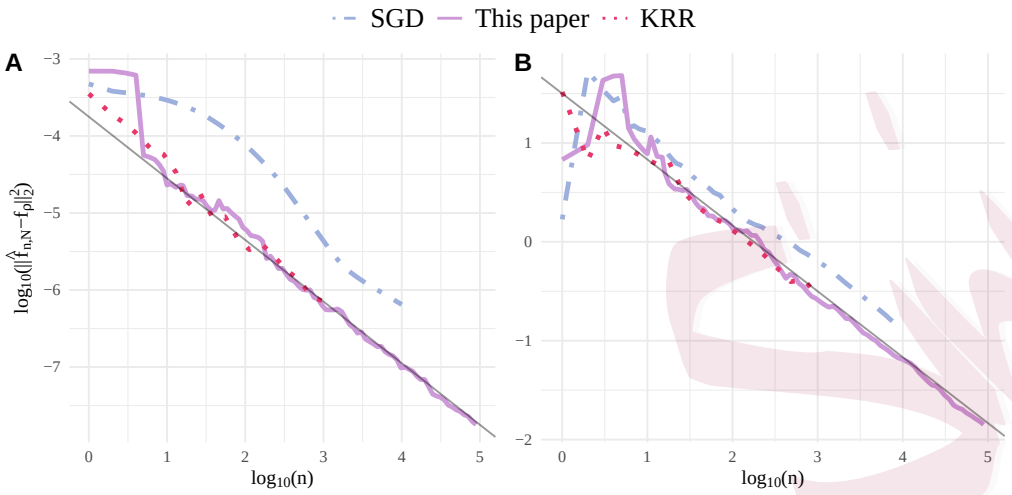

Figure 1: $\log_{10} \|\hat{f}_{n,N} - f_\rho\|_2^2$ against $\log_{10} n$. (**A**) Example 1, the thin black line has a slope $= -4/5$; (**B**) Example 2, slope $= -2/3$. Each curve is calculated as the average of 15 repetitions. Owing to different computational costs, we chose a different maximum n for different methods.

functions are added. Both phenomena match our analysis in Section 3.4. Although it seems beneficial, both computationally and statistically, to use a larger α, it is important to remember that too large a value may result in a poor generalization error. This occurs if the RKHS associated with α becomes so small that it no longer includes f_ρ (see the discussion in Simon and Shojaie (2018)).

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

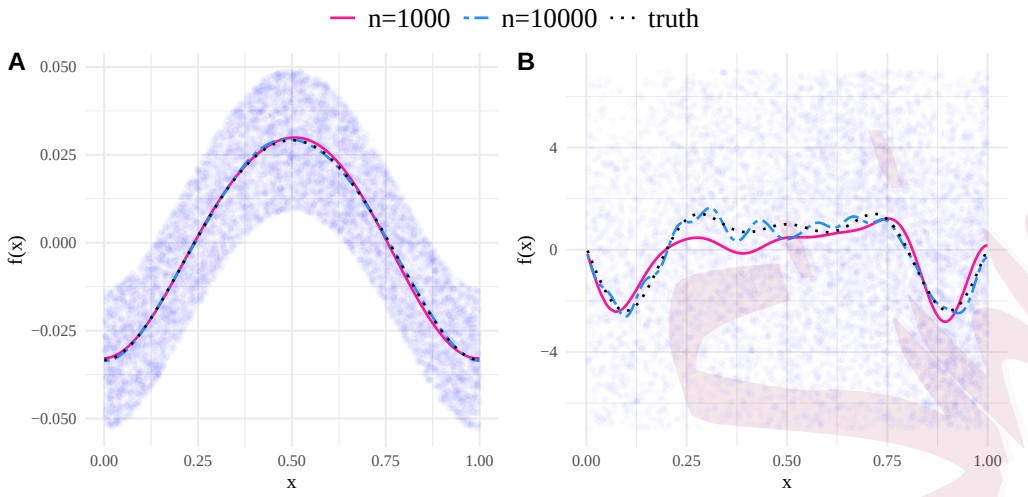

Figure 2: Realizations of $\hat{f}_{n,N}$. (**A**) Example 1; (**B**) Example 2.

