Modern Sieve Estimators for Nonparametric Problems: Streaming Data and High-dimensional Data

Tianyu Zhang

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:
Noah Simon, Chair
Marco Carone
Alex Luedtke

Program Authorized to Offer Degree:
Biostatistics-Public Health

University of Washington

Abstract

Modern Sieve Estimators for Nonparametric Problems:
Streaming Data and High-dimensional Data

Tianyu Zhang

Chair of the Supervisory Committee:
Dr. Noah Simon
Department of Biostatistics

Estimation of a regression function, linking a set of features to an outcome of interest, is a fundamental statistical task. This dissertation focuses on the application of sieve estimators in modern statistical learning problems. The method of sieves, or estimation via basis expansion, has its roots in Fourier analysis. In the past decades, it has achieved much success in smaller sample size, lower dimensional data science problems. In this dissertation, we will demonstrate its effectiveness in modern statistical learning settings. Sieve estimators can achieve statistical and computational optimality (almost) simultaneously, which makes them very suitable for online and/or large scale nonparametric estimation tasks. Sieve estimators can also be applied to high-dimensional nonparametric problems. They can effectively alleviate the "curse of dimensionality" by leveraging additional structures such as feature sparsity. For each topic covered in this dissertation, we will present both theoretical discussion and a variety of numerical examples.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

People cannot step into the same river twice, and the years of being a graduate student will eventually become a thing of the past. It is my great fortune to be able to study and conduct research in the Department of Biostatistics at UW. Here, I would like to express my gratitude to all those who have supported and encouraged me over the years.

I would like to thank my supervisor, Dr. Noah Simon. This dissertation cannot be completed without him. He has given me a lot of inspiration and help in my dissertation research, writing and presentation. I remember it was in 2018, I came to his office, motivated by his excellent seminar presentation, and told him I wanted to conduct some research that addresses both theoretical and practical concerns. He almost instantly agreed and I got the chance to conduct research under his mentorship, on topics of great interest to me ("fast curve fitting methods"). In my mind, Noah was an imaginative statistician who was able to discover important research questions without being stuck within the "standard" framework. He guided me to integrate statistical theory, machine learning methods and data science problems together (in some unexpected ways). He worked tirelessly to help me revise my thesis (and I am glad that both of us like using parenthesis a lot in writing). He gave me a lot of incisive advice on the basic structures and details of my oral presentation ("imagine you are playing StarCraft ... and the audience has limited attention resource"). His objective and emotional support has helped me overcome many obstacles along the way. I am honored to work with Noah.

I would like to thank my thesis committee members. I would like to thank Dr. Marco Carone, Dr. Alex Luedtke and Dr. Rekha Thomas for their invaluable advice on my research and general/final examinations. Also I would like to thank Dr. Qiyang Han and Dr. Jon

Wellner for answering my questions about their paper. They helped me elucidate the details of the relevant content in Chapter 2 of this dissertation. Also I would like to thank Dr. Lu Xia for her inspiration on the technical treatment in Chapter 4.

I would like to thank Dr. Fang Han from Department of Statistics, for opening the door to machine learning and pattern recognition theory for me.

I would also like to thank the faculty and staff of the Department. Their hard work backed me up. I would like to thank Ms. Gitana Garofalo, Dr. Ken Rice, Dr. Timothy Thornton and Dr. Lurdes Inoue. They gave me support at many critical moments.

Finally, I would like to thank my family and friends. They are an important part of my life. I would like to thank my parents: Lin Zhang and Yongge Yu, who raised me and gave me love and courage. I would also like to thank my cousin Di Wu, who helped me adjust to life in Seattle. I would like to thank Kisaki Takeuchi, she accompanied me through the most difficult period of the epidemic. I would like to thank Kendrick Li, who introduced me to UW Biostatistics. I would like to thank Xiudi Li, Xiaoshun Zeng, Kun Yue, Sijia Li, Yiqun Chen, Edward Zhao, Simeng Zhao, Simo Wang, Haowei Yu, Hao Shen, Zheng Liu and Yifei Luo. I cherish our friendship very much.

人不能两次踏进同一条河流，博士几年裹挟着酸甜苦辣也终将成为往事。我很荣幸能够在华大生统求学、研究。在此，我要向这些年来所有支持、鼓励我的人表达我的谢意。

我首先要对我的导师 Noah Simon 博士致以最真诚的谢意。他在我的论文写作中给予了我大量的灵感与帮助。我第一次在我心中，Noah 是一位富有想象力的统计学家，他能够发现重要的研究问题的同时而不拘泥于“标准”框架内。他指导我将统计学理论，机器学习方法和数据科学问题有机结合。我非常开心我能在他的指导下进行我非常感兴趣的课题的研究，回答我自己自本科研究以来的一些基本的统计学问题。他孜孜不倦地帮助我修改论文，针对我报告中的问题给予了很多鞭辟入里的建议。能和 Noah 一起工作我深感荣幸。

我要向论文委员会成员致以谢意。感谢 Marco Carone 博士，Alex Luedtke 博士和 Rekha Thomas 博士对我的研究的宝贵建议和几年来各方面的支持。同时我要感谢 Qiyang

Han 博士和 Jon Wellner 回答我关于他们的论文的疑惑。他们帮助我理清了论文中第二章中相关内容的细节。同时我要感谢夏璐博士，她给予了我第四章中的技术处理的灵感。

我要感谢华大统计系韩放博士，他为我打开了机器学习和模式识别理论的大门。

我也要感谢华大生统的老师和职员，他们的辛勤付出帮助我解决了很多生活上的后顾之忧。我想感谢 Gitana Garofalo 女士，Ken Rice 博士, Timothy Thornton 博士和 Lurdes Inoue 博士。他们在很多关键的时刻给予了我非常重要的支持。

最后，我要感谢我的家人和朋友。他们是我生命中重要的一环。我要感谢我的父母：张林和于永革，他们养育了我，给予了我爱和勇气。我也要感谢我的表哥吴迪，他帮助我更好地适应了在西雅图的生活。我要感谢竹内纪早，她陪伴我走过了疫情最艰难的时期。我要感谢李祺君，是他让我最初了解到华大生统。我要感谢李秀迪和曾小顺，岳琨, 李思佳, 陈一群和赵京晶，我非常珍视我们之间的友谊。我也要感谢伴我一路走来的朋友们：赵思萌, 王斯墨, 于昊惟, 沈昊, 刘政, 罗逸飞。

DEDICATION

to my parents, Lin Zhang and Yongge Yu.

献给我的父亲和母亲。

Chapter 1

INTRODUCTION

In medical, biological, or epidemiological research, we often aim to build a model to predict an outcome of interest (e.g., systolic pressure, forced expiratory volume of children, or the odds of getting cancer) based on features of the subjects under study (e.g., smoking status, BMI, age, height, the methylation level of certain genes). In many cases, this task reduces to estimating the conditional mean of the outcome given the features — often called regression in statistics. In modern causal inference, even when prediction is not the end-goal, it is common that conditional means must be estimated as a nuisance [58].

Historically, it has been common to assume that the conditional mean falls in some simple, known, parametric class (e.g., that there is a linear relationship between features and outcome). Parametric modeling has provided statisticians much insight into all sorts of problems and is still widely discussed in contemporary statistics research. However, in the field of statistical learning, it has become increasingly common not to make this potentially unrealistic assumption, and instead to use flexible estimators that may be appropriate for more complex non-linear/non-parametric function classes [24]. One common applied modeling strategy, in-line with this idea, is to include transformations of input features in a regression: Often polynomial transformations (or splines) are used. In this case, one needs to specify the degree of polynomial to use. There is a tradeoff: Higher order polynomials allow for more flexible fits but require more data to obtain stable estimates. In practice it is challenging to select the appropriate polynomial degree, however there is theoretical guidance available [119].

More specifically, in nonparametric regression, we often assume that the conditional mean function varies "smoothly" with the features (for some precisely specified type of smooth-

ness). Leveraging this assumption, one can often obtain minimax rate-optimal estimators by employing the aforementioned polynomial regression with a polynomial degree asymptotically specified by the degree of assumed smoothness and number of available observations.

Polynomial regression is a specific case of *sieve estimation*: There we estimate the conditional mean by applying "linear regression" with an enlarged set of predictor variables derived from our original features. The number of derived variables that we choose grows with the available sample size. The transformations that we employ (e.g., polynomial, Fourier) generally depend on the type of smoothness that we assume. For problems with a small number of original features, sieve estimators provide estimates that have good statistical properties (e.g., rate optimality) and are efficient to compute.

In this dissertation we will use the term "sieve" a bit more generally to refer to estimators that can be written as a linear combination of prespecified basis functions. The analytical forms of the basis functions (e.g., monomials, trigonometric functions) *will not* depend on the data, however the number we decide to use will depend on the available sample size.

However, classical sieve estimation procedures struggle in the face of contemporary data challenges. We consider two such challenges: 1) Engaging with high dimensional features; 2) Working with online/streaming data. In the high-dimensional setting, the number of candidate features is large or comparable with the sample size. It is in general much harder to perform nonparametric estimation only under smoothness assumptions: Due to the "curse of dimensionality", we will have difficulty approximating large multivariate functions spaces. For estimation to be tractable, we need to carefully specify our nonparametric function space and adjust our methods accordingly. In the streaming data or online setting, new data are generated continuously and we need to repeatedly update our function estimate whenever new data is available (potentially with every observation). Many nonparametric estimation procedures, including repeatedly fitting sieve estimators are prohibitively computationally expensive. We need nonparametric methods with both computationally and statistically favorable profiles.

In this dissertation, we aim to provide several sieve-type estimation procedures that

are suitable for these modern nonparametric scenarios. In Chapter 2 & 3 we will engage with the online setting and in Chapter 4 we will discuss applying sieve estimation with moderately high-dimensional data. The remaining three chapters contain supplementary materials for Chapter 2 - 4. Specifically, Chapter $n + 3$ is the appendix of Chapter n for $n \in \{2, 3, 4\}$. In this dissertation, different chapters are (inevitably) closely related to each other. The current chronological arrangement may not, unfortunately, be the best presentation. The proposed methods follow the same basic principle as the other chapters but in general are more restrictive than those proposed in the following chapter. Chapter 3 is the "core" chapter of this dissertation. In this chapter we engage with the statistical and computational joint-optimality of online regression problems. In this part of our research work, the nonparametric model we work with is the so-called "Sobolev ellipsoid". Our readers may treat it as a abstraction of the reproducing kernel Hilbert spaces in Chapter 2 or the Sobolev spaces in Chapter 4. More discussion on the intuition of Sobolev ellipsoids can be found in Section 7.2. In Chapter 4, we engage with multivariate function spaces and sieve estimation in these spaces. This part of the work was completed after the previous two chapters. However, the results established in Chapter 4 imply some direct extension of methods proposed in earlier chapters to more general multivariate/high-dimensional settings.

Before we go into individual sections, we would like to provide some very brief formal presentation about the general idea of sieve estimation. Suppose for each subject in our sample, we observe some features $X \in \mathbb{R}^d$ and an outcome of interest $Y \in \mathbb{R}$. We wish to find the function f^0 that best links them, under the independent, identically distributed sample assumptions. When we measure the error by the expected mean-squared distance $E[(Y - f^0(X))^2]$, the minimizer function is called the conditional mean function or regression function. The task of estimating conditional mean is possible if we can draw random "training" samples $(X_i, Y_i) \sim (X, Y)$ *and* when the function f^0 is reasonably "smooth". All the estimators $\hat{f}_n$ proposed in this work, which are also functions, all take the form of

$$\hat{f}_n = \sum_{j=1}^{J_n} \beta_{nj}\psi_j, \text{ for some } \beta_{nj} \in \mathbb{R}, J_n \in \{1, 2, ...\}, \tag{1.1}$$

with some *pre-specified* basis functions ψ_j. The truncation level/ dimension of the estimator J_n and coefficients β_{nj} are determined by the sample $\{(X_i, Y_i)\}$ whereas the basis function are not data-adaptive. For the purpose of both better practical application and theoretical concerns, we are often interested in answering the following questions (with different emphasis across chapters):

- What kind of nonparametric models should we impose on the conditional mean f^0? Under these models, what basis functions should we use?

- Is there any theoretical guidance on how many basis functions J_n we should use? What theoretical guarantees can we get when the "correct" basis number J_n is applied?

- How can we estimate the regression coefficients β_{nj} in a computationally efficient way (Chapter 2 & 3)? How can we modify the estimation procedure to alleviate the negative influence of non-informative features and perform feature selection (Chapter 4)? How much is the computational cost of each strategy?

The answers to some of these questions are standard and straightforward, but some are still relatively open in modern statistical learning settings. We hope our work can help clear some barriers for applying the method of sieves in nonparametric problems and demonstrate the method's effectiveness.

Chapter 2

AN ONLINE PROJECTION ESTIMATOR FOR NONPARAMETRIC REGRESSION IN REPRODUCING KERNEL HILBERT SPACES

2.1 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 some independent and identically distributed (i.i.d.) samples $(X_i, Y_i) \overset{i.i.d.}{\sim} \rho(X, Y)$, for $i = 1, 2, ..., n$, generated from the following statistical model:

$$Y_i = f_\rho(X_i) + \epsilon_i, \tag{2.1}$$

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 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

$$\min_{f \in \mathcal{F}} \mathbb{E}(Y - f(X))^2, \tag{2.2}$$

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

2.1.1 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 [126], general reproducing kernel Hilbert space (RKHS) [20], and Besov space [49]. 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 $x \in \mathcal{X}$,

$$f(x) = \langle f, K_x \rangle_{\mathcal{F}}, \tag{2.3}$$

where K_x is the so-called kernel function associated with $\mathcal{F}$ evaluated at x. This is discussed in more detail in Section 2.2.

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 [128]. It is defined by

$$\hat{f}_n^{KRR} := \underset{f \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - f\left(X_i\right)\right)^2 + \lambda_n^{KRR} \|f\|_{\mathcal{F}}^2, \tag{2.4}$$

where λ_n^{KRR} is a hyperparameter that balances the mean squared error and the complexity of the estimate. Owing to the reproducing property (2.3), $\hat{f}_n^{KRR}$ can be written as a finite linear combination of the kernel function evaluated at $(X_i)_{i=1}^n$ [100].

In general, (2.4) 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) [71] or a Nyström projection [42]. This is discussed further in Section 2.2.1.

2.1.2 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 function as a population minimizer,

$$\min_{\beta \in \mathbb{R}^p} \mathbb{E}[(Y - f_\beta(X))^2]. \tag{2.5}$$

From here, it is popular to directly apply a stochastic gradient descent (SGD) to (2.5), 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 [64, 6, 37, 5].

However, comparatively less attention has been given to online nonparametric regression. A few rate-optimal functional SGD algorithms have been proposed [116, 25], 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 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 2.2.1), the theoretical guarantees are usually not close to optimal, with strong restrictions on the noise variables.

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 [63, 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 the covariate X to be equally spaced or uniformly distributed, as in standard references [119] (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,

$$0 < \liminf_{n\to\infty} \left|\frac{a_n}{b_n}\right| \leq \limsup_{n\to\infty} \left|\frac{a_n}{b_n}\right| < \infty. \tag{2.6}$$

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 chapter, when we say two functions f and g are orthogonal with respect to the measure P, we mean $\int f(x)g(x)dP(x) = 0$.

2.2 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 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 2.2.1. *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 $f \in \mathcal{H}, x \in \mathcal{X}$,*

$$f(x) = \langle f, K_x \rangle_{\mathcal{H}}. \tag{2.7}$$

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 [22], [128], and [33].

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 $\| \cdot \|_{L^2_\nu}$). 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 2.2.2. *Define a Hilbert space*

$$\mathcal{H} = \left\{ f \in L^2_\nu \mid f = \sum_{k=1}^{\infty} \theta_j \phi_j \ \textit{with} \sum_{j=1}^{\infty} \left(\frac{\theta_j}{\sqrt{\mu_j}} \right)^2 < \infty \right\} \tag{2.8}$$

equipped with inner product:

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{\infty} \frac{a_j b_j}{\mu_j}, \tag{2.9}$$

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 2.2.1, see [22]. 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 *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

$$W_1^0([0,1]) = \left\{ f : [0,1] \to \mathbb{R} \mid f(0) = 0 \text{ and } \int_0^1 \left(f'(x)\right)^2 dx < \infty \right\}, \tag{2.10}$$

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

$$\psi_j(x) = \sqrt{2} \sin\left(\frac{(2j-1)\pi x}{2}\right) \quad \lambda_j = \frac{4}{(2j-1)^2\pi^2}. \tag{2.11}$$

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

$$K(x, z) = \sum_{j=1}^{\infty} \lambda_j \psi_j(x) \psi_j(z). \tag{2.12}$$

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 [33, 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.

2.2.1 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

(2.2). 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 $i = 1, 2, ..., n$, [116, 25]:

$$\hat{f}_n^{SGD} = \sum_{i=1}^{n} a_i K_{X_i}. \tag{2.13}$$

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 [105], [74], and [63], the authors choose a subset of features $(K_{X_i})_{i=1}^n$ with cardinality smaller than n. In [23] and [74], kernel-agnostic random Fourier features are used: typically, $O(\sqrt{n})$ basis functions are required in this setting; see [96]. Although computationally more efficient than a vanilla functional SGD (2.13), 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, 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 [98, 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 $L^2_{\bar{\rho}_X}$. More technically,

$$\sup_{\|f\|_{\mathcal{H}}=1} \left\| f - \Pi_{L^2_{\bar{\rho}_X},\, \mathcal{F}_N} f \right\|_{L^2_{\bar{\rho}_X}} = \inf_{V_N \subset L^2_{\bar{\rho}_X}} \sup_{\|f\|_{\mathcal{H}}=1} \left\| f - \Pi_{L^2_{\bar{\rho}_X},\, V_N} f \right\|_{L^2_{\bar{\rho}_X}} = \sqrt{\lambda_{N+1}}, \tag{2.14}$$

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 $L^2_{\bar{\rho}_X}$. 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 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.

2.3 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.

2.3.1 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 $f_\rho \in \mathcal{H}$, 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):

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^N} \sum_{i=1}^{n} (Y_i - \boldsymbol{\theta}^\top \boldsymbol{\psi}^N(X_i))^2. \tag{2.15}$$

The solution can be written in matrix form as

$$\hat{\boldsymbol{\theta}} := (\hat{\theta}_1, ..., \hat{\theta}_N)^\top = (\Psi_n^\top \Psi_n)^{-1} \Psi_n^\top \boldsymbol{Y}_n, \tag{2.16}$$

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

$$\hat{f}_{n,N} = \sum_{j=1}^{N} \hat{\theta}_j \psi_j \tag{2.17}$$

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) [119].

The optimal number of basis functions to use depends on both the sample size n and how fast the eigenvalues λ_j in (2.12) decay. As stated formally in Theorem 2.4.1, 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 [66]). 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 [119] or ρ_X is the uniform distribution [8]. 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.

2.3.2 Naive Online Projection Estimator

The most direct way of extending the projection estimator (2.17) 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 2.1, we provide this naive updating rule for our reader to better understand the proposed method. Our modified proposal in Section 2.3.4 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 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 $(\psi_j)_{j=1}^N$.

Algorithm 2.1: 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)

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

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

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

$N \leftarrow N + 1$

ENDIF

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

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 (2.12). 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 2.1 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.

2.3.3 Efficient Online Projection Estimator

In this section, we explicitly give our proposed method (the details of which are given in Algorithm 2.2). By using some common block/rank-one updating tools from linear algebra, we are able to substantially improve Algorithm 2.1. In particular, it is expensive to repeatedly calculate $(\Psi_n^\top \Psi_n)^{-1}$ 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},

$$\Psi_n = \begin{bmatrix} \Psi_{n-1} \\ \boldsymbol{\psi}_n^\top \end{bmatrix}, \tag{2.18}$$

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

$$\Psi_n^\top \Psi_n = \Psi_{n-1}^\top \Psi_{n-1} + \boldsymbol{\psi}_n \boldsymbol{\psi}_n^\top. \tag{2.19}$$

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 [103].

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

$$\Psi_n = \begin{bmatrix} \Psi_{n-1} & \boldsymbol{\psi}^{N+1} \end{bmatrix}. \tag{2.20}$$

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

$$\Psi_n^\top \Psi_n = \begin{bmatrix} \Psi_{n-1}^\top \Psi_{n-1} & \Psi_{n-1}^\top \boldsymbol{\psi}^{N+1} \\ \left(\boldsymbol{\psi}^{N+1}\right)^\top \Psi_{n-1} & \left(\boldsymbol{\psi}^{N+1}\right)^\top \boldsymbol{\psi}^{N+1} \end{bmatrix}. \tag{2.21}$$

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 [86].

The detailed updating rule of the proposed method is given explicitly in Algorithm 2.2. The basic structure of this algorithm is identical to that of Algorithm 2.1. 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

$$\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + \Phi_n \boldsymbol{\psi}_n \left[Y_n - \hat{f}_{n-1,N}(X_n) \right]. \tag{2.22}$$

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, Section 5.3.

2.3.4 Computational Cost of Algorithm 2.2

We now show that the computational cost of the updating rule in Algorithm 2.2 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 multiplied by an $N \times 1$ vector. Because the $N = \Theta(n^{\frac{d}{2\alpha+d}})$ at step n, the update is of order $n^{\frac{2d}{2\alpha+d}}$.

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.

Algorithm 2.2: 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 $(X_n, N, \Phi_{n-1}, \Psi_{n-1})$

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

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

RETURN (Φ_n, Ψ_n)

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

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

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

$\Psi_n \leftarrow \begin{bmatrix} \Psi_n & \boldsymbol{\psi}^{N+1} \end{bmatrix}$

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

RETURN (Φ_n, Ψ_n)

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

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

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

$N \leftarrow N + 1$

ENDIF

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

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 addition" 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 2.2 is of order $n^{\frac{2d}{2\alpha+d}}$.

2.4 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 [119], 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

$$f_N := \arg\min_{f \in \mathcal{F}_N} \mathbb{E}[(f(X) - f_\rho(X))^2]. \tag{2.23}$$

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 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:

$$\|\epsilon_i\|_{m,1} := \int_0^\infty \mathbb{P}(|\epsilon_i| > t)^{1/m} dt < \infty, \quad \text{for some } m > 1. \tag{2.24}$$

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 [65], 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 2.4.1, 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 [48, Theorem 1]. There are currently no other methodologies, 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 2.4.1 (Optimal convergence rate)**.** *Assume (A1–A4), let $\hat{f}_{n,N}$ be the projection estimator* (2.17)*. Assume that $\|\hat{f}_{n,N}\|_\infty \le M$, for some $M < \infty$. Choosing $N = \Theta(n^{\frac{d}{2\alpha+d}})$, we have*

$$\|\hat{f}_{n,N} - f_\rho\|_2 = O_P\left(n^{-\frac{\alpha}{2\alpha+d}}\sqrt{\log n} \vee n^{-\frac{1}{2}+\frac{1}{2m}}\sqrt{\log n}\right). \tag{2.25}$$

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

$$\mathbb{E}[\|\hat{f}_{n,N} - f_\rho\|_2] = O\left(n^{-\frac{\alpha}{2\alpha+d}}\sqrt{\log n} \vee n^{-\frac{1}{2}+\frac{1}{2m}}\sqrt{\log n}\right). \tag{2.26}$$

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 (2.26), 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 2.4.1, take $\zeta = \alpha/d$), let $B_R = \{f \in \mathcal{H} \mid \|f\|_{\mathcal{H}} \le R\}$ be the R-ball in the RKHS $\mathcal{H}$. Then, we have the minimax bound

$$\liminf_{n\to\infty} \inf_{\hat{f}} \sup_{f_\rho \in B_R} \mathbb{E}\left[n^{\frac{\zeta}{2\zeta+1}}\|\hat{f} - f_\rho\|_2\right] \ge C, \tag{2.27}$$

where the infimum ranges over all possible functions $\hat{f}$ that are measurable of the data. For a derivation of the lower bound, see [128, Chap. 15].

Upper bounds similar to our results in Theorem 2.4.1 have been shown in [116] and [25] 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 2.2.1). 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 [23], [105], [63], and [74]. 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 chapter, we relax the constraint on the noise variables: we require only finite moments of ϵ_i, and show the (in)sensitivity of our bound.

2.5 Multivariate Regression Problems

In most applications, the covariates X_i take values in $\mathbb{R}^d$ where $d > 1$. If the kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ has a known Mercer expansion (2.12), 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 [79, Section 3.5], [136, Section 5.2]. However, as presented in Section 2.4, 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 when d is large (unless, at the same time, a large α is assumed).

A popular low-dimensional structure is the nonparametric additive model [52, 141], 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 (2.1):

$$f_\rho(x) = \sum_{k=1}^{d} f_{\rho,k}\left(x^{(k)}\right), \tag{2.28}$$

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 $\mathcal{H}$ [92]. The proposed online method can be directly generalized to this setting. For further discussion and the empirical performance, see the Supplementary Material, Section 5.4.

2.6 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.

2.6.1 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 2.1. 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 [25], 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 [3].

Table 2.1: 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	$B_4(x)$ $+\cos^2(12x-6)$	$(6x-3)\sin(12x-6)$

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 (2.10). Because 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 2.1). The proposed method is compared with an online nonparametric SGD estimator [25] and the KRR estimator (2.4). 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 computationally 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 2.2, we present several typical realizations of $\hat{f}_{n,N}$ for both examples,

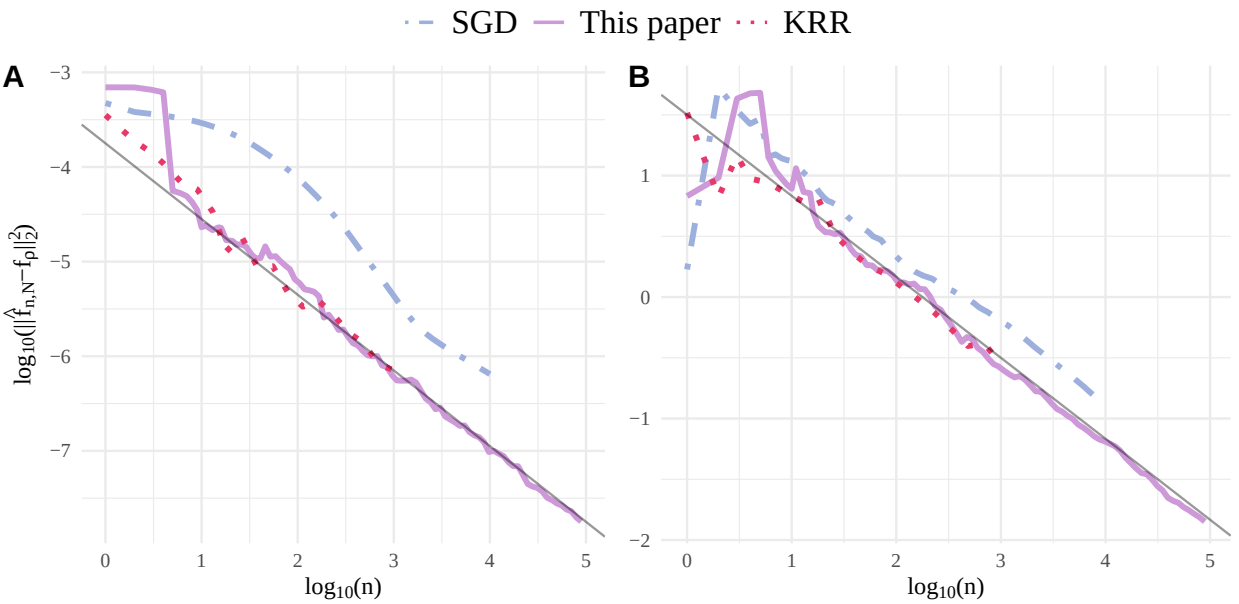

Figure 2.1: $\log_{10} \|\hat{f}_{n,N} - f_\rho\|_2^2$ against $\log_{10} n$. (**A**) Example 1, the thin black line has a slope $= -4/5$; (**B**) Example 2, slope $= -2/3$. Each curve is calculated as the average of 15 repetitions. Owing to different computational costs, we chose a different maximum n for different methods.

together with data points.

2.6.2 CPU Time

Figure 2.3 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 $n > 10^4$,

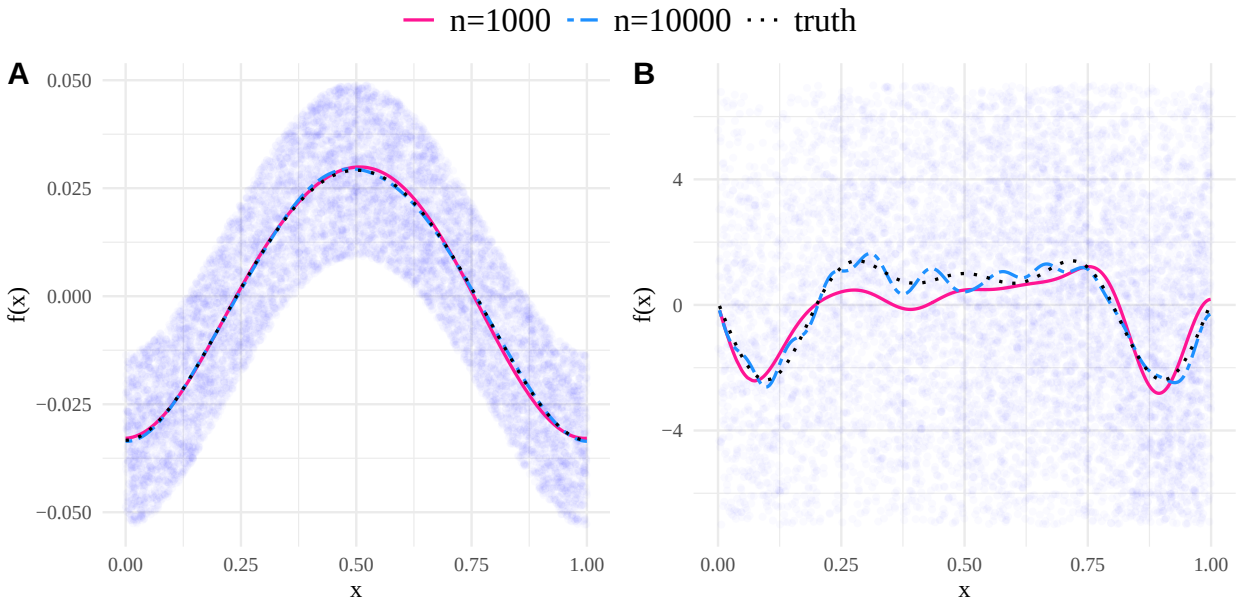

Figure 2.2: Realizations of $\hat{f}_{n,N}$. (**A**) Example 1; (**B**) Example 2.

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 functions are added. Both phenomena match our analysis in Section 2.3.4. 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 [107]).

2.7 Discussion

In this chapter, we have proposed a framework for constructing online nonparametric regression estimators when the hypothesis space is an RKHS. We showed that (i) the error of

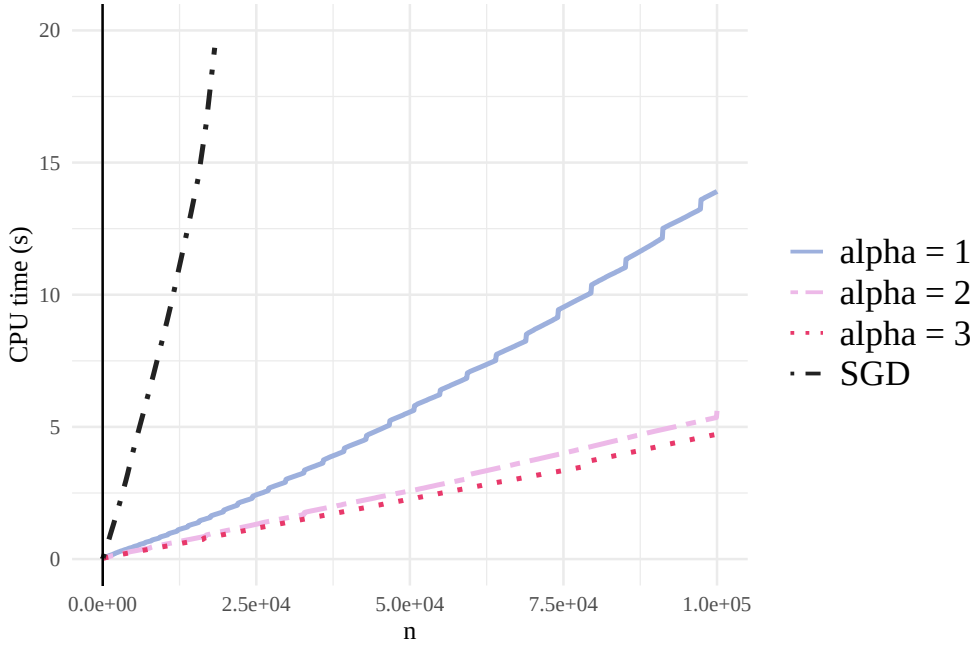

Figure 2.3: CPU time against sample size (10 runs each curve).

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).

In this chapter, 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 2.2.1), it would be interesting to design

estimators that are both computationally efficient to update and (approximate) ERM in a deterministic space.

Chapter 3

A SIEVE STOCHASTIC GRADIENT DESCENT ESTIMATOR FOR ONLINE NONPARAMETRIC REGRESSION IN SOBOLEV ELLIPSOIDS

3.1 Introduction

In this chapter, we will continue our discussion of estimating a condition mean function from noisy sample. Suppose we obtain n samples, (X_i, Y_i), where $X_i \in \mathcal{X} \subset \mathbb{R}^p$ denotes a p-vector of features from the i-th sample we observe, and $Y_i \in \mathbb{R}$ denotes the i-th outcome. Further suppose that each pair (X_i, Y_i) is independently and identically distributed (i.i.d.) from a fixed but unknown distribution ρ over $\mathcal{X} \times \mathbb{R} \subset \mathbb{R}^p \times \mathbb{R}$. A common target of estimation – in order to relate the outcome and the features – is the conditional mean function $f_\rho(X) := E_\rho[Y|X]$. Under extremely mild conditions, this conditional mean is the optimal function for predicting Y from X with regard to mean squared error. More formally,

$$f_\rho = \text{argmin}_{f \in L^2_{\rho_X}} E_\rho \left[(Y - f(X))^2 \right], \tag{3.1}$$

where $L^2_{\rho_X}$ is the collection of all ρ_X-mean square integrable functions and ρ_X is the marginal distribution of X. Our goal is to estimate f_ρ from our collection of observed data.

In order to make a tractable estimation of f_ρ from data, we need to make additional assumptions on its smoothness/structure: The entire $L^2_{\rho_X}$ space is too big to search within [7, 67]. We often formally assume that f_ρ belongs to a pre-specified function space $\mathcal{F} \subsetneq L^2_{\rho_X}$. This $\mathcal{F}$ is known as the *hypothesis space* of the regression problem.

If $\mathcal{F}$ can be indexed by a finite-dimensional parameter set $\Theta \subset \mathbb{R}^d$, $d \in \mathbb{N}^+$, we refer to $\mathcal{F}$ as a *parametric function space* or a *parametric class*. One common parametric class is $\mathcal{F} = \{X^\top \beta \,|\, \beta \in \mathbb{R}^d\}$, the set of all linear functions of X. Parametric classes can impose overly restrictive assumptions on the form of the regression function that may not be realistic in

practice. As such, it has become popular to assume less restrictive structure: It is common to define the hypothesis space based on constraints on derivatives, monotonicity, or other shape-related properties. Such an $\mathcal{F}$ is most naturally written as an infinite-dimensional subset of $L^2_{\rho_X}$. Commonly used examples of $\mathcal{F}$ in the statistics community include Hölder balls, Sobolev spaces [45, 81, 126], reproducing kernel Hilbert spaces (RKHS) [10, 20] and Besov spaces [49]. These are known as *nonparametric function spaces*, as they cannot naturally be parametrized using a finite length vector. The Sobolev ellipsoid, in particular, is a simple and useful abstraction of many important function spaces [126]. Therefore, we focus on them exclusively as the hypothesis spaces in this chapter.

In this chapter, we propose an estimator for *online* nonparametric regression. In online estimation, the data are seen sequentially, one sample at a time. After each sample is observed, our estimate of f_ρ must be updated, as a prediction may be required at any point in time before all the available samples are processed. In an online problem with n observations, we must sequentially construct n estimates. This is in contrast to the classical batch learning setting where we collect all the data initially and perform estimation only once. In the online setting, it is generally computationally infeasible to repeatedly refit the whole model from scratch for each new observation. Thus, online algorithms are generally carefully developed to permit more tractable *updates* after each new observation [27, 64].

An ideal estimator in online settings should be: i) statistically rate-optimal, i.e. achieve the minimax-rate for estimating f_ρ over $\mathcal{F}$; and ii) computationally inexpensive to construct/update. In this chapter, we present such an online nonparametric estimator for use when the hypothesis space is a Sobolev ellipsoid, which we term the *Sieve Stochastic Gradient Descent estimator (Sieve-SGD)*. This method can be thought of as an online version of the classical projection estimator [119], where the latter is a specific example of sieve estimators [120, 102]. We use the more general term "sieve" in naming our method to emphasize its nonparametric nature and avoid confusion with the term "stochastic projection" [124]. We will show that Sieve-SGD can achieve rate-optimal estimation error for $\mathcal{F}$ a Sobolev ellipsoid and asymptotically uses minimal memory (up to a log factor) among all rate-optimal esti-

mators. In addition, our estimator has the same computational cost (up to a constant) as merely examining each allocated memory location every time a new sample X_i is collected. This intimates that in scenarios when our estimator has near optimal space complexity, it may also have near optimal time complexity (though formal investigation of lower bounds for time complexity in this problem is beyond the scope of the dissertation).

The structure of this chapter continues as follows. In Section 3.2 we briefly cover classical results for batch, nonparametric estimation in Sobolev ellipsoids, focusing on projection estimators (which motivate our method). In Section 3.3 we return to the online setting and explore intuition for how one might combine projection estimation and stochastic gradient descent (SGD) [12]. The latter is a well-studied method that has been applied fruitfully to online parametric regression problems. This will help motivate our proposed method, which, as we will see, can be thought of as an SGD estimator with a parameter space of increasing dimension. In Section 3.4 we discuss existing nonparametric SGD estimators, and identify some notable drawbacks of current methods. In Section 3.5, we introduce the formal construction of Sieve-SGD and analyze its computational expense. From there, we show that our estimator has a dramatically smaller "dimension" than existing methods and discuss how this helps to reduce the computational expense. In Section 3.6, we give a theoretical analysis of the statistical properties of Sieve-SGD. In constructing our estimator, we need to decide how quickly to grow the dimension it projects onto. Under minimal assumptions, we characterize the required growth rate and learning rate for our estimator to be statistically and computationally (near) optimal. We will also investigate under what conditions such an optimality result is adaptive/insensitive to our choice of the "dimension-specific learning rate". Section 3.7 provides simulation studies to illustrate our theoretical results. Finally, in Section 3.8, we have some further discussion of Sieve-SGD and possible future research directions.

Notation: In this chapter, we use C to denote a generic constant that does not depend on sample size n (The value of C may be different in different parts of the chapter). Additionally the notation $a_n = \Theta(b_n)$ means $a_n = O(b_n)$ and $b_n = O(a_n)$. The function $\lfloor x \rfloor$ maps x to

the largest integer smaller than x. For a vector $x \in \mathbb{R}^p$, $x^{(i)}$ is the i-th component of x. The notation $x \vee y$ (resp. $x \wedge y$) is shorthand for $\max\{x, y\}$ (resp. $\min\{x, y\}$). The $\|\cdot\|_\infty$ norm of a continuous function f is defined as $\|f\|_\infty = \sup_{x\in\mathcal{X}} |f(x)|$, where $\mathcal{X}$ is the domain of f.

3.2 Batch Learning and the Projection Estimator

In this section we consider estimation in the classical batch setting where our estimate is constructed once after all n samples are observed. We will begin by formally introducing a Sobolev ellipsoid: This is the hypothesis space we will use throughout this chapter. This will be followed by presenting the classical projection estimator [119].

Consider a user-specified measure ν whose support contains $\mathcal{X}$, and the corresponding square-integrable function space L^2_ν. In many interesting cases ν can be simply taken as Lebesgue measure over $\mathcal{X}$ but it is not necessary in the general form of our theory. To define a Sobolev ellipsoid in L^2_ν, suppose we have a complete orthonormal basis $\{\psi_j, j = 1, 2, ...\} \subset L^2_\nu$ of L^2_ν [54]. This means

i) For any $f \in L^2_\nu$, there exists a unique sequence $(\theta_j)_{j=1}^\infty \in \ell^2$ such that

$$\lim_{N\to\infty} \int \left| f(z) - \sum_{j=1}^{N} \theta_j \psi_j(z) \right|^2 d\nu(z) = 0 \quad \text{(completeness)} \tag{3.2}$$

where ℓ^2 is the space of square convergent series.

ii) $\{\psi_j\}$ is an orthonormal system:

$$\int \psi_i(z)\psi_j(z) d\nu(z) = \delta_{ij} \quad \text{(orthonormality)} \tag{3.3}$$

where δ_{ij} is the Kronecker delta.

We define the *Sobolev ellipsoid* $W(s, Q, \{\psi_j\})$ as:

$$W(s, Q, \{\psi_j\}) := \left\{ f = \sum_{j=1}^{\infty} \theta_j \psi_j \;\middle|\; \sum_{j=1}^{\infty} (\theta_j j^s)^2 \le Q^2 \right\} \tag{3.4}$$

We refer to $(\theta_j)_{j=1}^{\infty}$ as the (general) *Fourier coefficients* of a function f. Throughout this chapter, we assume the measure ν, basis functions ψ_j and the regularity parameter s are all known. When it is clear which ψ_j we are using, we will denote a Sobolev ellipsoid simply by $W(s, Q)$. We may also use the further simplified notation $W(s)$ because the diameter Q usually plays a secondary role in our theoretical analysis and the proposed method is adaptive to it. Intuitively, by saying a function f belongs to a Sobolev ellipsoid, we are requiring its coefficients $\{\theta_j\}$ to converge to zero faster than $j^{-(s+1/2)}$ (if not, the sum $\sum_{j=1}^{\infty} (\theta_j j^s)^2$ would diverge to infinity). The larger s is, the faster the decay of θ_j will be, and thus the stronger our assumption is.

Sobolev ellipsoids are popular spaces to study for two reasons: 1) They impose a useful structure for theory and computations, especially as a basic example of hypothesis spaces with finite metric entropy; and 2) Many natural spaces of regular functions are Sobolev ellipsoids. For example, if $\mathcal{X} = [0, 1]$ with ν as Lebesgue measure, then for any $s > 0$, the periodic Sobolev space

$$\mathcal{F} = \left\{ f \in L_\nu^2 \mid \int \left(f^{(s)}(x)\right)^2 dx < Q^2, f^{(k)}(0) = f^{(k)}(1), k = 0, 1, .., s-1 \right\} \tag{3.5}$$

can be written as a Sobolev ellipsoid, using an orthogonal basis of trigonometric functions [126, Chapter 2]. More generally, for many important RKHSs $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$, it is possible to find a set of ψ_j such that $W(s, Q, \{\psi_j\}) = \{f \in \mathcal{H} \mid \|f\|_{\mathcal{H}} \leq Q\}$, i.e. a ball in an RKHS is a Sobolev ellipsoid (see [22, 113]): Under mild conditions [109], a Mercer kernel $K(s, t) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ has the following Mercer representation:

$$K(s, t) = \sum_{j \in \mathcal{J}} \lambda_j \psi_j(s) \psi_j(t), \tag{3.6}$$

where $\lambda_j > 0$, $\mathcal{J}$ is at most countably infinite. And $\{\psi_j\}$ is an orthonormal system (in L_ν^2) w.r.t. some measure ν on $\mathcal{X}$, and any function $f \in \mathcal{H}$ can be written as $f = \sum_{j \in \mathcal{J}} \theta_j \psi_j$. It is also known that the RKHS-norm can be identified as $\|f\|_{\mathcal{H}}^2 = \sum_{j \in \mathcal{J}} \theta_j^2 \lambda_j^{-1}$. So a ball in the RKHS, i.e. $\{f \in \mathcal{H} \mid \|f\|_{\mathcal{H}} \leq Q\}$, is the same as a Sobolev ellipsoid spanned by $\{\psi_j\}$ when $\mathcal{J} = \mathbb{N}^+$ and $\lambda_j = j^{-2s}$. This is the case for many Sobolev-type kernels (for example,

p.454 in [33]). When $\mathcal{J}$ is finite dimensional (polynomial kernels) or λ_j decays exponentially fast in j (Gaussian kernel, p.455 in [33]), a ball in the RKHS can be characterized as some "generalized" Sobolev ellipsoid.

In everything that follows we will assume that f_ρ, our target of estimation, lives in a known Sobolev ellipsoid $W(s, Q, \{\psi_j\})$; with $\{\psi_j\}$ specified, and orthonormal w.r.t. a specified measure ν (not necessarily equal to ρ_X); and s known (we allow Q to be unknown).

The *Projection Estimator* is a classical estimator naturally associated with a Sobolev ellipsoid. We can treat it as a special case of general sieve estimation [120, Chapter 10]: The estimates can be characterized by a sequence of finite dimensional linear spaces of increasing dimension (the dimension increases with sample size). For any given $f \in W(s, Q)$, the magnitude of its Fourier coefficients must asymptotically decrease with j fast enough. Thus, it might be sensible to consider an estimator that discards the basis functions far into the tail. This is precisely what the projection estimator does. More formally, for a user-specified truncation level J_n, the projection estimator is given by

$$\hat{f}_{n,J_n} = \sum_{j=1}^{J_n} \hat{\theta}_j \psi_j \tag{3.7}$$

where $\hat{\theta} = (\hat{\theta}_1, .., \hat{\theta}_{J_n})^\top$ is the solution of the least square problem:

$$\min_{\theta \in \mathbb{R}^{J_n}} \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{J_n} \theta_j \psi_j(X_i) \right)^2 \tag{3.8}$$

It has been shown (e.g. [119], Theorem 1.9) that when we choose $J_n = \Theta(n^{\frac{1}{2s+1}})$, the projection estimator is a rate-optimal estimator over $W(s, Q)$, i.e.

$$\limsup_{n \to \infty} \sup_{f_\rho \in W(s,Q)} E\left[\left\| \hat{f}_{n,J_n} - f_\rho \right\|_2^2 \right] = O(n^{-\frac{2s}{2s+1}}) \tag{3.9}$$

This result is usually shown in the literature for X_i equally spaced, or drawn from a uniform distribution. But in our theoretical analysis (Section 3.6), we allow ρ_X to be a much more general distribution.

Sieve-SGD is inspired by this (batch) projection estimator. The key here is that the number of basis functions we need to use can be dramatically smaller than the sample size,

and their analytical forms do not depend on the data (usually reproducing kernel methods use basis functions "centered" at the feature vectors X_i). This possibility has been rarely explored by existing nonparametric online estimation research.

3.3 Online Learning and Stochastic Approximation

We now move to the online learning setting where observations are collected sequentially from a data stream, and an estimate of our function is required after each sample. Such an infinite data stream may really exist, for example, with simulated samples as in reinforcement learning. Or the stream may serve as an abstraction used with large-scale data sets where it is not favorable to handle all the samples at once. It is generally computationally prohibitive to use a method developed for the "batch" setting and completely refit it after each observation. Instead methods that iteratively update are preferred. For example, fitting a single projection estimator (solving (3.8)) with n observations using $J_n = n^{\frac{1}{2s+1}}$ requires computation of $\Theta(n^{1+\frac{2}{2s+1}})$. Refitting a projection estimator (from scratch) after each observation $i = 1, \ldots, n$ with $J_i = \lfloor i^{\frac{1}{2s+1}} \rfloor$ would require an accumulated computation of $\sum_{i=1}^{n} i^{1+\frac{2}{2s+1}} = \Theta(n^{2+\frac{2}{2s+1}})$. This scales worse than quadratically in n. Our goal in the online nonparametric setting is to find a statistically rate-optimal estimator whose computation scales only slightly worse than linearly in n.