7. Discussion

In this paper, we have proposed a framework for constructing online non-parametric regression estimators when the hypothesis space is an RKHS. We showed that (i) the error of the proposed estimator is near-optimal, and (ii) the computational cost of calculating such estimators is much lower than when using other contemporary estimators with similar statistical guarantees. In addition, our estimator is actually precisely an empirical risk minimizer (in a linear space of slowly growing dimension), which allows us to give theoretical guarantees when the noise is heavy tailed (as compared to the previously required assumptions of boundedness).

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

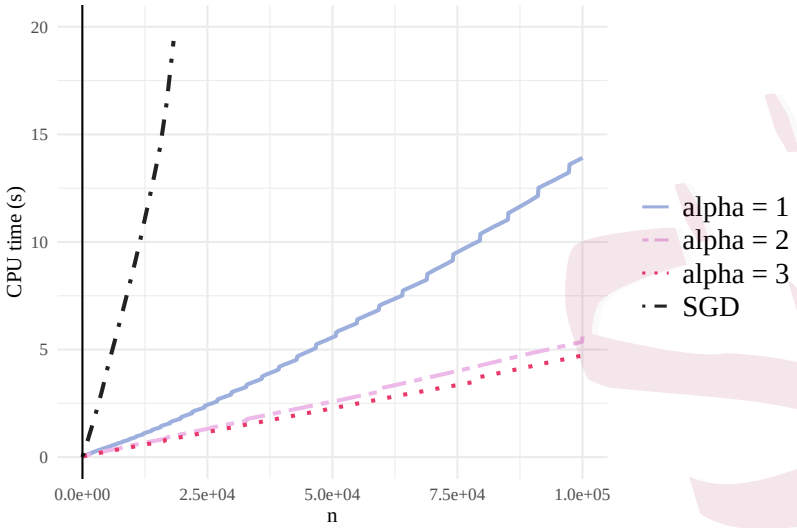

Figure 3: CPU time against sample size (10 runs each curve).

In this work, we leveraged properties of the least-squares loss to efficiently update the empirical risk minimizer $\hat{f}_{n,N}$ in an online manner. However, for a general convex loss function (e.g., logistic regression), the construction of an online nonparametric estimator that has both guaranteed optimal generalization capacity and is computationally feasible for larger problems remains an open question. Although there are functional SGD-type estimators designed for this purpose (see Section 2.1), it would be interesting to design estimators that are both computationally efficient to update and (approximate) ERM in a deterministic space.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

Supplementary Material

In the online Supplementary Material, we provide a proof for Theorem 3. We also describe the settings for the simulations from the main text, and provide additional examples. Furthermore, we include an additional discussion on applications of our estimator.

Acknowledgments

N.S and T.Z. were both supported by NIH grant R01HL137808.

References

Alexandridis, A. K. and Zapranis, A. D. (2013). Wavelet neural networks: A practical guide. *Neural Networks*, 42:1–27.

Babichev, D. and Bach, F. (2018). Constant step size stochastic gradient descent for probabilistic modeling. *stat*, 1050:21.

Bach, F. and Moulines, E. (2013). Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). In *Advances in neural information processing systems*, pages 773–781.

Belloni, A., Chernozhukov, V., and Wang, L. (2014). Pivotal estimation via square-root lasso in nonparametric regression. *The Annals of Statistics*, 42(2):757–788.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

Christmann, A. and Steinwart, I. (2008). Support vector machines.

Cucker, F. and Smale, S. (2002). On the mathematical foundations of learning. *Bulletin of the American mathematical society*, 39(1):1–49.

Dai, B., Xie, B., He, N., Liang, Y., Raj, A., Balcan, M.-F. F., and Song, L. (2014). Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems*, pages 3041–3049.