Online learning has been thoroughly studied for parametric $\mathcal{F}$. Many proposed methods are based on the concept of *stochastic approximation* [64]. One of the most popular methods in stochastic approximation is *Stochastic Gradient Descent* (SGD) [12]. In the parametric setting, SGD gives a statistically rate-optimal estimator $\hat{f}_n$ whose population mean-square-error $E\|\hat{f}_n - f_\rho\|^2_{L^2_{\rho_X}}$ is of order $O(n^{-1})$ [5, 6, 37]. Both vanilla SGD and its variants have been applied to general convex loss functions and are shown to be statistically rate-optimal under mild conditions [27, 82].

3.3.1 Parametric SGD

To motivate stochastic optimization in the nonparametric setting, we first give more details on SGD for parametric classes. Here we consider a specific class of functions $\mathcal{F} = \{f = \sum_{j=1}^{d} \beta^{(j)}\psi_j, \beta \in \mathbb{R}^d\}$ for a set of pre-specified basis functions $\psi_j : \mathbb{R}^p \to \mathbb{R}, j = 1, \ldots, d$. We use this example to illustrate the principle of (parametric) SGD. Solving $\text{argmin}_{f\in\mathcal{F}} E\left[(Y - f(X))^2\right]$ reduces to solving

$$\min_{\beta\in\mathbb{R}^d} \ell(\beta) := \min_{\beta\in\mathbb{R}^d} E\left[\left\{Y - \sum_{j=1}^{d} \beta^{(j)}\psi_j(X)\right\}^2\right] \tag{3.10}$$

We assume the minimizer of $\ell(\beta)$ exists and denote it as β^*.

If we knew the true joint distribution ρ of (X, Y) (which never happens in practice), then equation (3.10) is just a numerical optimization problem which does not involve data. We could use gradient descent to solve it. The gradient of ℓ at any point β is

$$\nabla\ell(\beta) = -2E\left[\left\{Y - \sum_{j=1}^{d} \beta^{(j)}\psi_j(X)\right\}\{\psi_1(X), ..., \psi_d(X)\}^\top\right] \tag{3.11}$$

Thus, the gradient descent updating rule one could use is:

$$\begin{aligned} \hat{\beta}_0 &= 0 \\ \hat{\beta}_n &= \hat{\beta}_{n-1} - \gamma_n \nabla\ell(\hat{\beta}_{n-1}) \end{aligned} \tag{3.12}$$

where $\{\gamma_n\}$ is a pre-specified sequence of step-sizes (or learning rate) and $\hat{\beta}_n \in \mathbb{R}^d$ is the sequence of approximations of β^*.

In practice, we do not know the joint distribution ρ: we must use data to estimate β^*. In the framework of SGD, this is done by using the data to get unbiased estimates of the gradients and substituting the estimates into our updating rule (3.12). In particular we note that $\widehat{\nabla\ell(\beta)} := -2\left(Y_i - \sum_{j=1}^{d} \beta^{(j)}\psi_j(X_i)\right)(\psi_1(X_i), ..., \psi_d(X_i))^\top$ is an unbiased estimator of the gradient $\nabla\ell(\beta)$ based on one sample.

This results in the SGD updating rule.

$$\begin{aligned}\hat{\beta}_0 &= 0 \\ \hat{\beta}_n &= \hat{\beta}_{n-1} - \gamma_n \widehat{\nabla \ell(\hat{\beta}_{n-1})} \\ &= \hat{\beta}_{n-1} + 2\gamma_n \left(Y_n - \sum_{j=1}^{d} \hat{\beta}_{n-1}^{(j)} \psi_j(X_n) \right) (\psi_1(X_n), ..., \psi_d(X_n))^\top\end{aligned} \tag{3.13}$$

So our estimator $\hat{f}_n$ of f_ρ has the following functional update rule, derived from (3.13):

$$\hat{f}_n = \hat{f}_{n-1} + 2\gamma_n \left(Y_n - \hat{f}_{n-1}(X_n) \right) \sum_{j=1}^{d} \psi_j(X_n)\psi_j. \tag{3.14}$$

Here we have shifted to considering our estimator $\hat{f}_n$ as a function, rather than thinking about $\hat{\beta}_n$ a vector of coefficients. This will be important in the nonparametric setting.

3.3.2 From parametric SGD to nonparametric SGD

In this section we discuss the intuition in moving from SGD in a finite dimensional parametric space to an infinite dimensional space.

We assume $f_\rho \in W(s, Q, \{\psi_j\}) \subset L^2_\nu$. Since ψ_j is a complete basis of L^2_ν, we can always find an expansion of f_ρ w.r.t. $\{\psi_j\}$:

$$f = \sum_{j=1}^{\infty} \theta_j \psi_j. \tag{3.15}$$

In section 3.3.1, we already discussed the SGD updating rule for a d-dimensional model $f(X) = \sum_{j=1}^{d} \beta^{(j)} \psi_j(X)$. In the nonparametric scenario, the number of basis function is increased from d to infinity: This causes problems if care is not taken.

One might naturally consider applying a direct analog to the finite-dimensional SGD rule (3.14) here (we omit the constant 2):

$$\hat{f}_n = \hat{f}_{n-1} + \gamma_n \left(Y_n - \hat{f}_{n-1}(X_n) \right) \sum_{j=1}^{\infty} \psi_j(X_n)\psi_j. \tag{3.16}$$

Unfortunately we run into a severe problem: The series $\sum_{j=1}^{\infty} \psi_j(X_n)\psi_j$ does not converge even if all ψ_j are bounded (it is direct to check when $X_n = 0$ and ψ_j are trigonometric

functions). However, as we assume $f_\rho \in W(s)$, we know that those higher order components, ψ_j, $j \gg 1$ should have very small coefficients. Thus, one natural solution is to use a different step size per component, that decreases as j increases. By doing "less fitting" for larger j, we can stabilize our update (smaller variance), and yet might still appropriately fit the overall regression function. In particular one might modify (3.16) to

$$\hat{f}_n = \hat{f}_{n-1} + \gamma_n \left(Y_n - \hat{f}_{n-1}(X_n) \right) \sum_{j=1}^{\infty} t_j \psi_j(X_n) \psi_j, \tag{3.17}$$

where the component-specific (or dimension-specific) learning rates $t_j > 0$ are monotonically decreasing with j. For t_j decreasing fast enough and uniformly bounded ψ_j, the function series $\sum_{j=1}^{\infty} t_j \psi_j(X_n)\psi_j$ is absolutely convergent. Now (3.17) becomes a sensible nonparametric SGD updating rule when the hypothesis space is a Sobolev ellipsoid. In addition, sometimes $\sum_{j=1}^{\infty} t_j \psi_j(X_n)\psi_j$ actually has a simply characterized closed form (in particular, for many RKHS). In such cases, (3.17) results in a relatively straightforward algorithm. More specifically, one can show that when $t_j = j^{-2s}$ and $\gamma_n = \Theta(n^{-\frac{1}{2s+1}})$, the average

$$\bar{f}_n := \frac{1}{n} \sum_{i=1}^{n} \hat{f}_i \tag{3.18}$$

is a rate-optimal estimator of $f_\rho \in W(s)$. This was recently proposed (though motivated quite differently) in the context of RKHS hypothesis spaces [25]. The authors there engage directly with the *kernel function* for the RKHS (though their updating rule is equivalent to eq (3.17)). This will be discussed in more detail in Section 3.4. Our work engages and extends these ideas (in combination with sieve estimation) to form a statistically rate-optimal online estimator with greatly reduced computational and memory complexity.

3.4 Related work

Nonparametric online learning is a relatively new area. A few remarkable functional stochastic approximation algorithms have been proposed in the last two decades [15, 25, 77, 116, 137]. The key ideas in that body of work are intimately related to those mentioned in Section 3.3.2,

however, they engage those ideas from a different direction: They assume that the hypothesis function space $\mathcal{F}$ is an RKHS, and then leverage the kernel in that space. In particular, the RKHS structure makes it possible to take the gradient of the evaluation functional $L_x(f) := f(x)$, with respect to the RKHS inner product $\langle \cdot, \cdot \rangle_K$, i.e.

$$L_x(f + \epsilon g) = f(x) + \epsilon g(x) = L_x(f) + \epsilon \langle g, K_x \rangle_K. \tag{3.19}$$

Thus, $K_x(\cdot) := K(x, \cdot) \in \mathcal{F}$ is the gradient of functional L_x at f. However, one cannot do this in the general $L^2_{\rho_X}$ space where the evaluation functional is no longer a bounded operator.

Thus when $\mathcal{F}$ is an RKHS associated with kernel K, there is a simple nonparametric SGD updating rule for minimizing $E[(Y - f(X))^2]$ over $\mathcal{F}$:

$$\begin{aligned} \hat{f}_0 &= 0 \\ \hat{f}_n &= \hat{f}_{n-1} + \gamma_n \left(Y_n - \hat{f}_{n-1}(X_n) \right) K(X_n, \cdot) \end{aligned} \tag{3.20}$$

Here, because the gradient is taken with respect to the RKHS inner product, we do not have the issue encountered in (3.16) where our series representation of the "gradient" actually did not converge. In fact, by working with the RKHS inner-product, we implicitly carry out the proposal of Section 3.3.2 and decrease the component-specific learning rate of higher order terms. More specifically, we usually have the Mercer expansion of the kernel function:

$$K(x, z) = \sum_{j=1}^{\infty} t_j \psi_j(x) \psi_j(z), \tag{3.21}$$

with respect to an orthonormal basis $\{\psi_j\}$ of L^2_ν. For many common RKHS, we have $t_j = \Theta(j^{-u})$ for some $u > 1$ [33, Appendix A]. Thus, (3.20) corresponds precisely to the previously discussed update (3.17). Most popular RKHS have a kernel $K(x, z)$ with a closed form representation, and thus, rather than having to store an infinite number of coefficients, after n steps the estimate from (3.20) would take the form of a weighted linear combination of n kernel functions [25]:

$$\hat{f}_n = \sum_{i=1}^{n} b_i K(X_i, \cdot). \tag{3.22}$$

Although such estimators (with one more Polyak averaging step (3.18)) have been shown to give rate-optimal MSE [25], updating them with a new observation (X_{n+1}, Y_{n+1}) usually involves evaluating n kernel functions at X_{n+1}, with computational expense of order $\Theta(n)$. This is in contrast with the constant update cost of $\Theta(d)$ in parametric SGD, where d is the dimension of the parameter space. Thus, the time expense of nonparametric kernel SGD will accumulate at order $\Theta(n^2)$. Also, one is required to store the n feature-values $\{X_i\}_{i=1}^n$ to evaluate the estimator which results in $\Theta(n)$ space expense. This relatively large time and space complexity indicates that those kernel-based SGD estimators are not ideal as methods that are nominally designed to deal with large data sets.

There has been some work in the literature aimed at improving the computational aspects of kernel SGD methods [105, 74, 63]. These methods select a subset of the n kernel functions centered at the feature vectors and use them as basis functions to construct estimators (which is also related to Nyström projection). Neither the statistical performance nor the computational expense of the aforementioned work is guaranteed to be optimal. Also, the theoretical analysis in that work typically requires the noise variable to have extremely light tails.

There has also been recent work [15, 77] aimed at improving kernel SGD algorithms by leveraging approximate second order information (SGD only uses the first order information). The estimator in [77] is shown to give rate-optimal MSE and have better (theoretical) computational efficiency than the vanilla kernel SGD mentioned above. However, these algorithms are usually dramatically more complicated and have a couple of hyper parameters that need to be tuned.

There is another branch of research also called "online nonparametric regression" that engages with a different but related setting [38, 88]. They do not aim to directly minimize the (population) generalization error. Their definition of "regret" is based on comparing a running average of prediction error and the empirical risk minimizer's training error. Formally, it is defined as $\sum_{i=1}^n l\left(\widehat{Y}_i, Y_i\right) - \inf_{f\in\mathcal{F}} \sum_{i=1}^n l\left(f\left(X_i\right), Y_i\right)$, where $\hat{Y}_i$ is the prediction of the algorithm based on the first $i-1$ observations, l is a convex loss and $\mathcal{F}$ is the hypothesis

function space. While this is an interesting area of research, and might be used to engage with population generalization error (using online-to-batch techniques), it is a less direct treatment than what we are considering in this chapter.

3.5 Online Learning and the Projection Estimator: Sieve-SGD

In this section, we combine ideas from the projection estimator (in the batch learning setting), and stochastic gradient descent to develop an estimator that is suitable for online nonparametric regression. The estimator we will propose achieves the minimax rate for MSE over a Sobolev ellipsoid, and is much more computationally efficient than standard kernel SGD methods.

As a reminder, the kernel SGD estimator based on (3.20) has minimax rate optimal MSE. When $\sum_{j=1}^{\infty} t_j \psi_i(s)\psi_j(t)$ has an available closed form, it requires $\Theta(n)$ memory and has $\Theta(n^2)$ time expense for sequentially processing n observations. We aim to improve this and furthermore to propose an effective estimator appropriate for cases where $\sum_{j=1}^{\infty} t_j \psi_i(s)\psi_j(t)$ has no closed form.

Motivated by the projection estimator, we opt to use truncated series in the updating rule, modifying (3.17) (or equivalently (3.20)) to get

$$\hat{f}_n = \hat{f}_{n-1} + \gamma_n \left(Y_n - \hat{f}_{n-1}(X_n)\right) \sum_{j=1}^{J_n} t_j \psi_j(X_n)\psi_j \tag{3.23}$$

Here J_n is an increasing sequence of integers that grows as we collect more observations. When J_n is larger, the updating rule (3.23) is closer to our original form (3.17); however, a smaller J_n is desirable because it results in a lower computational expense. Part of our task is identifying a "minimal" J_n that still maintains favorable statistical properties.

It turns out there are two ways to control the bias-variance tradeoff. One can use the truncation level J_n, or the component specific step sizes t_j. If the truncation level is used, then the methodology is more analogous to a projection estimator. In this case, so long as t_j is not too large (controlling the variance in the dynamics of SGD) or too small (controlling the bias term), we would get (near) optimal statistical performance for a relatively wide

range of choices for t_j. We give formal results related to this in Section 3.6.3. If, instead, we control the bias-variance tradeoff using t_j then our estimator is more analogous to kernel-SGD. In this case, the first order terms for bias and variance are determined by the sequence $\{t_j\}$ and J_n just needs to be sufficiently large (such that we do not induce excess bias). We give formal results for this in Section 3.6.2. This second way to control the tradeoff is similar to using a truncated basis for penalized regression in the batch learning setting. For example, in [46] and [133, Section 5.2], the authors propose to estimate f_ρ by solving a penalized regression spline problem, where they use a reduced spline basis for improved computation (rather than including a knot at every point). The bias/variance trade-off there is controlled via the penalty: They are careful to include enough basis elements so that the use of a reduced basis only contributes a second order term to the bias.

We will next give details of our proposal. For this proposal we are assuming that $f_\rho \in W(s, Q, \{\psi_j\}) \subset L^2_\nu$, and that s is known. Based on this, we choose our component-specific step-sizes as $t_j = j^{-2\omega}$ (for some $1/2 < \omega \leq s$). We also define

$$K^\omega_{x,J_n}(\cdot) = \sum_{j=1}^{J_n} j^{-2\omega}\psi_j(x)\psi_j(\cdot). \tag{3.24}$$

In addition to simplifying exposition, this notation relates our method to (3.21). The function $K^\omega_{x,J_n}(\cdot)$ can be seen as a truncated approximation of the kernel function

$$K^\omega_{x,\infty}(\cdot) = \sum_{j=1}^{\infty} j^{-2\omega}\psi_j(x)\psi_j(\cdot) \tag{3.25}$$

that drops all the ψ_j with index $j > J_n$.

3.5.1 Sieve Stochastic Gradient Descent

We now explicitly give our Sieve Stochastic Gradient Descent algorithm (Sieve-SGD) for estimation of f_ρ in a Sobolev ellipsoid $W(s, Q, \{\psi_j\})$.

Let $J_n = \lfloor n^\alpha \rfloor$ for some specified $\alpha > 0$ and $\omega \in (\frac{1}{2}, s]$. The parameter α is usually taken between $\frac{1}{2s+1}$ and 1. We use γ_i to denote the step size (learning rate) of the i-th update and typically choose $\gamma_i = \Theta(i^{-\frac{1}{2s+1}})$.

Proposed Algorithm: Sieve Stochastic Gradient Descent (Sieve-SGD)

Set $\alpha, \omega > 0$, step size $\{\gamma_i\}$ and basis functions $\{\psi_j\}$. Initialize $\bar{f}_0 = \hat{f}_0 = 0$.

For $i = 1, 2, \ldots$:

1. Calculate $J_i = \lfloor i^\alpha \rfloor$, collect data pair (X_i, Y_i).

2. Update $\hat{f}_i$:

$$\begin{aligned} \hat{f}_i &= \hat{f}_{i-1} + \gamma_i \left(Y_i - \hat{f}_{i-1}(X_i) \right) \sum_{j=1}^{J_i} j^{-2\omega} \psi_j(X_i) \psi_j \\ &= \hat{f}_{i-1} + \gamma_i \left(Y_i - \hat{f}_{i-1}(X_i) \right) K^{\omega}_{X_i, J_i} \end{aligned} \tag{3.26}$$

3. Polyak averaging: Update $\bar{f}_i$ by

$$\begin{aligned} \bar{f}_i &= \frac{1}{i+1} \sum_{k=0}^{i} \hat{f}_k \\ &\left(= \frac{i}{i+1} \bar{f}_{i-1} + \frac{1}{i+1} \hat{f}_i \right) \end{aligned} \tag{3.27}$$

We refer to the function $\bar{f}_i$ as the *Sieve-SGD estimate* of f_ρ. We will later show that $\bar{f}_i$ has rate-optimal MSE for estimating any $f_\rho \in W(s)$. Here we use the language of "updating a function", but in practice one would update the coefficient vector corresponding to the functions $\{\psi_j\}_{j=1}^{J_n}$. In Appendix 6.1 we attach a presentation of the algorithm that works directly with the coefficients. This estimator is quite simple, though it does require apriori selection/knowledge of $\{\psi_j\}$ and s (which can be done using a held-out validation set in practice). Unfortunately showing its favorable statistical properties (in Section 3.6) is somewhat more complex!

3.5.2 Computational expense

After examining the updating rule above, one can see that $\hat{f}_i$ has the form:

$$\hat{f}_i(x) = \sum_{j=1}^{J_i} b_j \psi_j(x) \tag{3.28}$$

This requires storing the coefficients $\{b_j\}_{j=1}^{J_i}$ in memory. The main computational burden of each update step is calculating $\hat{f}_{i-1}(X_i)$ and $K^{\omega}_{X_i,J_i}$. Both require evaluating J_i basis functions at X_i. Thus, the computational expense of the "Update $\hat{f}_i$" step above is of order $J_i = \Theta(i^{\alpha})$, when we take evaluating one basis function at one point as $O(1)$. And the total expense of processing n samples is of order $\Theta\left(n^{1+\alpha}\right)$. The space expense is of the same order $\Theta(i^{\alpha})$: We need only store coefficients of J_i basis functions. In Section 3.6.4 we will show that, under mild conditions, this memory complexity is near optimal among all estimators with rate-optimal MSE.

This compares favorably with standard kernel SGD (3.22) which uses i basis functions at step i: Our estimator uses fewer when $\alpha < 1$; as we will show later, α can be taken as small as $\frac{1}{2s+1}$ which is a substantial improvement. In practice, the parameter α can either be selected based on our assumptions about s (belief on the smoothness of f_ρ) or heuristically tuned for empirical performance.

3.5.3 General Convex loss

Although the main focus of this chapter is regression with squared-error loss, our algorithm has a straightforward extension to general convex loss. Suppose we want to minimize the population loss

$$E\left[\ell(Y, f(X))\right] \tag{3.29}$$

over all functions $f \in W(s, Q, \{\psi_j\})$ and the loss function $\ell(Y, \cdot)$ is convex for each Y. In this case, we need only modify step 2 of the Sieve-SGD estimator in Section 3.5.1. Given loss $\ell(\cdot, \cdot)$, the updating rule for $\hat{f}_i$ takes the general form:

2') Update $\hat{f}_i$:

$$\hat{f}_i = \hat{f}_{i-1} + \gamma_i \frac{\partial}{\partial v} \ell(u, v)\Big|_{\left(Y_i, \hat{f}_{i-1}(X_i)\right)} K^{\omega}_{X_i, J_i} \tag{3.30}$$

For example, with $Y = \{1, -1\}$ considering nonparametric logistic regression, the loss function one would use is $\ell(Y, f(X)) = \log(1 + \exp(-Yf(X)))$. In this case, we have

$$\frac{\partial}{\partial v} \ell(u, v)\Big|_{\left(Y_i, \hat{f}_{i-1}(X_i)\right)} = [1 + \exp\{Y_i \hat{f}_{i-1}(X_i)\}]^{-1} Y_i \in \mathbb{R} \tag{3.31}$$

Theoretical guarantees for Sieve-SGD using general convex loss are beyond the scope of this dissertation. However, in Section 3.7 we provide simulated experiments that show the empirical performance of Sieve-SGD for nonparametric logistic regression. These empirical results intimate that perhaps similar theoretical guarantees to those shown for squared-error-loss hold in a more general setting.

3.5.4 Choice of Basis Functions & Multivariate Problems

In practice, there are many available choices of univariate ψ_j that in general lead to interesting (Sobolev-type) hypothesis spaces. For example,

$$\psi_1(x) = 1, \quad \psi_j = \sqrt{2}\cos((j-1)\pi x), \text{ for } j \geq 2. \tag{3.32}$$

This set of basis functions are the "eigenfunctions" of Sobolev spaces over $[0, 1]$ (Appendix A.2 in [83]), which means they are orthogonal w.r.t to the Lebesgue inner product and the Sobolev inner product simultaneously. The corresponding Sobolev ellipsoid does not impose periodicity assumptions of f_ρ and is very convenient to use in practice. Among many other choices, we can also use algebraic polynomials, or a combination of algebraic polynomial and (periodic) Fourier basis [31].

In most applications, the covariate X_i's take value in $\mathbb{R}^p$ where $p > 1$. In some situations, there are some "canonical" choices of basis function $\psi(x) : \mathbb{R}^p \to \mathbb{R}$ that people might use for identifying their (multivariate) Sobolev ellipsoid. For example, when considering estimating

a function on a sphere $\mathbb{S}^2$, ψ_j could be taken as the orthonormal spherical harmonics ([60], [79]).

In many situations, the basis functions ψ_j can conveniently be taken as a tensor product of a one-dimensional complete basis, and Sieve-SGD can be directly applied in this multivariate setting. If we are using a univariate Sobolev ellipsoid to represent a ball in an RKHS, then the ellipsoid defined by the tensor product basis will correspond to a ball in the RKHS spanned by the tensor product kernel (though care needs to be taken with the ordering of the basis vectors). Some technical details and numerical examples on this can be found in Appendix 6.2 and the reference therein. In all of these cases, our theoretical results will hold (so long as the function f_ρ belongs to the specified space).

A common alternative approach in multivariate problems is to impose some additional structure on the hypothesis space to make estimation more tractable. This is particularly true when the feature dimension p is large. One popular model is the nonparametric additive model [112, 52, 141], which is thought to effectively balance model flexibility and interpretability. For $x \in \mathbb{R}^p$, we might consider assuming/imposing an additive structure on the regression function:

$$f_\rho(x) = \sum_{k=1}^{p} f_{\rho,k}\left(x^{(k)}\right) \tag{3.33}$$

where each of the component functions $f_{\rho,k}$ belong to a Sobolev ellipsoid $W_k(s_k, Q_k, \{\psi_{jk}\})$. For ease of exposition, in (3.33), we assume $E[Y] = 0$ to avoid the need for a common intercept term. For a more complete version with common intercept, see Appendix 6.2. For a fixed dimension p, when all $W_k = W^*$ (for some Sobolev ellipsoid W^*), the minimax rate for estimating such an additive model is identical (up to a multiplicative constant p) to the minimax rate in the analogous one-dimension nonparametric regression problem with the same hypothesis space W^* [92, 112]. For the additive model (3.33), the updating rule (3.26) of Sieve-SGD could be replaced by:

$$\hat{f}_i = \hat{f}_{i-1} + \gamma_i \left(Y_i - \sum_{k=1}^{p} \hat{f}_{i-1,k}\left(X_i^{(k)}\right) \right) \sum_{k=1}^{p} \sum_{j=1}^{J_{ik}} j^{-2\omega_k} \psi_{jk}\left(X_i^{(k)}\right) \psi_{jk} \tag{3.34}$$

here J_{ik} is the truncation level of k-th dimension when the sample size $= i$ and $\hat{f}_{i-1,k}$ is the estimate of $f_{\rho,k}$. Most of the theory that we develop in Section 3.6 could apply here.

3.6 Generalization Guarantees of Sieve-SGD

In this section, we show Sieve-SGD achieves the minimax rate for nonparametric estimation in Sobolev ellipsoids under mild assumptions.

We also show that Sieve-SGD has near minimal memory complexity among all estimators that are minimax rate-optimal for estimation in a Sobolev ellipsoid. The conditions on the hyperparameters can be used as theoretical guidance when applying Sieve-SGD to real data problems.

3.6.1 Model Assumptions

We begin by listing the conditions we will require in our proof. They reflect different aspects of the problem: independent observations (A1), distribution of feature X (A2), the hypothesis space assumed for f_ρ (A3) and tail behaviour of the noise (A4). These conditions ensure the MSE rate-optimality of Sieve-SGD.

A1 (i.i.d. data) The data points $(X_n, Y_n)_{n\in\mathbb{N}} \in \mathcal{X} \times \mathbb{R}$ are independently, identically sampled from a distribution $\rho(X, Y)$.

A2 (feature distribution) Let ν be a user-specified measure that is strictly positive on $\mathcal{X}$. Assume the distribution of feature X, ρ_X, is absolutely continuous w.r.t. ν. Let $p_X = d\rho_X/d\nu$ denote its Radon–Nikodym derivative. We assume for some u, ℓ such that $0 < \ell < u < \infty$:

$$\ell \leq p_X(x) \leq u \quad \text{for all } x \in \mathcal{X}$$

A3 (Sobolev ellipsoid) Let $\{\psi_j\}_{j=1}^{\infty}$ be a set of uniformly bounded ($\|\psi_j\|_\infty \leq M$), continuous, orthonormal basis of L^2_ν. We assume the regression function f_ρ falls in a Sobolev

ellipsoid, with basis functions given by $\{\psi_j\}$, i.e. for some $s > \frac{1}{2}, Q < \infty$,

$$f_\rho \in W(s, Q, \{\psi_j\}) \tag{3.35}$$

A4 (noise) One of the following two assumptions is satisfied by the noise variable $\epsilon = Y - f_\rho(X)$:

- ϵ is bounded by some C_ϵ almost surely.
- ϵ is independent of the features, X, and has a finite second moment $E[\epsilon^2] = C_\epsilon^2$.

Note 1: The lower bound requirement of p_X in A2 may be due to artifacts in our proof. In reality, especially when the dimension of our feature-space X is large, such an requirement may be hard to satisfy. According to our simulation results, Sieve-SGD still achieves the minimax rate even when ρ_X has a strictly smaller support than ν. As compared with other work in nonparametric online learning [25, 116, 137], our assumptions are more direct. We discuss this in detail later in this section.

Note 2: In assumption A3, we do not require ψ_j to be orthonormal w.r.t. ρ_X (and it is in general not true), but only require them to be orthonormal w.r.t. the known measure ν. In many cases ν might be taken to be Lesbesgue (or uniform) measure over a domain containing $\mathcal{X}$, as this is the canonical measure under which function spaces such as Sobolev spaces and Besov spaces are defined. As long as the density function p_X satisfies A2, using the non-orthonormal (w.r.t. ρ_X) basis functions ψ_j, does not prevent Sieve-SGD from having rate-optimal MSE.

Note 3: It is a common convention to think about a hypothesis space where the Sobolev(-type) norm of the regression function is bounded by a constant Q (A3), rather than just $< \infty$. Such a bounded space has a finite minimax rate: the exponent is determined by s, and the constant is proportional to Q (see also note 5 under Theorem 3.6.1). We also would like to note that the proposed algorithm does not use radius Q at any point and the theoretical guarantee is adaptive w.r.t. Q. (More specifically, the final bounds given in Appendix D.3

and Lemma D.1 have $\|f_\rho\|_K$, which can be thought of as the effective value for Q, on the right-hand-side.)

3.6.2 Rate optimality when $t_j = j^{-2s}$

In this section, we present the rate-optimality results of Sieve-SGD when we choosing the component-specific learning rate to be $t_j = j^{-2s}$ (or $\omega = s$ in (3.26)). In this setting, our theoretical analysis treats Sieve-SGD as a truncated-version (in the basis expansion domain) of a "correct" kernel SGD procedure (we will discuss the "incorrect" version very soon in Section 3.6.3, and show that it can actually still have favorable statistical and computational properties). Here is the main result in this setting:

Theorem 3.6.1. *Assume A1-A4. Set the component-specific learning rate as $t_j = j^{-2s}$. Also set the overall learning rate to be $\gamma_n = \gamma_0 n^{-\frac{1}{2s+1}}$ with $\gamma_0 \leq (2M^2\zeta(2s))^{-1}$, where $\zeta(k) = \sum_{i=1}^{\infty} i^{-k}$. Choose the number of basis functions to be $J_n \geq n^\alpha \log^2 n \vee 1$ for an arbitrary $\alpha \geq \frac{1}{2s+1}$.*

Then the MSE of Sieve-SGD (3.27) *converges at the following rate*

$$E\|\bar{f}_n - f_\rho\|^2_{L^2_{\rho_X}} = O\left(n^{-\frac{2s}{2s+1}}\right). \tag{3.36}$$

This implies that Sieve-SGD is a minimax rate-optimal estimator of f_ρ over $W(s, Q, \{\psi_j\})$.

We now discuss our assumptions and results in more detail, and relate them to what is currently in the literature.

Note 1: In the analysis of many reproducing kernel methods for nonparametric estimation [25, 116, 140], the spectrum of the *covariance operator* plays an important role in controlling the statistical behavior of estimators. It is conventional in the community to make assumptions associated with this spectrum, which we find less natural than our related assumptions A2 and A3. The covariance operator is an analog of the covariance matrix in infinite dimensional spaces. For our problem setting, one natural covariance operator T_X is

defined as:

$$\begin{aligned} T_X : L^2_{\rho_X} &\to L^2_{\rho_X} \\ g &\mapsto \int_{\mathcal{X}} g(\tau) \left(\sum_{j=1}^{\infty} j^{-2s} \psi_j(\tau) \psi_j \right) d\rho_X(\tau). \end{aligned} \tag{3.37}$$

A direct analysis of the spectrum of T_X is hard. However, there is a simpler operator that we have in hand which we can relate T_X to:

$$\begin{aligned} T_\nu : L^2_{\nu} &\to L^2_{\nu} \\ g &\mapsto \int_{\mathcal{X}} g(\tau) \left(\sum_{j=1}^{\infty} j^{-2s} \psi_j(\tau) \psi_j \right) d\nu(\tau). \end{aligned} \tag{3.38}$$

We know the eigensystem of T_ν: It is exactly (j^{-2s}, ψ_j) (eigenvalue, eigenfunction). It is direct to check because $\{\psi_j\}$'s are orthonormal w.r.t. ν, so $\int \psi_j(\tau) \sum_{j=1}^{\infty} j^{-2s} \psi_j(\tau) \psi_j d\nu(\tau) = j^{-2s} \psi_j$. As an additional contribution, our work shows that with the simple assumptions A2 & A3, we can get knowledge about T_X's eigenvalues from those of T_ν.

Lemma 3.6.2. *Given assumptions A2, A3, the j-th eigenvalue, λ_j, (sorted in a decreasing order) of T_X satisfies $\lambda_j = \Theta(j^{-2s})$.*

Moreover, the upper bound of the density in A2 ensures the upper bound in Lemma 3.6.2 ($\lambda_j = O(j^{-2s})$), and the lower bound of the density ensures the other half of the result. The proof of the above Lemma uses the underlying connection between the eigenvalues of an operator and its metric entropy. For rigorous definitions and proof of Lemma 3.6.2, see Appendix 6.3.

Although the exact result of Lemma 3.6.2 is not used in the proof of Theorem 3.6.1 (or Theorem 3.6.3). We still present it here since it may be of interest itself and the stated results is less technical and easier to comprehend. The proof of the more technical version (Lemma 6.3.14) follows very closely to that of Lemma 3.6.2. In that more general version, we investigate the spectrum of covariance operators of form: $T^{\omega}_{X,J_n}(f) = \int f(\tau) \left(\sum_{j=1}^{J_n} j^{-2\omega} \psi_j(x) \psi_j \right) d\rho_X(\tau)$.

To prove Theorem 3.6.1, we need to engage with a series of RKHSs with kernels given by

$$\begin{aligned} K^s_{J_n} : \mathcal{X} \times \mathcal{X} &\to \mathbb{R} \\ (s,t) &\mapsto \sum_{j=1}^{J_n} j^{-2s} \psi_j(s)\psi_j(t) \end{aligned} \tag{3.39}$$

While we discuss our work in the context of Sobolev ellipsoids, there is an equivalent formulation directly in RKHS. See Appendix 6.3 for more discussion. Although an explicit form for $K^s_{J_n}$ is not in general necessary or accessible for Sieve-SGD, the existence (i.e. the absolute convergence of the infinite sum) of $K^s_{J_n}$ is a direct consequence of A3. This is enough for theoretical analysis. For kernel SGD methods, a fixed kernel (with $J_n = \infty$) is used and there is only one relevant RKHS. This means, on average, kernel SGD is applying the same procedure each iteration; but for Sieve-SGD, we need to engage with a series of increasing RKHSs (on average, Sieve-SGD may not be doing the same thing between iterations). As a side contribution, we present how to handle such a more technically involved case.

Note 2: In contrast to our assumption A3, the hypothesis spaces in [25, 116, 137, 77] are described in terms of "T_X" and its eigen-decomposition. This unfortunately obfuscates difficulties related to verifying those conditions when analyzing their statistical performance (though applying the learning algorithm in practice does not need the knowledge of the eigensystem). In particular because ρ_X is involved in the definition of T_X (3.37), we need knowledge of (generally unknown) ρ_X to characterize T_X, and understand its eigenvalues and eigenfunctions.

More specifically, in the literature we mentioned above, it is often assumed that for some $r \in [1/2, 1]$ (Definition 6.3.6):

$$\|T_X^{-r}(f_\rho)\|^2_{L^2_{\rho_X}} < \infty \tag{3.40}$$

This can be related to a Sobolev ellipsoid-type condition

$$\|T_X^{-r}(f_\rho)\|^2_{L^2_{\rho_X}} = \sum_{j=1}^{\infty} \lambda_j^{-2r}\theta_j^2 < \infty \quad \text{where } f_\rho = \sum_{j=1}^{\infty} \theta_j \phi_j \tag{3.41}$$

where $(\lambda_j, \phi_j)_{j=1}^{\infty}$ are the eigenvalue and eigenfunctions of operator T_X, and ϕ_j's are orthonormal w.r.t. $L^2_{\rho_X}$. Unfortunately, we cannot directly engage with $(\lambda_j, \phi_j)_{j=1}^{\infty}$, since calculating

them requires knowledge of ρ_X. Thus, assumptions formulated in the language of T_X^{-r} are difficult to directly understand. In contrast, our assumptions translate to analyzing the spectrum of T_ν, which has no dependence on ρ_X, and its spectrum can been directly calculated (as noted above).

Note 3: For parametric SGD methods, usually a bound on the second moment of the gradient vector is required to guarantee rate-optimal performance (both theoretically and in practice). Formally, for optimization problem (3.10), it is usually required that $E[\|\nabla \ell(\beta)\|^2] \leq R^2 < \infty$ for all $\beta \in \mathbb{R}^d$ [11, 27].

For nonparametric stochastic approximation, there is a similar regularity requirement for the "gradient". The assumptions A2-A3 are enough to ensure this for Sieve-SGD. In our proof, we show that there exists a number $R < \infty$ such that for all $x \in \mathcal{X}$ and any J_n, we have $\|K_{x,J_n}^s\|_K^2 \leq R^2$. This result is listed in Lemma 6.4.1 where $R^2 = M^2\zeta(2s)$ and $\zeta(k) = \sum_{i=1}^{\infty} i^{-k}$. In Theorem 3.6.1, we required γ_0 to be smaller than $(2M^2\zeta(2s))^{-1}$ to ensure our theoretical guarantees.

Note 4: For completeness, here we state the minimax-rate of our nonparametric regression problem over a Sobolev ellipsoid:

$$\liminf_{n\to\infty} \inf_{\hat{f}} \sup_{f_\rho \in W(s,Q,\{\psi_j\})} E\left[n^{\frac{2s}{2s+1}} \|\hat{f} - f_\rho\|_{L^2_{\rho_X}}^2 \right] \geq C \tag{3.42}$$

where the infimum ranges over all possible functions $\hat{f}$ that are sufficiently measurable. For a derivation of this lower bound, see [128, Chapter 15]. Many other online methods we mentioned in Section 3.4 can achieve this lower bound, however, their computational expense is in general unfavorable compared with the proposed method.

Also, the bound (3.36) should not be understood as a dimension-free result. When the feature $X \in \mathbb{R}^p$ is a multivariate vector, the parameter s should be treated as $s = s^*/p$, where s^* is, for example, the order of derivative that we assume the regression function f_ρ has. Plugging this into the result presented in Theorem 3.6.1 gives a dimension-dependent bound of order $n^{-\frac{2s^*}{2s^*+p}}$ in which both the smoothness assumption and dimension show up in the exponent. Such a bound is minimax optimal when learning in a (large) homoge-

neous multivariate space [110]. In practice, one can usually achieve better performance by leveraging other low dimensional structure (See Section 3.5.4 and Appendix 6.2).

3.6.3 Robustness of t_j for Properly Chosen J_n

In Section 3.6.2 we presented the optimality guarantees of Sieve-SGD in the case when the component-specific learning rate is chosen as the most "natural" value, i.e. $t_j = j^{-2s}$. In that case, Sieve-SGD is statistically optimal so long as the number of basis functions does not increase too slowly, that is, $J_n \geq n^{\frac{1}{2s+1}} \log^2(n)$. Specifically, when $J_n = \infty$, the Sieve-SGD updating rule reduces to the kernel SGD updating rule (3.20) with kernel $K^s_\infty(X_n, \cdot) = \sum_{j=1}^{\infty} j^{-2s} \psi_j(X_n)\psi_j(\cdot)$. So long as we have access to the closed-form of $K^s_\infty(X_n, \cdot)$, the corresponding kernel SGD estimator is also statistically optimal under the same conditions. In such a scenario, Sieve-SGD can be seen as a truncated-version of a "correct" kernel SGD method with much better computational properties.

Although truncating the kernel in the spectral domain may be seen as an extension of kernel SGD, it can alternatively be seen as related to projection estimators: In that case however, two pieces of Theorem 3.6.1 may seem unnatural: 1) the strict requirement on t_j $(= j^{-2s})$; and 2) The fact that we only lower bound the truncation rate, rather than requiring a precise value for the growth of J_n. In the case of the original Theorem 3.6.1, the bias-variance tradeoff is actually not balanced via truncation. Instead, it is balanced directly using the t_j. The required lower bound on the truncation rate is just given to ensure that we do not accrue excess bias. Alternatively, to better parallel projection estimators, it might seem more natural to directly use the number of basis functions J_n to control the bias-variance tradeoff (there is nothing akin to t_j in (3.8)). In this section we will explore this idea: That if we are more precise in specifying J_n, perhaps we can be more flexible with t_j.

More specifically, we are interested in milder conditions on the sequence (t_j) such that if we properly select the rate at which our "dimension" increases (i.e. the rate at which J_n grows), Sieve-SGD would still attain its favorable statistical and computational properties.

Since we will be using J_n as the tuning parameter to balance bias and variance, one might expect kernel SGD, which sets $J_n = \infty$, would not always have optimal statistical performance for all sequences (t_j) satisfying the "milder" conditions. This is confirmed via the following theorem: For Sieve-SGD one can actually use large component-wise step-sizes that need only satisfy $t_j < j^{-1}$ for any smoothness class $W(s)$, so long as the truncation level is appropriately set; while the largest t_j that can be used for kernel-SGD (without truncation) needs to ensure $t_j < j^{-(s+1/2)}$, which depends on the smoothness of f_ρ.

Theorem 3.6.3. *Assume A1-A4. Set the component-specific learning rate to be* $t_j = j^{-2\omega}$ *with* $\frac{1}{2} < \omega \leq s$. *Choose the learning rate to be* $\gamma_n = \gamma_0 n^{-\frac{1}{2s+1}}$, *with* $\gamma_0 \leq M^2\zeta(2\omega)/2$. *Choose the number of basis functions to be* $J_n = n^{\frac{1}{2s+1}} \log^2 n \vee 1$.

Then the MSE of Sieve-SGD (3.27) *converges at the following near optimal rate*

$$E\|\bar{f}_n - f_\rho\|^2_{L^2_{\rho_X}} = O\left(n^{-\frac{2s}{2s+1}} \log^2 n\right) \tag{3.43}$$

Note 1: The requirement of $t_j < j^{-1}$ is to guarantee a finite "second moment" of the gradient (as in Note 3 under Theorem 3.6.1). In this theorem, once this minimal requirement is satisfied, the decay rate of t_j does not influence either the statistical guarantees, nor the computational expense of the estimators — both of these are determined entirely by the truncation level. As we will discuss very soon in Section 3.6.4, the choice of $J_n = n^{\frac{1}{2s+1}} \log^2 n$ in Theorem 3.6.3 and Theorem 3.6.1 would result in algorithms that are both statistically and computationally near-optimal up to a polylog term.

Note 2: The most direct form of the projection estimator determines the basis functions' coefficients by solving a (unpenalized) least square problem (3.8) in which there are no learning rates involved. It is the truncation level J_n that determines the bias-variance trade-off and statistical performance. In Theorem 3.6.3 we present a stochastic approximation analog to that result. From a reproducing-kernel methodology perspective, Theorem 3.6.1 investigates the cases when the capacity of the kernel (ω) matches the source smoothness (s); in Theorem 3.6.3 we show under what conditions the mismatch between these two quantities does not affect the statistical (and computational) properties of Sieve-SGD. It is very common

to discuss the generalization ability of a reproducing kernel method in the literature when the kernel capacity does not match the source smoothness. For example, in [25] the authors use a pair of parameters (r, α) to state the hypothesis space and the capacity of the kernels. The smoothness of the hypothesis space is determined by the product of the two parameters $r\alpha$. When $r \neq 1/2$, they are considering using a kernel whose capacity does not match the smoothness of f_ρ. Their proposed method must modify the learning rate properly to recover rate-optimality (or it is impossible due to saturation).

Comparing their results with Theorem 3.6.3, there are ω such that the kernel SGD estimator, using kernel $K_\infty^\omega(X_n, \cdot) = \sum_{j=1}^\infty j^{-2\omega}\psi_j(X_n)\psi_j(\cdot)$, may not be optimal no matter how we modify the learning rate γ_n (described as "saturation" in [25]). Whereas for Sieve-SGD using the truncated "kernels" $K_{J_n}^\omega(X_n, \cdot) = \sum_{j=1}^{J_n} j^{-2\omega}\psi_j(X_n)\psi_j(\cdot)$, the statistical and computation performance can still be jointly near optimal. Theorem 6.3 is formally similar to such a "source-capacity" discussion, but the results are quite different in nature — in particular it is the truncation level that "saves" us, and allows a much larger mismatch between kernel capacity and source smoothness.

Note 3: The overall proof structures of Theorem 3.6.1 and Theorem 3.6.3 are similar; the difference is, in the proof of Theorem 3.6.1 we need Lemma 6.4.4 and related technical results, but for Theorem 3.6.3 we use Lemma 6.5.1 instead.

Note 4: We also provide some intuition for using a decreasing learning rate γ_n: For rate-optimal *parametric* SGD methods with averaging, the learning rate γ_n can be taken as a constant γ_0. However, the employed constant γ_0 is inversely proportional to the dimension of parameter (assuming each dimension of the feature has a bounded support) [6], which is, in some sense, consistent with our results (though we have seen no other results in the literature that engage with a dimension that increases as the learning process proceeds). We require the learning rate to be a decreasing sequence so that it can "cancel out" the effect of increasing estimator-dimension: The increasing dimension would have resulted in a noise variance that is increasing if care was not taken.

3.6.4 Near optimal space expense

In this section we will show that Sieve-SGD is asymptotically (nearly) space-optimal for estimating f_ρ in a Sobolev ellipsoid under the conditions listed in Section 3.6.1. We will show that, even with computer round-off error, Sieve-SGD only needs $\Theta(n^{\frac{1}{2s+1}}\log^3 n)$ *bits* to achieve the minimax rate for MSE (or off by a $\log^2(n)$ term when $\omega \neq s$ as stated in Theorem 3.6.3), and further, that there is no estimator with $o(n^{\frac{1}{2s+1}})$ bits of space expense that can achieve the minimax-rate for estimating $f_\rho \in W(s, Q)$. In our analysis we note that computers cannot store decimals in infinite precision, and formally deal with a modified version of our algorithm that stores coefficients in fixed precision (that grows in n): This necessitates the extra $\log(n)$ term (compared with the number of basis function needed in Theorem 3.6.1 and 3.6.3). The modified algorithm with fixed, but growing precision still results in the same MSE when round-off error is not considered.

We first give a more formal definition of the space expense of an estimator in our analysis. A regression estimator can be seen as a mapping M_n from the data $Z_1^n = \{(X_i, Y_i) \mid i = 1, 2, ..., n\}$ to a function $\hat{f}_n \in \mathcal{F}$. For any such M_n that can be engaged by a computer, must be decomposable into an "encoder-decoder" pair (E_n, D_n). Here E_n represents the "encoder" that compresses the information into computer memory. Formally, we define E_n to be a mapping from Z_1^n to a binary sequence of length b_n. And the corresponding D_n is the "decoder" of the binary sequence that translates the information saved in memory back to a mathematical object $\hat{f}_n$. Generally, the binary sequence length b_n will increase with n: As more information is contained in the data, we need more memory to store an increasingly accurate estimate of our regression function.

Given an estimator that can decomposed into a pair (E_n, D_n), one can see that the decomposition is not unique. There are, in fact, infinitely many pairs that are trivially different from each other for any such estimator. Moreover, E_n, D_n's can be random mappings if we allow random algorithms: For example, random forests include additional randomness due to bootstrapping/variable selection. In order to be more precise regarding memory

complexity constraints, we introduce the following formalization.

Definition 3.6.4 (b_n-sized estimator)**.** *Given a sequence of integers $(b_i)_{i\in\mathbb{N}^+}$, we say an estimator $M_n : (\mathcal{X}\times\mathbb{R})^n \to \mathcal{F}$ is a b_n-sized estimator if it satisfies the following conditions:*

1. *For every n, there exists an encoder mapping $E_n : (\mathcal{X}\times\mathbb{R})^n \to \{0,1\}^{b_n}$, and a decoder mapping $D_n : \{0,1\}^{b_n} \to \mathcal{F}$ such that*

$$M_n = D_n \circ E_n \tag{3.44}$$

2. *The decoder D_n is a known, fixed mapping. E_n can be either a random or fixed mapping.*

We use the sample mean as a toy example to illustrate the above definition. In practice, the sample mean is usually a 64-sized estimator of the population mean. Here 64 stands for the number of bits needed to represent a double-precision floating point number. In this case the size $b_n = 64$ does not increase with sample size n. However not every real number can be arbitrarily precisely specified by a fixed-length floating-point number, so a careful asymptotic analysis of estimation of the mean suggests that perhaps we should store a sample mean with growing levels of precision, i.e. b_n would need to grow with n. A binary sequence of length s can specify 2^s real numbers, so to achieve the $O(n^{-1})$ statistically optimal bound for mean estimation, a $\log(n)$-sized version of sample mean is formally required. In practice, 64-bit precision is generally more than enough for mean estimation. Nevertheless, in this chapter we aim to give a more formal and precise asymptotic analysis of our Sieve-SGD estimator.

Readers who are more familiar with computational complexity theory may find our definition similar to a (probabilistic) Turing machine. However, in our framework the machine does not use binary sequences on tapes as input and output; nor do we need to identify the basic operations on the "machine". We aimed to remove unnecessary complexity for readers with a more statistical background. Discussion of Turing machines using finite length working tape can be found in [4, Chapter 4].