Dieuleveut, A. and Bach, F. (2016). Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics*, 44(4):1363–1399.

Fasshauer, G. E. and McCourt, M. J. (2015). *Kernel-based approximation methods using Matlab*, volume 19. World Scientific Publishing Company.

Frostig, R., Ge, R., Kakade, S. M., and Sidford, A. (2015). Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory*, pages 728–763.

Gittens, A. and Mahoney, M. W. (2016). Revisiting the nyström method for improved large-scale machine learning. *The Journal of Machine Learning Research*, 17(1):3977–4041.

Han, Q., Wellner, J. A., et al. (2019). Convergence rates of least squares

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

regression estimators with heavy-tailed errors. *Annals of Statistics*, 47(4):2286–2319.

Härdle, W., Kerkyacharian, G., Picard, D., and Tsybakov, A. (2012). *Wavelets, approximation, and statistical applications*, volume 129. Springer Science & Business Media.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Koppel, A., Warnell, G., Stump, E., and Ribeiro, A. (2019). Parsimonious online learning with kernels via sparse projections in function space. *The Journal of Machine Learning Research*, 20(1):83–126.

Kushner, H. and Yin, G. G. (2003). *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media.

Ledoux, M. and Talagrand, M. (2013). *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media.

Leoni, G. (2017). *A first course in Sobolev spaces*. American Mathematical Soc.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

Liu, F., Huang, X., Chen, Y., and Suykens, J. A. (2020). Random features for kernel approximation: A survey in algorithms, theory, and beyond. *arXiv preprint arXiv:2004.11154*.

Lu, J., Hoi, S. C., Wang, J., Zhao, P., and Liu, Z.-Y. (2016). Large scale online kernel learning. *The Journal of Machine Learning Research*, 17(1):1613–1655.

Michel, V. (2012). *Lectures on Constructive Approximation: Fourier, Spline, and Wavelet Methods on the Real Line, the Sphere, and the Ball*. Springer Science & Business Media.

Petersen, K. B. and Petersen, M. S. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15):510.

Raskutti, G., Yu, B., and Wainwright, M. J. (2009). Lower bounds on minimax rates for nonparametric regression with additive sparsity and smoothness. In *Advances in Neural Information Processing Systems*, pages 1563–1570.

Rudi, A. and Rosasco, L. (2017). Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

Santin, G. and Schaback, R. (2016). Approximation of eigenfunctions in kernel-based spaces. *Advances in Computational Mathematics*, 42(4):973–993.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer.

Sherman, J. and Morrison, W. J. (1950). Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127.

Si, S., Kumar, S., and Li, Y. (2018). Nonlinear online learning with adaptive nystr\"{o} m approximation. *arXiv preprint arXiv:1802.07887*.

Simon, N. and Shojaie, A. (2018). Convergence rates of nonparametric penalized regression under misspecified smoothness. *Statistica Sinica Preprint, No: SS-2018-0144*.

Tarres, P. and Yao, Y. (2014). Online learning as stochastic approximation of regularization paths: Optimality and almost-sure convergence. *IEEE Transactions on Information Theory*, 60(9):5716–5735.

Statistica Sinica: Preprint
doi:10.5705/ss.202021.0018

Tsybakov, A. (2008). *Introduction to Nonparametric Estimation*. Springer Science & Business Media.

Wahba, G. (1990). *Spline models for observational data*, volume 59. Siam.

Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press.

Xiu, D. (2010). *Numerical methods for stochastic computations: a spectral method approach*. Princeton university press.

Yuan, M. and Zhou, D.-X. (2016). Minimax optimal rates of estimation in high dimensional additive models. *The Annals of Statistics*, 44(6):2564–2593.

Department of Biostatistics, University of Washington, Seattle, WA 98195, USA

E-mail: zty@uw.edu

Department of Biostatistics, University of Washington, Seattle, WA 98195, USA

E-mail: nrsimon@uw.edu