To construct Sieve-SGD estimators that achieve (near) optimal MSE, we only need to store the coefficients of the $J_n = \Theta(n^{\frac{1}{2s+1}}\log^2 n)$ basis functions. However, as in our example

with the sample mean, we need to be careful about the precision with which we store those coefficients. We need to determine: i) how small we require the round-off error to be in order to maintain the statistical optimality of Sieve-SGD, and ii) how much space expense is required to achieve such precision. In Appendix 6.6.1 we identify how round-off error is introduced into the system and how it decreases as more bits are used to store each coefficient. In Corollary 6.6.2 we show that by using $\Theta(\log n)$ bits per coefficient, a $O(n^{\frac{1}{2s+1}}\log^3 n)$-sized version of Sieve-SGD can achieve the same optimal convergence rate as in the infinite precision setting (or equivalently round-off-error-free setting).

Combining the above result with the following theorem, we can conclude that no MSE rate-optimal estimator can require less memory by a polynomial factor than Sieve-SGD.

Theorem 3.6.5. *Let b_n be a sequence of integers, and $b_n = o\left(n^{\frac{1}{2s+1}}\right)$. Let $\mathcal{M}(b_n)$ be the collection of all b_n-sized estimators, then we have*

$$\lim_{n\to\infty}\inf_{M_n\in\mathcal{M}(b_n)}\sup_{f_\rho\in W(s,Q,\{\psi_j\})} E\left[n^{\frac{2s}{2s+1}}\|M_n(Z_1^n)-f_\rho\|^2_{L^2_{\rho_X}}\right]=\infty \tag{3.45}$$

i.e. no such b_n-sized estimators can be rate-optimal.

This theorem tells us we cannot find any minimax rate-optimal $o(n^{\frac{1}{2s+1}})$-sized estimator. Thus the best rate-optimal estimator one can expect to find is a $\Theta(n^{\frac{1}{2s+1}})$-sized estimator: Sieve-SGD's space expense only misses this lower bound by a poly-logarithmic factor.

We give the proof of the above theorem in Appendix 6.6.2. Although here we focus on regression in Sobolev spaces, the technique used can be applied to other hypothesis spaces. The proof is based on the concept that metric-entropy is the minimal number of bits needed to represent an arbitrary function from a function space up to ϵ-error, which can be traced back to [62]. Also, following a very similar argument, one can prove that no constant-sized estimator can be rate-optimal (or even consistent) for parametric regression problems. We discuss this further in the Appendix 6.6.2. We also include some discussion of the time expense in Section 3.8.

3.7 Simulation study

3.7.1 Sieve-SGD for online regression

In this section, we illustrate both the statistical and computational properties of Sieve-SGD with simulated examples. The two examples we use have different f_ρ, $W(s, Q, \{\psi_j\})$ and ρ_X. The user-specified measure ν is taken as the uniform distribution over $[0, 1]$ in both. We provide the details of our simulation settings in Table 3.1. These two examples are designed for verifying our theoretical guarantees: The f_ρ we use have known explicit series expansion or is constructed explicitly using the basis function ψ_j (to ensure the truth is hard enough to learn in the assumed Sobolev ellipsoid). In Appendix 6.2 we provide more numerical examples to better mimic the practical application: we engage with multivariate features and compare Sieve-SGD with many popular machine learning methods.

Example 1 In this example, we examine the empirical performance of Sieve-SGD and compare it with two other methods in batch or online nonparametric regression: kernel ridge regression (KRR) [128] and kernel SGD [25]. We will see that the relationship between generalization error $E\|\bar{f}_n - f_\rho\|_2^2$ and sample size corresponds well with our theoretical expectations presented in Theorem 3.6.1 (Fig 3.1).

The true regression function we chose for Example 1 is also used in the analysis of kernel SGD [25]. In that paper, kernel SGD with Polyak averaging is compared with other (kernel-based) nonparametric online estimators [116, 137], and has been shown to have similar or better performance, so we include only kernel SGD with averaging as the reference online-estimator. We also note that although KRR performs slightly better than online methods, its time expense (which is of order $\Theta(n^3)$ per update) is dramatically more than online-estimators (kernel SGD $\Theta(n)$, Sieve-SGD $\Theta(J_n)$, per update).

We compare the empirical performance of Sieve-SGD under two different distributions of X. In Fig 3.1 panel (A), X has an uniform distribution over $[0, 1]$ and in panel (B) it has a distribution with a strictly smaller support (uniform over $[0.25, 0.75]$). The trigonometric basis functions we use are orthonormal w.r.t. ν, the Lebesgue measure over $[0, 1]$ (panel (A))

but not w.r.t. the one in panel (B). Although only the feature distribution in panel (A) satisfies the distribution assumption in A2, in both cases Sieve-SGD achieves the optimal-rate. This is a heuristic evidence indicating the lower bound requirement in A2 may be due to some artifacts in the proof.

Table 3.1: Settings of simulation studies. $B_4(x) = x^4 - 2x^3 + x^2 - \frac{1}{30}$ is the 4-th Bernoulli polynomial. $\{x\}$ indicates the fractional part of x.

	Example 1	Example 2
True f_ρ	$B_4(x)$	$4\sqrt{2}\sum_{j=1}^{50}(-1)^{j+1}j^{-4}\sin((2j-1)\pi x/2)$
ellipsoid para. s	2	3
J_n	$n^{0.21}$	$n^{0.10}$ & $n^{0.15}$ & $n^{0.43}$
t_j	$j^{-1.02}$ & j^{-4}	j^{-6}
$\psi_j(x)$	$\sin(2\pi\lceil j/2\rceil x)$, j is even $\cos(2\pi\lceil j/2\rceil x)$, j is odd	$\sqrt{2}\sin(\frac{(2j-1)\pi x}{2})$
Kernel $K(s,t)$	$-\frac{1}{24}B_4(\{s-t\})$	$\min(s,t)$
Noise	Unif$[-0.02, 0.02]$ or Unif$[-0.2, 0.2]$	Normal(0,1)
γ_0	3	1

Example 2 In this example, we consider the performance of Sieve-SGD under different $J_n = \lfloor n^\alpha \rfloor$ (number of basis functions). The f_ρ we use is explicitly constructed with basis functions $\psi_j(x) = \sqrt{2}\sin((2j-1)\pi x/2)$ and we tune the proposed method based on the (correct) assumption that it belongs to Sobolev ellipsoid $W(3, Q, \{\psi_j\})$ (see Theorem 4.1 of [54, Chapter 1] for completeness and orthonormality of $\{\psi_j\}$).

According to Theorem 3.6.1, in order to guarantee statistical optimality, we need $\alpha \geq \frac{1}{(2s+1)} \sim 0.14$. We consider several values of α, one below the this threshold, and two above

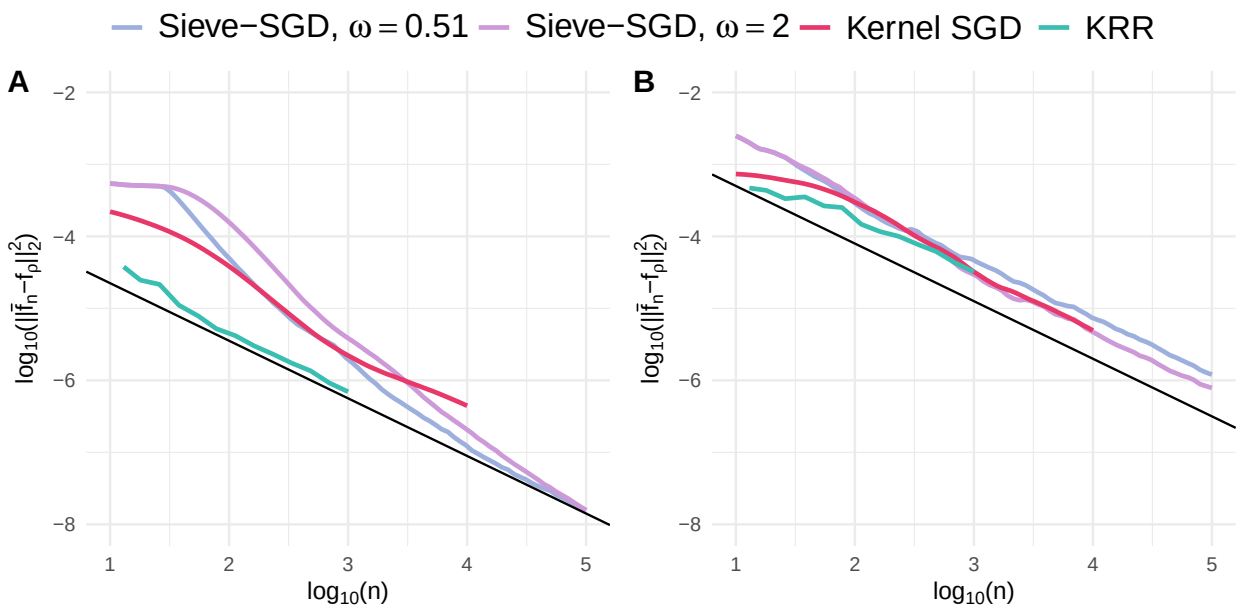

Figure 3.1: Example 1, $\log_{10} \|\bar{f}_n - f_\rho\|_2^2$ against $\log_{10} n$. The Black line has slope $= -4/5$, which represents the optimal-rate. Each curve is calculated as the average of 100 repetitions. (**A**) X is uniformly distributed over $[0, 1]$. In this setting, SNR ~ 3. (**B**) X has a distribution in which ψ_j are not orthonormal. We present the results with very large noise, SNR~ 0.03. Due to different computational costs, we chose different maximum n for different methods.

it:

$$\mathbf{0.10} < \frac{1}{2s+1} \sim 0.14 < \mathbf{0.15} < \mathbf{0.43} \tag{3.46}$$

As we can see from Fig 3.2 (A), when $\alpha = 0.15$ & 0.43, Sieve-SGD is rate-optimal as expected. When $\alpha = 0.10$, we are using too few basis functions, which results in the sub-optimal statistical performance. Such a suboptimality is because of the bias term: there are too few basis functions used. In fact, the parameter setting $\alpha = 0.1$ is so small that there are only 3 basis functions used when $n = 10^5$. To verify the above statement, we can briefly calculate when the second and the third basis functions are added in: $(10^3)^{0.1} \sim 2$, this corresponds to the first acceleration of the learning rate around $\log_{10}(n) = 3$; similarly, $(10^{4.8})^{0.1} \sim 3$, which explains the second one.

In Fig 3.2 (B), we show the CPU time for reference. For Sieve-SGD, the *accumulated* CPU time should be on the order of $\Theta(n^{1+\alpha})$: The larger α, the more basis functions required, the slower the algorithm. We also include the CPU time of kernel SGD with averaging as a benchmark, which has a cumulative computational expense of order $\Theta(n^2)$. The code is written in R (4.0.4), and runs on (the CPU of) a machine with 1 Intel Core m3 processor, 1.2 GHz, with 8 GB of RAM.

3.7.2 Sieve-SGD for Alternative Convex Losses

In this section, we provide the results of an experiment applying Sieve-SGD to online non-parametric logistic regression. Although this chapter gives no theoretical guarantees in this setting, it is still of interest to see the empirical performance of Sieve-SGD for general convex loss. Here, the distribution of class labels Y was generated by $Y \sim 2\,\mathrm{Ber}(g(X)) - 1$, where $(g(x))^{-1} = 1+\exp\{-5(1-2|x-0.5|)\}$; and the distribution of X was uniform over $[0, 1]$. Thus, the minimizer f^* of loss $E[\ell(Y, f(X))] = E[\log(1 + \exp(-Y f(X)))]$ is $f^* = 5(1 - 2|x - 0.5|)$.

When we apply the Sieve-SGD estimator (3.30) to this problem, we assume

$$f^* \in W\left(1, Q, \left\{\sqrt{2}\sin\left((2j-1)\pi x/2\right)\right\}\right) \tag{3.47}$$

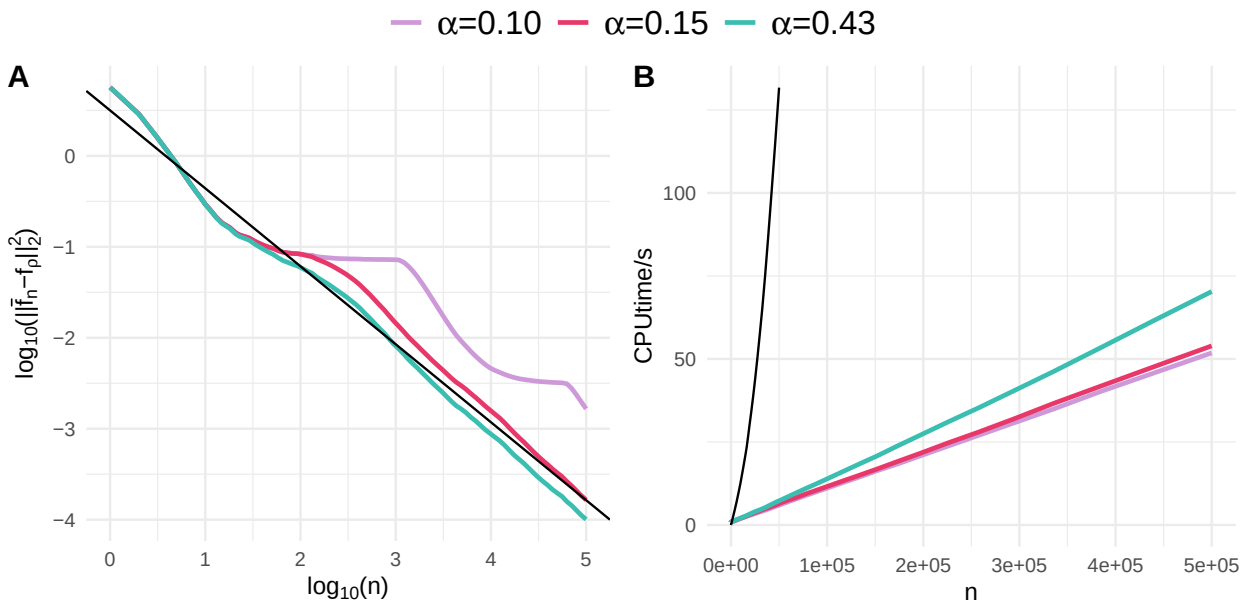

Figure 3.2: Example 2, effect of truncation exponents $\alpha = 0.10, 0.15, 0.43$. (A) Statistical performance, $\log_{10} \|\bar{f}_n - f_\rho\|_2^2$ against $\log_{10} n$. The black line has slope $= -6/7$, which represents the optimal-rate. Each curve is calculated as the average of 100 repetitions. (B) The accumulated CPU time to process n observations. The black line is the CPU time of kernel SGD, included for benchmark.

We try several $\alpha = 0.10, 0.33, 0.50$, all with $\gamma_0 = 6$. As we can see from Fig 3.3, the regret $E[\ell(\bar{f}_n) - \ell(f^*)]$ converges to zero at an apparent rate of $n^{-2/3}$ when $\alpha = 0.33, 0.50$ (which would agree with our result for squared error loss). When the number of basis functions increases too slowly (here is $\alpha = 0.10$), the regret decreases slowly after ~ 10 observations (for similar reason of overflowing bias term as we noted in section 3.7.1).

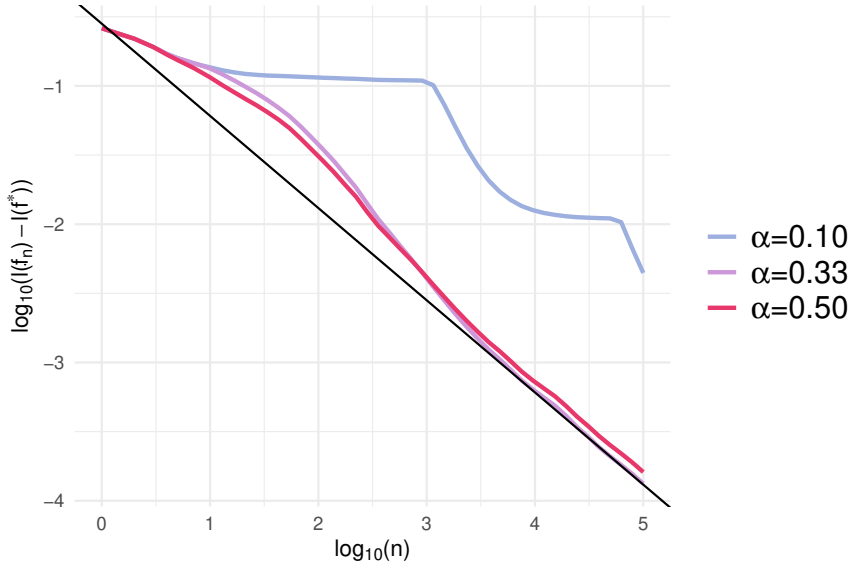

Figure 3.3: Example 3, empirical performance of Sieve-SGD in nonparametric logistic regression problem. Plot $\log_{10}(l(\bar{f}_n) - l(f^*))$ against $\log_{10} n$. The Black line has slope $= -2/3$. Each curve is calculated as the average of 100 repetitions.

3.8 Discussion

In this chapter, we considered online nonparametric regression in a Sobolev ellipsoid. We proposed the *Sieve Stochastic Gradient Descent estimator (Sieve-SGD)*, an online estimator inspired by both a) the nonparametric projection estimator, which is a special realization of general sieve estimators; and b) estimators constructed using stochastic gradient descent algorithms. By using an increasing number of basis functions, Sieve-SGD has rate-optimal

estimation error and is computationally very efficient.

For online learning problems with general convex losses, the optimal estimation rate depends on both the hypothesis space and loss function (e.g. whether it is Lipschitz or strongly convex). In this chapter we did not establish theoretical guarantees for Sieve-SGD when applied to general convex loss, however, we gave some empirical evidence that it can perform well there. We believe our proof techniques might be extended beyond squared-error loss, perhaps using ideas in [6, 16, 77, 78].

We have seen a rich collection of work in the past decade targeting the optimality of estimators under computational (especially time expense) constraints. A lot of those results are established in the context of sparse PCA and related sparse-low-rank matrix problems, e.g. [14, 39, 40, 75, 129, 144]. The main focus of these work is usually comparing the statistical performance of the best polynomial-time algorithm with that of the "optimal" algorithm without any computational restrictions. By relating their statistical problem with a known NP problem [4], they can usually show the sub-optimality of polynomial-time algorithms under the famous conjecture $P \neq NP$. However, for the nonparametric regression problem in this chapter, there is a polynomial-time estimator that can achieve the global minimax-rate. It is of theoretical interest to know if there are any statistically rate-optimal online estimators that require less than $\Theta(n^{1+\frac{1}{2s+1}})$ time-expense: We hypothesize that there are not.

Chapter 4

REGRESSION IN TENSOR PRODUCT SPACES BY THE METHOD OF SIEVES

4.1 Introduction

Understanding the relationship between an outcome of interest and a set of predictive features is an important topic across domains of scientific research. To this end, one often needs to estimate an underlying predictive function, e.g., the conditional mean function, that best relates the features and the outcome using available noisy observations. During the past two decades, there has been extensive research focusing on nonparametric learning methods that only require the outcome to vary smoothly with the features.

One challenge of applying nonparametric methods in multivariate problem is the "curse of dimensionality" [94]. Briefly, as the number of features grows linearly, we need an exponentially growing number of samples to achieve a specified threshold of predictive performance. In real-world applications, although the total number of candidate features may be large, it is very likely that only a small proportion are conditionally associated with the outcome. This smaller number, D, of *active features* should be the primary driver of the difficulty of the problem, in a minimax sense. Sparse estimation [53, 118] is a vast field addressing such data science problems and developing effective estimation procedures, which is especially interesting when the total number of features, d, is much larger than D.

In this paper, we consider a nonparametric procedure that can simultaneously select important features and estimate the conditional mean function (using only those selected features). For this procedure, the estimation error scales favorably with total dimension (proportional to $\log(d)$). Moreover, engaging with a tensor product space additionally means that our active dimension, D, only shows up multiplicatively in a $\log^D(n)$ term (as compared

to modifying the rate of convergence in n in classical multivariate Sobolev/Holder spaces). Finally, our proposed framework is also seen to be empirically effective in our data example comparisons in Section 4.8.

The proposed method considers penalized sieve estimation in multivariate tensor product spaces. Sieve estimation, also known as projection estimation [119] or estimation using orthogonal functions [130], is a classical estimation strategy that has been shown to be very effective in univariate regression problems. Sieve estimators can suffer in classical multi-dimensional spaces (as a large number of basis vectors are required). This work can be seen as an attempt to extend the method of sieves into multivariate models that scale more efficiently (statistically and computationally) with the problem dimension. In the univariate case, sieve estimation procedures leverage some natural ordering of the orthogonal basis functions, which is usually based on frequency or polynomial degree. These natural choices, however, are no straightforward in multivariate settings. We also study appropriate strategies for reordering multivariate basis functions under tensor product models: This is critical for both method implementation and theoretical understanding.

Notation: In this paper, we will use bold letters to emphasize a Euclidean vector $\mathbf{x} \in \mathbb{R}^d$ when its dimension d is strictly greater than 1. The notation $\mathbf{x}^k \in \mathbb{R}$ is the k-th entry of $\mathbf{x} \in \mathbb{R}^d$ (rather the k-th power of it). We use $\mathbb{N}$ to represent the non-negative integer set $\{0, 1, 2, ...\}$, and use $\mathbb{N}^+$ for strictly positive integers $\{1, 2, 3, ...\}$. The $(\mathbb{N}^+)^d$ is the set of positive d-tuple grids: for example $(\mathbb{N}^+)^2 = \{(1,1), (1,2), (2,1), (3,1), (2,2), ...\}$.

4.2 Univariate Nonparametric Regression and Sieve Estimation

One can frame the goal of regression as estimating the function f that minimizes the population mean-squared error (MSE): $E[(Y - f(\mathbf{X}))^2]$, where Y is our outcome, and $\mathbf{X}$ are our features. We denote the distribution of $\mathbf{X}$ as ρ_X. The minimizer is the well-known condition mean function $f^0(\mathbf{X}) = E[Y|\mathbf{X}]$. In nonparametric regression, we assume f^0 belongs to some regular function space. An informative univariate model space that we will engage with is

the 1^{st}-order Sobolev space $W_1([0,1])$:

$$f^0 \in W_1([0,1]) = \{f \in L_2([0,1]) \ \mid \ f' \text{ exists and } f' \in L_2([0,1])\}. \tag{4.1}$$

Here f' can be understood as the weak derivative of f. In this framing, the set of piecewise linear functions is a subset of $W_1([0,1])$. Without loss of generality, we will assume feature $\mathbf{X}$ belongs to the d-dimensional unit cube $[0,1]^d$. Sieve estimation for f^0 in the W_1 space is built upon the following basic fact: It is possible to express f^0 as an (infinite) linear combination of some basis functions $\{\phi_j\}$. Among many possibilities, we choose the following function system as a concrete example:

$$\phi_1(x) = 1, \phi_j(x) = \sqrt{2}\cos((j-1)\pi x). \tag{4.2}$$

The aforementioned "infinite linear combination" can be expressed as: $f^0 = \sum_{j=1}^{\infty} \beta_j^0 \phi_j$. Moreover, it is also known that the (generalized) Fourier coefficients β_j^0 decay at a rate faster than $j^{-1.5}$ for $f^0 \in W_1([0,1])$. Therefore, it is plausible to truncate the infinite series at a certain finite level J_n: Using only the first more important J_n basis vectors, one can construct an estimator of f^0 with relatively small bias. Formally, a sieve estimator f_n takes the form that $\hat{f}_n = \sum_{j=1}^{J_n} \hat{\beta}_j \phi_j$ where the coefficients are determined using the available training data $\{(\mathbf{X}_i, Y_i), i = 1, ..., n\}$. The coefficients can be determined by solving least-square problems [119] or using stochastic approximation methods [143], both strategies would lead to rate-optimal generalization error (in a minimax-rate sense).

Remark: The cosine functions ψ_j presented above are not periodic over our domain themselves, and thus do not impose a periodic assumption on f^0. This is in contrast to periodic sine/cosine systems that are more commonly engaged with, and would imply a periodic assumption on f^0 [126]. One can add polynomials to the periodic systems to fit non-period functions [31]. For simplicity of exposition and to provide our readers a basis that is easy to implement, we choose to proceed with paper primarily using this cosine basis. For readers more familiar with the topic, the above rate statement on β_j^0 ("faster than $j^{-1.5}$") can be more precisely stated as Sobolev ellipsoid conditions. For more discussion, see Appendix 7.2.

4.3 Multivariate Nonparametric Models

4.3.1 Additive Models and Classical Smoothness Classes

In most real-world problems, we have more than one feature under consideration. In addition it is not always apriori clear which model space to use. The nonparametric additive model [36] has been seen as one of the most direct models for multivariate nonparametric learning problems. There, we assume features do not interact, or more formally that the regression function takes the following additive form:

$$f^0(\mathbf{x}) = \sum_{k=1}^{d} f_k^0(\mathbf{x}^k), \quad f_k^0 \in W_1([0,1]). \tag{4.3}$$

The lack of interaction between features makes the additive model quite restrictive. There are also some very flexible models widely discussed in the literature, such as Sobolev-type smooth function spaces. Formally, let $\mathbf{a} = (\mathbf{a}^1, ..., \mathbf{a}^d) \in (\mathbb{N})^d$, we define the (weak) partial derivative function $D^{\mathbf{a}} f$ of f as:

$$D^{\mathbf{a}} f = \frac{\partial^{\|\mathbf{a}\|_1}}{\partial x_1^{\mathbf{a}^1} \cdots \partial x_d^{\mathbf{a}^d}} f, \text{ where } \|\mathbf{a}\|_1 = \sum_{k=1}^{d} \mathbf{a}^k. \tag{4.4}$$

In this notation, we assume f^0 satisfies the following smoothness conditions:

$$f^0 \in W_s\left([0,1]^d\right) = \left\{ f \in L_2\left([0,1]^d\right) \;\middle|\; D^{\mathbf{a}} f \in L_2\left([0,1]^d\right) \text{ for all } \|\mathbf{a}\|_1 \leq s \right\}. \tag{4.5}$$

These types of smooth classes do not explicitly assume any specific form such as additivity, but as a cost, suffer substantially more from the curse of dimensionality. Specifically, the minimax rate (in MSE) of estimation in $W_s([0,1]^d)$ is of order $n^{-\frac{2s}{2s+d}}$ [111]. Although less likely to be miss-specified, this type of model is sometimes thought to be too large to explain the success of many machine learning methods, or be directly applied.

In the literature it is typical to put a more strict regularity requirement to cancel out the influence of dimension, that is, only considering smoother models in higher dimensions. Formally, this can be easily done by increasing the parameter s to ask for regular higher

order derivatives. For many statistical procedures that need to estimate conditional mean as a nuisance, e.g. semiparametric inference [59] and independence structure inference [134], we typically have to require the smoothness parameter s to be at least $d/2$. The resulting model space $W_{d/2}([0,1]^d)$ is sufficiently tame to allow estimators of f^0 that can satisfy certain (minimax rate) benchmark conditions. However, for d as small as 4, such a requirement already prevents f^0 from being a piece-wise linear function.

4.3.2 Tensor Product Models

Additive models (mentioned earlier) are an attractive approach for extending univariate smooth functions to multivariate regression. If the true regression function is nearly additive, then with a relatively small number of samples, one can fit a strong additive estimate. However, in some applications there may be important interactions between features to consider. One natural extension to the additive model is to include product-terms of basis functions between individual features. For example, we may consider:

$$f^0(\mathbf{x}) = \sum_{k=1}^{d} f_k^0(\mathbf{x}^k) + a(\mathbf{x}^1)b(\mathbf{x}^2) + C(\mathbf{x}^1)d(\mathbf{x}^3) + e(\mathbf{x}^1)f(\mathbf{x}^2)g(\mathbf{x}^3) + \ldots, \tag{4.6}$$

where all the univariate functions above belong to class of smooth functions $W_1([0,1])$. This type of model has been studied in the literature as a *Tensor Product Space model* [70]. In more compact notation:

$$f^0 \in S_1([0,1]^d) = \left\{ f = \sum_{m=1}^{N} \prod_{k=1}^{d} f_{mk}(\mathbf{x}^k) \text{ for some finite } N, \text{ and } f_{mk} \in W_1([0,1]) \right\}. \tag{4.7}$$

Although we defined the S_1 space in (4.7) by addition and multiplication of univariate regular functions, there is an (almost) equivalent characterization of it in the language of partial derivatives:

$$S_1([0,1]^d) = \left\{ f \in L_2\left([0,1]^d\right) \;\middle|\; D^{\mathbf{a}} f \in L_2\left([0,1]^d\right) \text{ for all } \|\mathbf{a}\|_\infty \leq 1 \right\}. \tag{4.8}$$

Function spaces similar to (4.8) are called Sobolev spaces with dominating mixed derivatives. They are also characterized as the tensor product spaces of univariate Sobolev spaces

$W_1([0,1])$. Compared with the (isotropic) Sobolev spaces defined in (4.5), tensor product spaces may appear to be formally similar, but have different (and favorable) properties related to statistical estimation. For function space $W_1([0,1]^d)$, we required regular partial derivatives for any index $\mathbf{a}$ satisfying $\|\mathbf{a}\|_1 \leq 1$. But for tensor product space $S_1([0,1]^d)$, we require partial derivatives for those indices satisfying $\|\mathbf{a}\|_\infty \leq 1$. The latter requirement is strictly stronger and as the dimension d increases, the difference between these two requirements becomes more meaningful. At the same time, the $S_1([0,1]^d)$ space requires less regularity than the d-th order isotropic Sobolev space $W_d([0,1]^d)$. In particular, assuming $f^0 \in W_d([0,1]^d)$ means that $\frac{\partial^d}{\partial^d \mathbf{x}^k} f^0$ exists and is square-integrable for any $k = 1, 2, ..., d$, however functions in $S_1([0,1])$ space do not need to have second partial derivatives $\frac{\partial^2}{\partial^2 \mathbf{x}^k} f$ for any k (so piece-wise linear functions can be elements of $S_1([0,1]^d)$). More formally, we have the following inclusion relationship:

$$W_d([0,1]^d) \subsetneq S_1([0,1]^d) \subsetneq W_1([0,1]^d). \tag{4.9}$$

Functions in S_1 are allowed to have different degrees of regularity in different directions. Specifically, they have almost minimal smoothness in the coordinate axis directions. These are the directions in which people believe most variation should be observed, which is also supported by the success of additive models in practice.

4.4 Literature Review

In this section, we will provide a quick overview of the literature on tensor product models in statistical learning and nonparametric sieve estimators.

In [70], the author presents regression estimators in *tensor product models* by the method of smoothing spline / kernel ridge regression. The estimators achieve the nonparametric minimax rate but typically have a high computational expense. Compared with the proposal in this work, there is limited ability to apply variable selection or dimension adaptivity. This makes that work less applicable to sparse nonparametric models. Other work in this line of research includes [127], [69] and [41].

In addition to using product reproducing kernels, other types of product bases are also used to construct multivariate regression estimators. For example, these are used in multivariate adaptive regression spline [35] and the highly adaptive lasso [9]. This class of methods select a collection of adaptive basis functions that center at the training data points. The set of basis functions, unlike the sieve estimator basis, are usually not orthogonal to each other under any natural measures. More comprehensive discussion can be found in the monograph [44].

A lot of work has been done over the last decade to adapt the tensor product model to ultra-high dimensional settings. This line of research typically assumes that the features must have a main effect on the outcome in order to have second order interaction effects (formalized as some heredity assumptions). These methods target application cases when the feature dimension is very large and computational resources are restricted (For example, assuming d^2 derived-features would not fit into the memory). See [50], [114], and the references therein for a more detailed description of these computationally efficient methods.

In contrast to the kernel or spline based methods, in this paper we will discuss how to apply sieve estimators in tensor product models. In [119], the author presents the classical least-square sieve estimator (termed as a projection estimator) with theoretical discussion (many parts in our exposition will be of that flavor). In [19], the author provides an extensive review of commonly used/ theoretically interesting sieve bases. [30] provides an extensive review of the method of sieves in density estimation. See also Section 7.5 of [29] for a discussion of sieve estimation for multivariate analytic functions. In [55], the authors discuss estimation with orthogonal series under additive models. However, there is no existing work that formally engages with tractable sieve estimation procedures under tensor product models to the best knowledge of the authors. In contrast, it has been repeatedly discussed in the literature to directly generalize univariate sieve estimators to multivariate settings with estimators of the form (eg. here we take the dimension = 3)

$$\hat{f}_n(\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3) = \sum_{i=1}^{J_n} \sum_{j=1}^{J_n} \sum_{k=1}^{J_n} \theta_{ijk} \psi_i(\mathbf{x}^1) \psi_j(\mathbf{x}^2) \psi_k(\mathbf{x}^3) \tag{4.10}$$

This kind of direct extension does not lead to rate-optimal estimators in commonly discussed function classes, and is not computationally scalable to even moderate dimension d in practice (J_n^d basis functions are used).

4.5 Least-square Sieve Estimators in Tensor Product Models

Sieve estimation leverages the fact that smooth functions can be written as an infinite linear combination of some basis functions ϕ_j and the coefficients decay quickly. To construct estimates, we can use a truncated series to balance the approximation and estimation errors. Since functions in $S_1([0,1]^d)$ can be approximately written as the addition and multiplication of a set of univariate functions in $W_1([0,1])$, we may expect a function $f \in S_1([0,1]^d)$ to have the expansion

$$f(\mathbf{x}) = \sum_{\mathbf{j}\in(\mathbb{N}^+)^d} \beta_{\mathbf{j}}^0 \psi_{\mathbf{j}}(\mathbf{x}), \text{ for some } \beta_{\mathbf{j}}^0 \in \mathbb{R}, \tag{4.11}$$

where $\mathbf{j} = (\mathbf{j}^1, \mathbf{j}^2, .., \mathbf{j}^d) \in (\mathbb{N}^+)^d$, and $\psi_{\mathbf{j}}$ is a product of the univariate cosine basis $\psi_{\mathbf{j}}(\mathbf{x}) = \prod_{k=1}^d \phi_{\mathbf{j}_k}(\mathbf{x}^k)$ described in (4.2).

In contrast to the univariate case, there is no single obvious natural ordering of the basis functions $\psi_{\mathbf{j}}$ since they are indexed by some d-tuples $\mathbf{j}$. To apply sieve estimation in tensor product spaces (or for any multivariate nonparametric models), we need to establish an order on $\{\psi_{\mathbf{j}}\}$ and determine which basis functions should be used for each n. In other words, we need to unravel the *set* $\{\psi_{\mathbf{j}}, \mathbf{j} \in (\mathbb{N}^+)^d\}$ to a *sequence* of functions $\{\psi_j, j \in \mathbb{N}^+\}$. They contain the same set of functions but the latter is an ordered set.

Let (ψ_j) be the sequence of functions unravelled from $\{\psi_{\mathbf{j}}\}$ (we postpone the details of the rearrangement rule to Section 4.6). In the new notation, any $f \in S_1([0,1]^d)$ has the expansion $f(\mathbf{x}) = \sum_{j=1}^\infty \beta_j^0 \psi_j(\mathbf{x})$, $\beta_j^0 \in \mathbb{R}$. To perform sieve estimation in $S_1([0,1]^d)$, we also truncate the series at a proper level J_n. The least-square sieve estimator f_n^{OLS} is $f_n^{OLS}(\mathbf{x}) = \sum_{j=1}^{J_n} \beta_{jn}^{OLS} \psi_j(\mathbf{x})$, whose coefficients are the minimizers of the following empirical

least-squares problem:

$$\left(\beta_{1n}^{OLS},...,\beta_{J_nn}^{OLS}\right) = \underset{(\beta_1,...,\beta_{J_n})\in\mathbb{R}^{J_n}}{\operatorname{argmin}} \sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{J_n}\beta_j\psi_j(\mathbf{X}_i)\right)^2 \tag{4.12}$$

Using analysis tools from empirical process theory, it is possible to derive some theoretical guarantees regarding the performance of f_n^{OLS}.

Theorem 4.5.1. *Suppose* $\{(\mathbf{X}_i, Y_i) \in [0,1]^d \times \mathbb{R}, i = 1,2,...,n\}$ *is an i.i.d. training sample and the true regression function* $f^0 \in S_1([0,1]^d)$*. Let* $\epsilon_i = Y_i - f^0(\mathbf{X}_i)$ *be sub-Gaussian, mean-zero random variables. We further assume that the distribution of* $\mathbf{X}$*,* ρ_X*, is continuous with an upper-bounded density function.*

Then, for the least-square sieve estimator f_n^{OLS}*, constructed with product of non-periodic cosine basis functions* (4.2)*, we have:*

$$E\left\|f_n^{OLS} - f^0\right\|_{L_2(\rho_X)}^2 = O\left(\left(\frac{\log^{d-1}(n)}{n}\right)^{2/3}\log(n)\right), \tag{4.13}$$

when $J_n = \Theta\left(n^{1/3}\log^{2(d-1)/3}(n)\right)$*. The order of including the multivariate product basis is described in Section 4.6.*

The proof of the above statement is very similar to that of Theorem 1 in [142]. However, to determine the proper truncation level J_n and approximation error, we need the new technical results presented in Lemma 7.3.6. The above theoretical guarantee is almost minimax-optimal [70], up to a logarithm term.

The generalization MSE of this least-squares sieve estimator only differs from $n^{-2/3}$ (the rate for univariate Sobolev space $W_1([0,1])$) by a polylog term (with the dimension d in the exponent). This is much improved as compared with estimation in spaces such as $W_s([0,1]^d)$. For that classical space, the minimax rate is of order $n^{-2s/(2s+d)}$. The dimension d shows up in the exponent of n rather than $\log n$. That horrible dependence on the dimension is one manifestation of the curse of dimensionality. It is much alleviated, as we have shown, in tensor product spaces. Many semiparametric procedures require convergence of intermediate

components at a rate of at least $n^{-1/2}$ [59]. Classical Sobolev models must assume $s \geq d/2$ to give such a guarantee. This requirement may be too strong for many applications: specifically, it already rules out all the piece-wise linear truths when $d \geq 4$.

4.6 Important Technical Details: Unravelling

In this section, we are going to talk about how to rearrange a set of functions $\psi_{\mathbf{j}}$ indexed by d-tuple to a sequence of functions ψ_j. For ease of discussion, we will term this kind of rearrangement process as *unravelling*. Now we present our proposed unravelling rule for tensor product models.

In Figure 4.1, we present how to rearrange two dimensional grids $(\mathbb{N}^+)^2$ into a sequence ($d = 2$). We first assign a number $c_{\mathbf{j}}$ to each grid element $\mathbf{j} \in (\mathbb{N}^+)^2$ that equals to the elemental product $c_{\mathbf{j}} = \mathbf{j}^1 \cdot \mathbf{j}^2$. We then rearrange the grid on the left based on the product value $c_{\mathbf{j}}$ in increasing order. In the right panel of Figure 4.1, we can see grid-elements assigned with smaller $c_{\mathbf{j}}$ values get a more prioritized position in the sequence indexed by $j \in \mathbb{N}^+$. For example, $(1, 1)$ is mapped to the first element on the right because it has the smallest product. In contrast, $(2, 2)$ gets the 7-th position. For grids with the same $c_{\mathbf{j}}$ values (such as $(1, 2)$ and $(2, 1)$), their relative order can be defined arbitrarily. We put $(2, 1)$ in front of $(1, 2)$ because it has a larger value in the first dimension. In many parts of the theoretical analysis, we are interested in the magnitude of the unravelled sequence (c_j) (the series presented in right panel of Figure 4.1).

This unraveling in $(\mathbb{N}^+)^d$ directly induces a rule for rearranging the basis functions $\{\psi_{\mathbf{j}}\}$ in a sequence (ψ_j). Using the $c_{\mathbf{j}}$-unravelling rule presented in Figure 4.1, the first several basis functions in the unravelled sequence are $\psi_1 = \psi_{(1,1)}$, $\psi_2 = \psi_{(2,1)}$, $\psi_3 = \psi_{(1,2)}$ and $\psi_7 = \psi_{(2,2)}$. These are exactly the basis functions we used in constructing least-square sieve estimators in (4.12). We now give a formal presentation of the above $c_{\mathbf{j}}$-unravelling rule:

Definition 4.6.1. *Given a function $c : (\mathbb{N}^+)^d \to \mathbb{N}^+$ defined on the d-tuple grids , we define $\mathcal{U}(\mathbf{m}) : (\mathbb{N}^+)^d \to \mathbb{N}^+$ to be the unique surjective mapping satisfying the following conditions:*

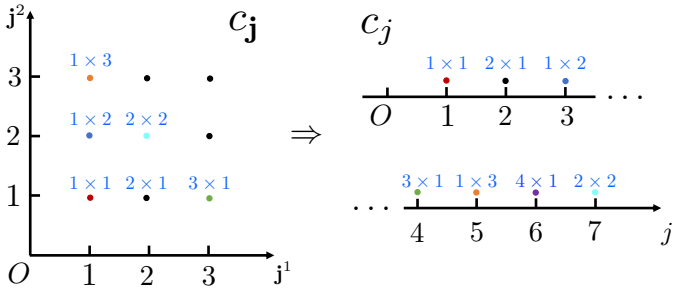

Figure 4.1: Illustration of unravelling. The rule function is $c_{\mathbf{j}} = \prod_{k=1}^{d} \mathbf{j}^k$.

1. $\mathcal{U}(\mathbf{m}) \leq \mathcal{U}(\mathbf{n})$ *if and only if* $c_{\mathbf{m}} \leq c_{\mathbf{n}}$;

2. *(tie-breaker) For* $\mathbf{m}, \mathbf{n} \in (\mathbb{N}^+)^d$ *that have the same* c *values:* $c_{\mathbf{m}} = c_{\mathbf{n}}$*, we set* $\mathcal{U}(\mathbf{m}) < \mathcal{U}(\mathbf{n})$ *if and only the following condition holds: There exists a value* $k \in \{1, 2, ..., d\}$ *such that,* $\mathbf{m}^l = \mathbf{n}^l$ *for all* $l \leq k$*, but* $\mathbf{m}^k > \mathbf{n}^k$.

We call such a mapping, $\mathcal{U}$*, the* $c_{\mathbf{j}}$*-unravelling rule.*

Condition 1 in Definition 4.6.1 is essential: grids with smaller $c_{\mathbf{j}}$ values get a more prioritized position in the unravelled sequence. Condition 2 is an arbitrary tie-breaking rule and could be modified. To summarize, for each function c defined on $(\mathbb{N}^+)^d$, there is a uniquely defined unravelling rule $\mathcal{U}$, which gives one way to rearrange a set of basis functions into a sequence. For tensor product models such as $S_1([0,1]^d)$, we propose using the rule defined by $c_{\mathbf{j}} = \prod_{k=1}^{d} \mathbf{j}^k$, which leads to computationally more feasible and statistically near-optimal estimators.

4.7 Penalized Sieve Estimators in Sparse Models

In this section, we will discuss how to apply l_1-penalized sieve estimators for nonparametric sparse models. The difference between this section and the previous is analogous to the difference between sparse additive models [90] and additive models (dicussed in Section 4.3), though the technical tools employed differ.

Although there may be a substantial number of features collected, it is common that only a small active subset of those features are needed to build the optimal predictive model. We will show that, similar to many other sparse methods, our proposed method is relatively robust to the ambient dimension d. It is the active dimension of the problem that has a significant impact. We now formalize this nonparametric sparse model:

Condition 4.7.1. *There exists a D-variate function $f^* : [0,1]^D \to \mathbb{R}$, and a set of indices $\{k_1, .., k_D\} \subset \{1, 2, ..., d\}$ such that for any $\mathbf{u} \in [0,1]^d$: we have $f^0(\mathbf{u}) = f^*(\mathbf{u}^{k_1}, \mathbf{u}^{k_2}, ..., \mathbf{u}^{k_D})$. Moreover, we assume*

$$f^* \in S_1([0,1]^D).$$

The first half of Condition 4.7.1 formally states that there are D features that have dominating association with the outcome; The later half is a smoothness assumption, which can potentially be replaced by other nonparametric model assumptions. Here, we take the S_1 space as an example for presenting our ideas, for general discussion and theory, see Condition 7.3.8 in Appendix 7.3.3.

In Sections 4.5 and 4.6, we discussed the need to order the multivariate basis functions; we additionally showed that using the unravelling rule $c_{\mathbf{j}} = \prod_{k=1}^d \mathbf{j}^k$ would lead to nearly rate-optimal least-square estimators (up to polylog). In the sparse model setting, the unravelling rule is very similar except that we allow ourselves to remove some higher-order interaction terms for computational ease. In particular, we begin with a conservative guess D' for the active dimension D. We then remove any interactions of order $> D'$. So long as $D \leq D'$ this will not affect the theoretical performance of our estimator. Formally, our new unravelling rule is:

Condition 4.7.2. *Let $\{\phi_j\}$ be the univariate cosine basis: $\phi_1(x) = 1$, $\phi_j(x) = \sqrt{2}\cos((j-1)\pi x)$. Consider their natural d-dimensional product extension $\psi_{\mathbf{j}}(\mathbf{x}) = \prod_{k=1}^d \phi_{\mathbf{j}^k}(\mathbf{x}^k)$, denote ψ_j to be the $c_{\mathbf{j}}$-unravelling sequence of $\{\psi_{\mathbf{j}}\}$. The unravelling rule $c_{\mathbf{j}}$ is defined as*

$$c_{\mathbf{j}} = \begin{cases} \prod_{k=1}^d \mathbf{j}^k & \text{, if at most } D' \text{ entries of } \mathbf{j} \text{ are greater than } 1 \\ \infty & \text{, otherwise} \end{cases} \tag{4.14}$$

Suppose $d = 3$ and we choose the working dimension $D' = 2$. Then $\psi_{(1,1,1)}$ will get the first place when unravelling $\{\psi_{\mathbf{j}}\}$ to the (ψ_j) sequence. Similarly, $\psi_{(2,1,1)}$ gets the second position and $\psi_{(1,2,1)}$ gets the third. However, basis functions that vary in more than $D' = 2$ dimensions will not be used for our estimate. For example, $\psi_{(2,2,2)}(\mathbf{x}) = 2^{3/2}\prod_{k=1}^{3}\cos(\pi \mathbf{x}^k)$ is excluded since it varies in all three dimensions. We formalize this using an infinite value for the index in our rule (4.14).

For problems with higher feature dimension d and limited samples, the empirical least-squares problem (4.12) is likely to be under-determined (we will have more basis functions than samples), and thus regularization is required for numerical optimization. In addition, basis functions from non-active features should have 0 coefficient. Toward this end, we add a sparsity-inducing penalty. More specifically our estimator is given by solving the following penalized optimization problem:

$$\left(\beta_1^{PLS}, ..., \beta_{J_n}^{PLS}\right) = \underset{(\beta_1,..,\beta_{J_n})\in\mathbb{R}^{J_n}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\left\{Y_i - \sum_{j=1}^{J_n}\beta_j \cdot \psi_j(\mathbf{X}_i)\right\}^2 + \lambda_n \sum_{j=1}^{J_n}|\beta_j|, \tag{4.15}$$

where our final estimate is given by $f_n^{PLS}(\mathbf{x}) = \sum_{j=1}^{J_n}\beta_j^{PLS}\psi_j(\mathbf{x})$. In Appendix 7.1.2 we include more details on the implementation of the above method. We have the following theoretical guarantee for this estimator's generalization error:

Theorem 4.7.3. *Suppose $\{(\mathbf{X}_i, Y_i) \in [0,1]^d \times \mathbb{R}, i = 1, 2, ..., n\}$ is an i.i.d. training sample and the true regression function f^0 satisfies Condition 4.7.1. Let $\epsilon_i = Y_i - f^0(\mathbf{X}_i)$ be sub-Gaussian, mean-zero random variables. We further assume that the distribution of $\mathbf{X}$, ρ_X, is continuous with a bounded density function (from above and away from zero), and the working dimension D' in Condition 4.7.2 is no smaller than the active dimension D in Condition 4.7.1.*

Then, for the l_1-penalized sieve estimator f_n^{PLS}, constructed with basis functions described in Condition 4.7.2, we have:

$$\left\|f_n^{PLS} - f^0\right\|_{L_2(\rho_X)}^2 = O_p\left(\log(d)\log(n)\left(\frac{\log^{D-1}(n)}{n}\right)^{2/3}\right), \tag{4.16}$$

when $J_n = C(D)d^{D'}n^{1/3}(\log n)^{D'-1}$ *and* $\lambda_n = (\log(J_n)/n)^{1/2}$. *Here* $C(D)$ *is a constant that only depends on* D.

This convergence rate for f_n^{PLS} looks similar to the rate obtained for the unpenalized estimator f_n^{OLS} with two substantial differences: 1) The $\log^{d-1}(n)$ has been replaced by $\log^{D-1}(n)$ which now only involves the active dimension; and 2) The ambient dimension d is only included through a $\log(d)$ term (as is common in sparse regression).

The l_1-penalized optimization problem in (4.15) can be solved directly using standard lasso solvers such as `glmnet` [106]. The overall task of fitting the nonparametric estimator f_n^{PLS} can be done with R package `Sieve`. Asymptotically, the time complexity for constructing the above l_1-penalized sieve estimator is of order $O(nJ_n) = O(d^{D'}n^{4/3}\log^{D'-1}n)$. In contrast, standard applications of reproducing kernel ridge regression require $\Theta(n^3)$ computation and give no adaptivity guarantees under feature sparsity. Computationally, the proposed sieve estimator is more suitable for large data sets as its dependency on sample size is almost linear. Other theoretically guaranteed methods, such as highly adaptive lasso [9], require solving optimization problems that scale as $2^d n$, which is substantially more resource intensive than the proposed method.

4.8 Numerical Examples

So far in this manuscript, we have introduced and discussed the (dense and sparse) tensor product models and the theoretical performance guarantees of sieve estimators. In this section, we will demonstrate the finite-sample performance of the proposed methods and their applicability in practice via simulated and real data sets. The methods discussed in this manuscript, penalized and least-square sieve estimators, are implemented in the R package `Sieve`. Currently, the package is available on the Comprehensive R Archive Network (CRAN).

We first present some numerical results based on simulated data sets. In this section we will consider two types of true regression functions. (We give an additional example in

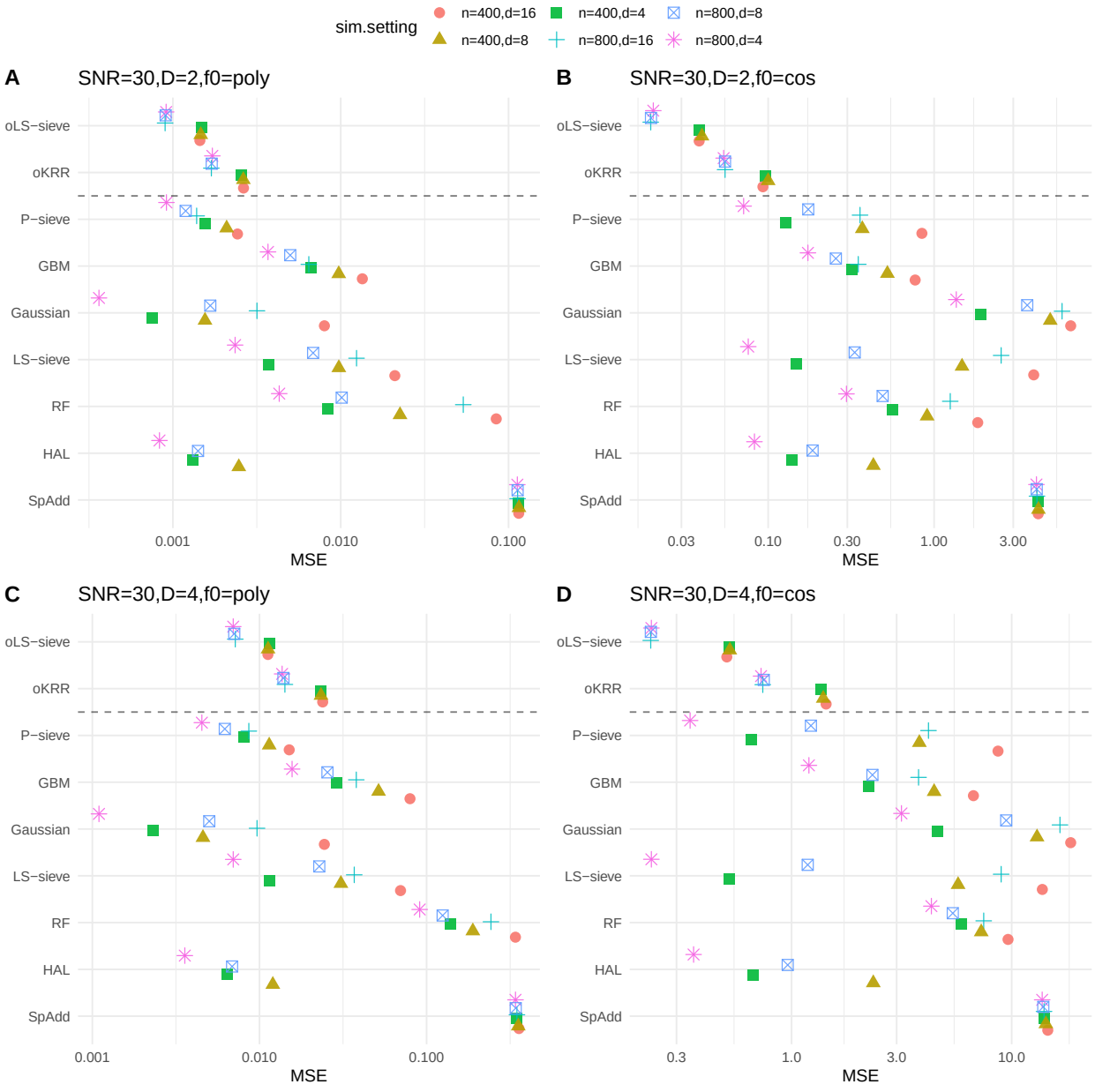

Figure 4.2: Simulation study results. Low noise settings, SNR = 30.

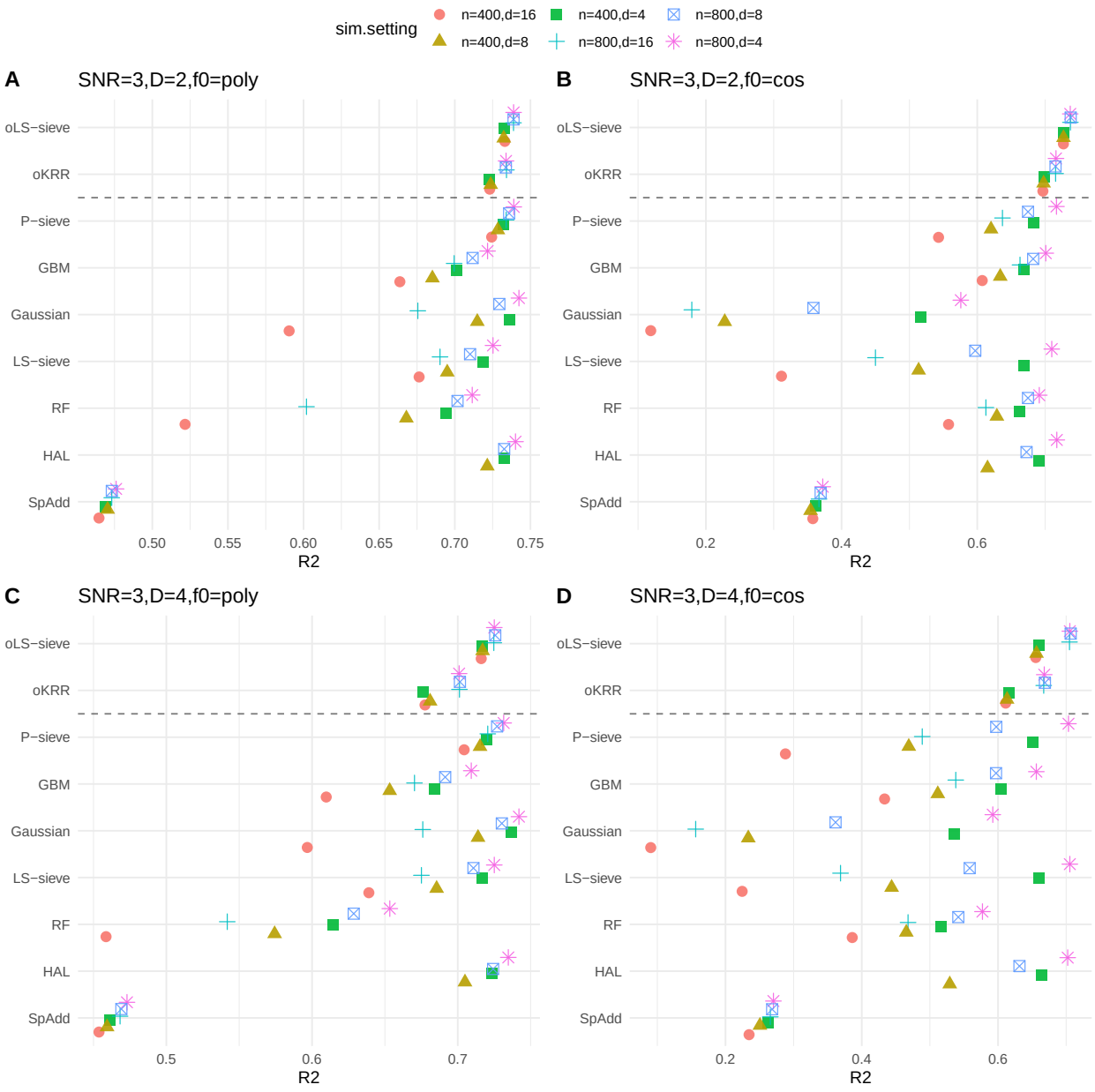

Figure 4.3: Simulation study results. High noise settings, SNR = 3.

Table 4.1: Functional form and highest interaction order for simulated data

	Example 1	Example 2
Truth form	$\sum_{k=1}^{D-1} Leg(2(\mathbf{x}^k - 0.5), 3) + Leg(2(\mathbf{x}^k - 0.5), 2) \cdot Leg(2(\mathbf{x}^{k+1} - 0.5), 2)$	$\sum_{\mathbf{j} \in (\mathbb{N}^+)^d, c_\mathbf{j} \leq 8} \prod_{k=1}^{D} \cos((\mathbf{j}^k - 1)\pi \mathbf{x}^k)$
Interaction	highest: second order	highest: third order

Appendix 7.1.1 where the true conditional means only contain interaction terms without main effects. In this setting the proposed methods perform much better than tree-based methods.) In Table 4.1 we present the detailed functional forms of the true regression functions. The $Leg(x, j)$ function in the table is the j-th Legendre polynomial: $Leg(x, 2) = x, \quad Leg(x, 3) = (3x^2 - 1)/2$. And the function $c_\mathbf{j}$ is $c_\mathbf{j} = \prod_{k=1}^{d} \mathbf{j}^k$.

In the simulation study, we considered active dimension $D \in \{2, 4\}$ and ambient dimension $d \in \{4, 8, 16\}$. We used signal-noise-ratio (SNR) = 3 and 30 with normally distributed noise random variables. Here SNR is defined as the ratio between the squared 2-norm of f^0 and the variance of the noise variables. This means the oracle (best possible) testing R^2 should be 0.75 (SNR = 3) and 0.97 (SNR = 30). We choose sample size $n \in \{400, 800\}$. The feature vectors $\mathbf{X}$ we consider are uniformly distributed over the $[0, 1]^d$ cube. We performed 100 simulations for each setting. We use oracle hyperparameters for each method (number of basis functions, regularization parameter, number of trees, etc.), which is determined based on an independent $n = 2000$ testing data set.

The regression estimators we considered in the simulation study are: sieve estimators proposed in this work (least-square and penalized), random forest (RF, R package randomForest), gradient boosting (GBM, R package gbm), Gaussian kernel ridge regression (also known as radial kernel support vector machine), highly adaptive lasso (HAL, R package hal9001, only applied for the lower dimension case $d = 4$ due to the exponential memory

Table 4.2: Basic information for public data sets used in performance comparison

Name	Sample size (n)	Feature dimension (d)	Feature type	References
gdp	616	6	6 continuous	[72]
fev	654	4	2 continuous, 2 binary	[95]
fev50	654	54	52 continuous, 2 binary	–
bio	779	9	9 continuous	[43]
aba	4177	8	7 continuous, 1 categorical	[131]
supc	21263	81	81 continuous	[47]

requirement of this method) and sparse additive models. We also include some oracle estimators that know which D dimensions are truly associated with the outcome Y in order to demonstrate the dimension adaptivity of the other methods. The univariate basis ϕ_j we used for sieve estimators are: $\phi_1(x) = 1$, $\phi_j(x) = \sin((j + 1/2)\pi x)$ (sine basis, for the f^0_{cos} settings) and $\phi_j(x) = \cos((j - 1)\pi x)$ (cosine basis, for all the other truth f^0). The oracle kernel ridge regression method, denoted as oKRR in Figure 4.2 and Figure 4.3, uses the reproducing kernel of $S_1([0, 1]^D)$, see Appendix 7.2. In Fig 4.2, we present the results under high signal-noise-ratio settings and we evaluate the performance of each method using (absolute) testing MSE. In Fig 4.3, the larger noise settings, model performance is evaluated via testing R^2. Sometimes R^2 is more interpretable in practice than absolute MSE, but we chose to present absolute MSE in Fig 4.2 simply because it can differentiate methods better (all methods have high R^2 values in some settings).

We also compare the predictive performance of these methods on 5 publicly available data sets. Some basic information for the data sets is reported in Table 4.2. In Figure 4.4, we present the relative testing MSE and (absolute) R^2 of each method. We saved 30% of the samples as the test set and the hyperparameters of each method are determined using

a 5-fold cross-validation on the training set (more details presented in Table 7.1 of the supplement). The `fev50` data set combines the true outcome and features from `fev`, with 50 artificially constructed non-informative features (independent, Unif$[0, 1]$). We use this data set as a moderately high-dimensional, sparse feature example. One of the data sets, `supc`, has been used as an example to demonstrate the effectiveness of tree-based methods [47], so we also include it for a more comprehensive comparison. We only applied highly adaptive lasso to 3 data sets and Gaussian kernel ridge regression to 5 data sets due to their high computational resource requirement: These would not efficiently run on a machine with 1 Intel Core m3 processor, 1.2 GHz, with 8 GB of RAM. The linear model with all interaction terms is not applicable to `fev50` because the empirical problem is not well-posed without further modification (number of coefficients is larger than the sample size).

We compared sieve estimators based on different univariate bases ϕ_j, including polynomial, cosine basis and sine basis (the basis defined earlier in this section), as well as a combination of polynomial and trigonometric functions [31]. The performance of penalized sieve methods using different basis functions is quite similar. The random forest estimator is more sensitive to the extra dimensions of `fev50` than penalized sieve and GBM. For more information on the data sets, see Table 4.2 in the supplementary material.

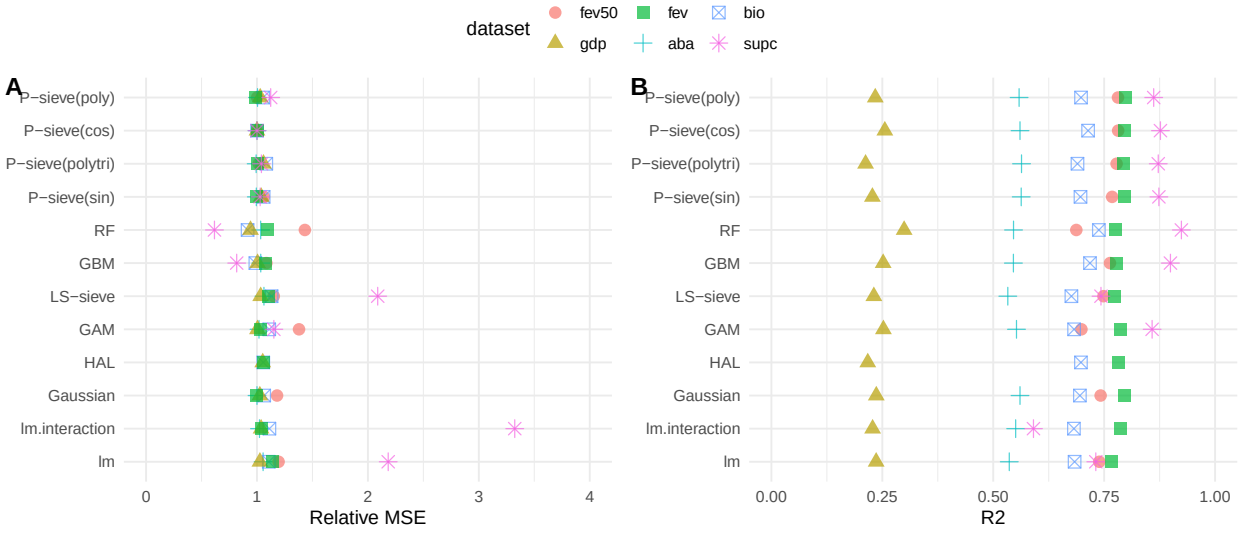

Figure 4.4: Relative MSE and R^2 on real data sets. The MSE values are normalized to that of penalized sieve estimator with cosine functions. Methods requiring significantly more computational resource are not reported.

Chapter 5

SUPPLEMENTARY MATERIALS FOR CHAPTER 2

5.1 Supplementary Discussion on RKHS

In the main text we gave two equivalent definitions of RKHS: one based on the reproducing property and another one based on the Mercer expansion of the kernel.

The proposed method directly works with the eigenfunctions ψ_j, and it does not directly approximate either the kernel function K or the kernel matrix $\mathbb{K}$. Although in many cases we start with a Mercer kernel in hand and calculate its eigendecomposition afterwards, it is not uncommon to begin with features and then attempt to calculate a closed-form of an implied kernel. This situation suits perfectly with our method: for the well-known the smoothing spline method proposed in [126, Chapter 2], the author starts with $\psi_j(x) = \sin(2j\pi x), \cos(2j\pi x)$ and shows us how to get the closed-form of the reproducing kernel for periodic Sobolev space $W_m^0(\text{per})$. However, such a Bernoulli polynomial closed-form of the kernel is no longer available when m is not an integer, which corresponds to a fractional Sobolev space case; when considering kernel space on sphere $\mathbb{S}^2$, some effort is required to obtain the closed-form expression even for simple cases ([60], [79]), but the features are just orthonormal spherical harmonics; for multiscale kernels defined by compactly-supported wavelet eigenfunctions [84] or Legendre polynomials [136, Section 3.3.2], it is also simplest to work directly with features rather than attempting to identify a closed-form expression for the implied kernel.

In the main text we provide the Mercer expansion of a Sobolev space $W_1^0([0,1])$. We also state the (correct) expansion for Gaussian kernel (there are several versions in the literature that are not correctly normalized):

When $\bar{\rho}_X$ has density (w.r.t Lebesgue measure on $\mathbb{R}$) $\bar{p}_X = \frac{\alpha}{\sqrt{\pi}} \exp(-\alpha^2 x^2)$, we have the

expansion of Gaussian kernel $K(x,z) = \exp(-\epsilon^2|x-z|^2)$ with

$$\begin{aligned} \lambda_j &= \sqrt{\frac{\alpha^2}{\alpha^2+\delta^2+\epsilon^2}}\left(\frac{\epsilon^2}{\alpha^2+\delta^2+\epsilon^2}\right)^{j-1} \\ \psi_j(x) &= \gamma_j \exp(-\delta^2 x^2) H_{j-1}(\alpha\beta x) \end{aligned} \tag{5.1}$$

where the H_j are Hermite polynomials of degree j, and

$$\beta = \left(1+\left(\frac{2\epsilon}{\alpha}\right)^2\right)^{1/4},\ \gamma_j = \sqrt{\frac{\beta}{2^{j-1}\Gamma(j)}},\ \delta^2 = \frac{\alpha^2}{2}\left(\beta^2-1\right) \tag{5.2}$$

The multivariate Gaussian kernel's eigenfunctions and eigenvalues are just the tensor product of the 1-dimension Gaussian kernel. Formally, the multivariate Gaussian kernel $K(\mathbf{x},\mathbf{z}) = \exp(-\epsilon^2\|\mathbf{x}-\mathbf{z}\|^2)$ has the following expansion:

$$K(\mathbf{x},\mathbf{z}) = \sum_{\mathbf{j}\in\mathbb{N}^d} \lambda_{\mathbf{j}}^* \psi_{\mathbf{j}}^*(\mathbf{x})\psi_{\mathbf{j}}^*(\mathbf{z}) \tag{5.3}$$

where the eigenvalues and eigenfunctions are related to (5.1) as

$$\lambda_{\mathbf{j}}^* = \prod_{l=1}^{d} \lambda_{\mathbf{j}_l},\ \psi_{\mathbf{j}}^*(\mathbf{x}) = \prod_{l=1}^{d} \psi_{\mathbf{j}_l}(x^{(l)}), \tag{5.4}$$

where $x^{(l)}$ is the l-th component of $x \in \mathbb{R}^d$. There are also available numerical methods (independent of $(X_i, Y_i)'s$) for approximating kernel eigenfunctions in cases where analytical forms are not available, see [89, 98], [93, Section 4.3], [13] and [33, Chapter 12].

There is also an interesting formal similarity between Mercer expansions and Bonchner's theorem (see, e.g. [87]) which gives rise to random Fourier feature-based methods. On one hand, we have the Mercer expansion:

$$K(x,z) = \sum_{j=1}^{\infty} \lambda(j)\psi(x,j)\psi(z,j) \tag{5.5}$$

On the other hand, the positive-definite (real-valued) kernel has a convolutional representation by Bonchner's theorem [87]):

$$K(x,z) = \int_{\mathcal{X}\times[0,2\pi]} p(\omega,b)\cos(\omega^\top x + b)\cos\left(\omega^\top z + b\right) d\omega db \tag{5.6}$$

The random Fourier feature expansion (5.6) uses a set of basis functions (cosines) that is not sensitive to the expanded kernel. Only the probability distribution we sample ω from depends on the kernel. Such a choice may bring some convenience in application, but at the price of using an approximation that converges to the kernel much slower. Another difference is in the basis selection strategy: For the Mercer expansion it is very straightforward – we choose the eigenfunctions corresponding to larger eigenvalues. By this strategy, we can ensure the features we choose are more important and orthogonal to each other w.r.t. RKHS inner product. For random feature-based methodologies, one has to sample from a probability distribution because there are uncountably infinitely many ω (versus countably infinite j) and there is less we can say about the geometric properties of random features [139].

Our readers can also find expansions of various kernels in [128, 126, 32, 132, 104, 68, 34]. There are also several existing online nonparametric learning methods not mentioned in the main text, e.g. [61, 138, 96, 2, 135] .

5.2 Proof of Theorem 3

We can decompose the $L^2_{\rho_X}$-distance(i.e. $\|\cdot\|_2$-distance) between $\hat{f}_{n,N}$ and f_ρ into two parts by inserting a f_N function in between. Recall the definition of the previous two are:

$$\begin{aligned} \hat{f}_n &:= \operatorname*{argmin}_{f\in\mathcal{F}_N} \frac{1}{n}\sum_{i=1}^{n} \left(Y_i - f\left(X_i\right)\right)^2 \\ f_\rho &:= \operatorname*{argmin}_{f\in L^2_{\rho_X}} \int_{\mathcal{X}\times\mathbb{R}} \left(Y - f\left(X\right)\right)^2 d\rho(X,Y) \end{aligned} \tag{5.7}$$

where $\mathcal{F}_N$ is a subset of the N-dimension vector space spanned by $\psi_1, ..., \psi_N$:

$$\mathcal{F}_N = \mathcal{F}_N(M) := \{f \in L^2_{\rho_X} \mid f \in \operatorname{span}(\psi_1, ..., \psi_N), \|f\|_\infty < M\} \tag{5.8}$$

. We insert a deterministic function f_N in-between to facilitate the use of the triangle inequality.

$$f_N := \operatorname*{argmin}_{f\in\mathcal{F}_N} \int_{\mathcal{X}\times\mathbb{R}} \left(Y - f\left(X\right)\right)^2 d\rho(X,Y) \tag{5.9}$$

So we have the following decomposition of $L^2_{\rho_X}$ distance:

$$E\|\hat{f}_{n,N} - f_\rho\|_2 \leq E\|\hat{f}_{n,N} - f_N\|_2 + \|\hat{f}_N - f_\rho\|_2 \tag{5.10}$$

If we can bound the two terms at the correct rates separately at the desired order, combining them together would give the result in Theorem 3.

5.2.1 Bound $\|f_N - f_\rho\|_2$

We first handle the second term in (5.10). It is a deterministic quantity which represents the approximation error of our estimator. In the main text, we given two equivalent definitions of RKHS, respectively based on the reproducing property and the Mercer expansion. We will use the second one to explicitly calculate the approximation error. Let $\mathcal{H}$ denote the native space of K (the RKHS of interest).

Lemma 5.2.1. *Assume (A1),(A2),(A4), we have*

$$\|f_N - f_\rho\|_2 \leqslant (D\|f_\rho\|_{\mathcal{H}}\lambda_N)^{1/2} \tag{5.11}$$

where $\|\cdot\|_{\mathcal{H}}$ is the RKHS-norm. If we further assume (A3) and choose $N = \Theta(n^{\frac{d}{2\alpha+d}})$, then

$$\|f_N - f_\rho\|_2 = O(n^{-\frac{\alpha}{2\alpha+d}}) \tag{5.12}$$

Proof. Since $f \in \mathcal{H}$ by assumption, we know f_ρ has the following expansion w.r.t ψ_j: $f_\rho = \sum_{j=1}^{\infty} \theta_j \psi_j$. Recall that we defined (λ_j, ψ_j) as the eigen-system of operator $T_{k,\bar{\rho}_X}$ in Section 2. By the definition of RKHS in Proposition 2, the condition $\|f_\rho\|_{\mathcal{H}} < \infty$ in (A2) can be rewritten as:

$$\|f_\rho\|^2_{\mathcal{H}} = \sum_{j=1}^{\infty} \left(\frac{\theta_j}{\sqrt{\lambda_j}}\right)^2 < \infty \tag{5.13}$$

Define $f_{\rho,N} = \sum_{j=1}^{N} \theta_j \psi_j \in \mathcal{F}_N$ to be a truncated approximation of f_ρ (which does not depend on data). We know that $\|f_N - f_\rho\|_2$ is smaller than $\|f_{\rho,N} - f_\rho\|_2$ because f_N is the minimizer of $\|f - f_\rho\|_2$ over $f \in \mathcal{F}_N$.

So we have:

$$
\begin{aligned}
\|f_N - f_\rho\|_2 &\leq \|f_{\rho,N} - f_\rho\|_2 \\
&= \left(\int_{\mathcal{X}} \left(f_{\rho,N}(x) - f_\rho(x) \right)^2 d\rho_X(x) \right)^{1/2} \\
&\overset{(1)}{\leq} D^{1/2} \left(\int_{\mathcal{X}} (f_{\rho,N}(x) - f_\rho(x))^2 d\bar{\rho}_X(x) \right)^{1/2} \\
&\overset{(2)}{=} \left(D \sum_{j=N+1}^{\infty} \theta_j^2 \right)^{1/2} \\
&\leq \left(D\lambda_N \sum_{j=N+1}^{\infty} \theta_j^2 \lambda_j^{-1} \right)^{1/2} \\
&\leq (D\|f_\rho\|_{\mathcal{H}} \lambda_N)^{1/2}
\end{aligned}
\tag{5.14}
$$

In (1) we use assumption (A4) about the relationship between ρ_X and $\bar{\rho}_X$. In (2) we use Parseval's identity noting that ψ_j's are orthonormal w.r.t. $\bar{\rho}_X$.

If we take $N = \Theta(n^{\frac{1}{2\alpha+d}})$ and assume $\lambda_j = \Theta(j^{-2\alpha/d})$, we have $\lambda_N = \Theta(n^{-\frac{2\alpha}{2\alpha+d}})$, therefore $\|f_N - f_\rho\|_2 = O(n^{-\frac{\alpha}{2\alpha+d}})$. Thus we have proven the first part of the Lemma. □

5.2.2 Bound $\mathbb{E}\|\hat{f}_{n,N} - f_N\|_2$

In this section we bound the term associated with the stochastic error. Our proof engages the following steps: We first show the hypothesis space is a VC-class, then use this property to bound its localized Rademacher complexity. This will further lead us to the final convergence rate because $\hat{f}_{n,N}$ is an M-estimator (ERM of the negative loss) over this hypothesis space. We use the novel result presented in [48] to bound the multiplier process with a Rademacher process, which allows us to quantify the interplay between hypothesis space size and the level of noise.

Proposition 5.2.2. *Let $\mathcal{F}_N$ be the N-dimension linear space defined in* (5.8)*, then we know $\mathcal{F}_N$ is VC-subgraph class with index less than or equal to $N + 2$.*

Proof. The definition of VC-subgraph class, together with the fact that a N-dimension vector

space $\mathcal{F}_N$ of measurable functions is a VC-class of index no more than $N+2$, can be found in [123, Lemma 2.6.15] or [128, Proposition 4.20]. □

Now we use the fact that $\mathcal{F}_N$ is a VC-class to get an upper bound on its covering number. For this, we need the following result.

Proposition 5.2.3. *For a VC-subgraph class of functions $\mathcal{F}$. One has for any probability measure Q:*

$$\mathcal{N}(\epsilon\|F\|_{Q,2}, \mathcal{F}, L_Q^2) \leq CN(16e)^N \left(\frac{1}{\epsilon}\right)^{2(N-1)} \tag{5.15}$$

where N is the VC-dimension of $\mathcal{F}$ and $0 < \epsilon < 1$. And F is the envelope function of $\mathcal{F}$, i.e. $|f(x)| \leq F(x)$ for any $x \in \mathcal{X}, f \in \mathcal{F}$.

Proof. One can find the proof of a slightly more general version in [123, Theorem 2.6.7]. □

For a function space $\mathcal{F}$, define the localized uniform entropy integral as:

$$J(\delta, \mathcal{F}, L_2) := \int_0^\delta \sup_Q \sqrt{1 + \log \mathcal{N}\left(\epsilon\|F\|_{Q,2}, \mathcal{F}, L_2(Q)\right)} d\epsilon \tag{5.16}$$

Applying this to the space $\mathcal{F}_N$, we have the following result:

Lemma 5.2.4. *Let $\mathcal{F}_N$ be the function space defined in* (5.8)*, we have*

$$J(\delta, \mathcal{F}_N, L_2) \leq C_M \sqrt{N\delta^2 \log\left(\frac{1}{\delta}\right)} \tag{5.17}$$

for sufficiently small δ. The constant C_M only depends on M.

Proof. We first note $\mathcal{F}_N$ is a subset of an N-dimension vector space with envelope $F(x) = M$. By Proposition 5.2.2 and Proposition 5.2.3, we have

$$\begin{aligned}
\mathcal{N}(\epsilon M, \mathcal{F}_N, L^2(Q)) &\leq CN(16e)^N \left(\frac{1}{M\epsilon}\right)^{2N-2} \quad \text{for any measure } Q \\
\Rightarrow J(\delta, \mathcal{F}, L^2) &\leq C \int_0^\delta \sqrt{N \log\left(\frac{1}{M\epsilon}\right)} d\epsilon \quad \text{for sufficiently small } \delta \\
&\leq C\sqrt{N} \int_\infty^{\frac{1}{M\delta}} \frac{\sqrt{\log u}}{M^2 u^2} du \\
&\leq CM\delta \sqrt{N \log\left(\frac{1}{M\delta}\right)}
\end{aligned} \tag{5.18}$$

□

We can see for the linear space $\mathcal{F}_N$, the localized uniform entropy is basically $O(\sqrt{N}\delta)$ (if we omit the $\sqrt{\log(1/\delta)}$ term). When we construct the online projection estimator, the dimension of hypothesis space N increases with sample size (we can also call $\mathcal{F}_N$ a sieve). As we will see later, the local diameter $\delta = \delta_n$ we consider decreases to zero at rate $\Theta(n^{-\frac{\alpha}{2\alpha+d}})$.

We use $\epsilon_i = Y_i - g_\rho(X_i), i = 1, 2, .., n$ to denote the i.i.d zero-mean noise variables and use $e_i, i = 1, 2, .., n$ to denote n i.i.d. Rademacher variable, that is $\mathbb{P}(e_1 = 1) = \mathbb{P}(e_1 = -1) = \frac{1}{2}$.

In the following Proposition we require the noise to have a finite $\|\epsilon_i\|_{m,1}$-moment, which is defined as

$$\|\epsilon\|_{m,1} := \int_0^\infty \mathbb{P}(|\epsilon| > t)^{1/m} \mathrm{d}t \tag{5.19}$$

Let $\Delta > 0$, it is known that if ϵ_1 has a finite $m + \Delta$-th moment, then it has a finite $\|\cdot\|_{m,1}$-moment [65, Chapter 10]. So requiring having a finite $\|\cdot\|_{m,1}$, as assumed in (A1), is only slightly stronger than requiring a finite m-th moment.

Now we state and prove a proposition that connects the bounds on the multiplier/Rademacher process to the convergence rate of our M-estimator. This proposition is essentially the same as Theorem 3.4.1 in [123] and is a slight generalization of Proposition 2 in [48]. In Proposition 5.2.5, for better presentation we drop the subscript of $\mathcal{F}_N$ and simply denote it as $\mathcal{F}$. But we should keep in mind that $\mathcal{F}$ is a function space that depends on n.

Proposition 5.2.5. *Denote $\mathcal{F} - f_\rho := \{f - f_\rho \mid f \in \mathcal{F}\}$ and $\mathcal{F} - f_N := \{f - f_N \mid f \in \mathcal{F}\}$. Assume $(\mathcal{F} - f_\rho) \bigcup (\mathcal{F} - f_N)$ has an envelope function $F(x) \leq 1$. Let $X_i \overset{i.i.d.}{\sim} \rho_X$ and assume ϵ_i are i.i.d. with finite $\|\epsilon_1\|_{m,1}$-norm for some $m > 1$. Assume that for any $\delta \geq 0$, for each $f^* \in \{f_\rho, f_N\}$,*

$$\mathbb{E} \sup_{f \in \mathcal{F}: \|f - f^*\|_2 \leq \delta} \left| \frac{1}{\sqrt{n}} \sum_{i=1}^n \epsilon_i \left(f - f^*\right)(X_i) \right| = O\left(\phi_n(\delta)\right) \tag{5.20}$$

and

$$\mathbb{E} \sup_{f \in \mathcal{F}: \|f - f^*\|_2 \leq \delta} \left| \frac{1}{\sqrt{n}} \sum_{i=1}^n e_i \left(f - f^*\right)(X_i) \right| = O\left(\phi_n(\delta)\right) \tag{5.21}$$

for some ϕ_n such that $\delta \mapsto \phi_n(\delta)/\delta$ is nonincreasing. Further assume that $\|f_N - f_\rho\|_2 \leq C\delta_n$.

Then

$$\left\|\hat{f}_{n,N} - f_N\right\|_2 = O_P\left(\delta_n\right) \tag{5.22}$$

for any $\delta_n \geq n^{-\frac{1}{2}+\frac{1}{2m}}$ *such that* $\phi_n\left(\delta_n\right) \leq \sqrt{n}\delta_n^2$. *If* ϵ_1 *has a finite* m*-th moment for some* $m \geq 2$, *then:*

$$\mathbb{E}\left[\left\|\hat{f}_{n,N} - f_N\right\|_2\right] = O\left(\delta_n\right) \tag{5.23}$$

Proof. The proof is a slight generalization of Proposition 2 in [48]. The distance we are going to bound is not between $\hat{f}_{n,N}$ and f_ρ but between $\hat{f}_{n,N}$ and f_N (the population risk minimizer over $\mathcal{F}$). We first define a random process and its mean functional:

$$\begin{aligned} \mathbb{M}_n f &:= \frac{2}{n}\sum_{i=1}^{n}\left(f - f_\rho\right)\left(X_i\right)\epsilon_i - \frac{1}{n}\sum_{i=1}^{n}\left(f - f_\rho\right)^2\left(X_i\right) \\ Mf &:= \mathbb{E}\left[\mathbb{M}_n(f)\right] = -P\left(f - f_\rho\right)^2 \end{aligned} \tag{5.24}$$

We have the following property of $M(\cdot)$. For any $f \in \{f \in \mathcal{F} \mid \|f - f_N\|_2 \geq 4\|f_N - f_\rho\|_2\}$, $Mf - Mf_N \leq -\frac{1}{4}\|f - f_N\|_2^2$. For the proof of this elementary inequality, see p.337 Exercise 5 in [123], taking their $x = f, y = f_N, z = f_\rho$.

Our proof is a standard peeling argument. Let

$$\mathcal{F}_j := \left\{f \in \mathcal{F} : 2^{j-1}t\delta_n \leq \|f - f_N\|_2 < 2^j t\delta_n\right\} \tag{5.25}$$

We choose a fixed t large enough such that $t\delta_n \geq 4\|f_N - f_\rho\|_2$, we use the ERM property of $\hat{f}_{n,N}$:

$$\begin{aligned} &\mathbb{P}\left(\left\|\hat{f}_{n,N} - f_N\right\|_2 \geq t\delta_n\right) \leq \sum_{j\geq 1}\mathbb{P}\left(\sup_{f\in\mathcal{F}_j}\left(\mathbb{M}_n(f) - \mathbb{M}_n\left(f_N\right)\right) \geq 0\right) \\ &\leq \sum_{j\geq 1}\mathbb{P}\left(\sup_{f\in\mathcal{F}_j}\left(\mathbb{M}_n(f) - \mathbb{M}_n\left(f_N\right) - M(f) + M(f_N)\right) \geq 2^{2j-2}t^2\delta_n^2\right) \end{aligned} \tag{5.26}$$

We write $\left(\mathbb{M}_n(f) - \mathbb{M}_n\left(f_N\right) - M(f) + M(f_N)\right)$ explicitly:

$$\begin{aligned} &\mathbb{M}_n(f) - \mathbb{M}_n\left(f_N\right) - M(f) + M(f_N) \\ &= \frac{2}{n}\sum_{i=1}^{n}(f - f_N)(X_i)\epsilon_i + (P - \mathbb{P}_n)(f - f_\rho)^2 + (\mathbb{P}_n - P)(f_N - f_\rho)^2 \end{aligned} \tag{5.27}$$

Then we can continue the peeling argument:

$$
\begin{aligned}
&\mathbb{P}\left(\left\|\hat{f}_{n,N}-f_N\right\|_2 \geq t\delta_n\right) \\
&\leq \sum_{j\geq 1}\mathbb{P}\left(\sup_{f\in\mathcal{F}:\|f-f_N\|_2\leq 2^j t\delta_n}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(f-f_N)(X_i)\epsilon_i\right| \geq 2^{2j-5}t^2\sqrt{n}\delta_n^2\right)+ \\
&\mathbb{P}\left(\sup_{f\in\mathcal{F}:\|f-f_N\|_2\leq 2^j t\delta_n}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(f-f_\rho)^2(X_i)-\mathbb{E}(f-f_\rho)^2\right| \geq 2^{2j-4}t^2\sqrt{n}\delta_n^2\right)+ \\
&\mathbb{P}\left(\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(f_N-f_\rho)^2(X_i)-\mathbb{E}(f_N-f_\rho)^2\right| \geq 2^{2j-4}t^2\sqrt{n}\delta_n^2\right) \\
&\leq \sum_{j\geq 1}\mathbb{P}\left(\sup_{f\in\mathcal{F}:\|f-f_N\|_2\leq 2^j t\delta_n}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(f-f_N)(X_i)\epsilon_i\right| \geq 2^{2j-5}t^2\sqrt{n}\delta_n^2\right)+ \\
&2\mathbb{P}\left(\sup_{f\in\mathcal{F}:\|f-f_N\|_2\leq 2^j t\delta_n}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(f-f_\rho)^2(X_i)-\mathbb{E}(f-f_\rho)^2\right| \geq 2^{2j-4}t^2\sqrt{n}\delta_n^2\right)
\end{aligned}
\tag{5.28}
$$

The first term is the multiplier process that contains the noise variable ϵ_i's, for which we have bound (given by our assumptions). The second term can be related to the Rademacher process by standard symmetrization and contraction principles [123]. There is still a miss-match between the supremum and the random variable to be bounded, to fix this we need to use the condition $\|f_N - f_\rho\|_2 \leq C\delta_n$:

$$
\begin{aligned}
\|f-f_\rho\|_2 &\leq \|f-f_N\| + \|f_N-f_\rho\|_2 \\
&\leq \|f-f_N\| + C\delta_n \\
\Rightarrow \{f\in\mathcal{F}:\|f-f_N\|\leq 2^j t\delta_n\} &\subset \{f\in\mathcal{F}:\|f-f_\rho\|_2\leq (2^j t+C)\delta_n\}
\end{aligned}
\tag{5.29}
$$

Therefore the second term is bounded by

$$
2\mathbb{P}\left(\sup_{f\in\mathcal{F}:\|f-f_\rho\|_2\leq (2^j t+C)\delta_n}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(f-f_\rho)^2(X_i)-\mathbb{E}(f-f_\rho)^2\right| \geq 2^{2j-4}t^2\sqrt{n}\delta_n^2\right) \tag{5.30}
$$

And the rest of the proof is the same as Proposition 2 in [48]. □

When ϵ_i is sub-Gaussian noise (note that sub-Gaussian/sub-exponential random variables have finite moments of all orders), the bound on the empirical process terms (5.20) and (5.21)

usually only depend on the entropy of $\mathcal{F}_N$: Thus the convergence rate will only depend on the entropy as well. However if we only assume moment conditions, then $\phi_n(\delta)$ will depend on both the entropy *and* the moment order [48, Lemma 9]: Thus the convergence rate would depend on both as well when m is not large enough.

Now we state the following Lemma to complete our bound of $\mathbb{E}\|\hat{f}_{n,N} - f_N\|_2$. Its proof is postponed to after we conclude the main result.

Lemma 5.2.6. *Assume (A1) and $\hat{f}_{n,N} \in \mathcal{F}_N$ defined in (5.8). We select $N = \Theta\left(n^{\frac{d}{2\alpha+d}}\right)$. (Recall that α is the smoothness parameter, d is the dimension of X_i and m is the moment index of ϵ_i)*

Then with $\delta_n = \Theta\left(n^{-\frac{\alpha}{2\alpha+d}} \vee n^{-\frac{1}{2}+\frac{1}{2m}}\right)$, for each $f^* \in \{f_N, f_\rho\}$ we have

$$\begin{aligned}\mathbb{E}\sup_{f\in\mathcal{F}_N:\|f-f^*\|_2\le\delta_n}\left|\sum_{i=1}^n \epsilon_i\left(f-f^*\right)(X_i)\right| \vee \mathbb{E}\sup_{f\in\mathcal{F}_N:\|f-f^*\|_2\le\delta_n}\left|\sum_{i=1}^n e_i\left(f-f^*\right)(X_i)\right| \\ \le C_\alpha\begin{cases} n^{\frac{d}{2\alpha+d}}\sqrt{\log n}\left(1\vee\|\epsilon_1\|_{2\alpha+1,1}\right), & m\ge 2\alpha/d+1 \\ n^{\frac{1}{m}}\sqrt{\log n}\left(1\vee\|\epsilon_1\|_{m,1}\right), & 1\le m<2\alpha/d+1\end{cases}\end{aligned} \tag{5.31}$$

where $\|\epsilon_1\|_{2\alpha+1}$ is the $2\alpha+1$-th moment of ϵ_1.

In light of Proposition 5.2.5, (5.31) can be written as

$$\begin{aligned}\mathbb{E}\sup_{f\in\mathcal{F}_N:\|f-f^*\|_2\le\delta_n}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^n \epsilon_i\left(f-f^*\right)(X_i)\right| \\ \vee\ \mathbb{E}\sup_{f\in\mathcal{F}_N:\|f-f^*\|_2\le\delta_n}\left|\frac{1}{\sqrt{n}}\sum_{i=1}^n e_i\left(f-f^*\right)(X_i)\right| \le \phi_n(\delta_n)\end{aligned} \tag{5.32}$$

where

$$\phi_n(\delta) = \begin{cases} C_\alpha\sqrt{\log n/n}\delta^{-1/\alpha}\left(1\vee\|\epsilon_1\|_{1+2\alpha,1}\right), & m\ge 1+2\alpha \\ C_\alpha\sqrt{\log n/n}\delta^{-2/(m-1)}\left(1\vee\|\epsilon_1\|_{m,1}\right), & 1\le m<1+2\alpha\end{cases} \tag{5.33}$$

Lemma 5.2.7. *Assume (A1) and $\hat{f}_{n,N} \in \mathcal{F}_N$. Choosing $N = \Theta(n^{\frac{d}{2\alpha+d}})$,*

$$E[\|\hat{f}_{n,N} - f_N\|_2] = O\left(n^{-\frac{\alpha}{2\alpha+d}}\sqrt{\log n} \vee n^{-\frac{1}{2}+\frac{1}{2m}}\sqrt{\log n}\right) \tag{5.34}$$

Proof. We use the result of Lemma 5.2.6 as conditions of Proposition 5.2.5, and then identfy the smallest δ_n satisfying $\phi_n(\delta_n) \leq \sqrt{n}\delta_n^2$, which will give the stated convergence rate. □

Proof of Theorem 3. We need only combine the bounds in Lemma 5.2.1 and Lemma 5.2.7 using the triangle inequality. □

We now return to proving Lemma 5.2.6. We first state two results, Propositions 5.2.8, and 5.2.9, from the literature which we will use to prove our Lemma. We begin with a standard result connecting Rademacher complexity and the entropy integral.

Proposition 5.2.8 (Theorem 2.1, [122])**.** *Suppose that $\mathcal{G}$ has a finite envelope $G(x) \leq 1$ and $X_1, \ldots, X_n$'s are i.i.d. random variables with law P.*

Then with $\mathcal{G}(\delta) := \{g \in \mathcal{G} : Pg^2 < \delta^2\}$,

$$\mathbb{E} \sup_{g \in \mathcal{G}(\delta)} \left| \frac{1}{\sqrt{n}} \sum_{i=1}^{n} e_i g\left(X_i\right) \right| = O\left(J\left(\delta, \mathcal{G}, L_2\right) \left(1 + \frac{J\left(\delta, \mathcal{G}, L_2\right)}{\sqrt{n}\delta^2 \|G\|_{P,2}}\right) \|G\|_{P,2} \right) \tag{5.35}$$

We next give a recent inequality established in [48]. This allows us to relax common subgaussian assumptions to only moment conditions on the ϵ_i's.

Proposition 5.2.9 (Theorem 1,[48])**.** *Suppose X_i's, ϵ_i's are all i.i.d. random variables and X_i's are independent of ϵ_i's. Let $\{\mathcal{G}_k\}_{k=1}^n$ be a sequence of function classes such that $\mathcal{G}_k \supset \mathcal{G}_n$ for any $1 \leq k \leq n$. Assume further that there exists a nondecreasing concave function $\psi_n : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ with $\psi_n(0) = 0$ such that*

$$\mathbb{E} \sup_{f \in \mathcal{G}_k} \left| \sum_{i=1}^{k} e_i f\left(X_i\right) \right| \leq \psi_n(k) \tag{5.36}$$

holds for all $1 \leq k \leq n$. Then

$$\mathbb{E} \sup_{f \in \mathcal{G}_n} \left| \sum_{i=1}^{n} \epsilon_i f\left(X_i\right) \right| \leq 4 \int_0^\infty \psi_n \left(\sum_{i=1}^{n} \mathbb{P}\left(|\epsilon_i| > t\right) \right) \mathrm{d}t \tag{5.37}$$

With these two results in hand, we are now ready to prove Lemma 5.2.6.

Proof of Lemma 5.2.6. We need to show the result for both $f^* = f_N$ and $f^* = f_\rho$. We will explicitly show the result for $f^* = f_N$: The proof in the case $f^* = f_\rho$ is exactly the same.

Denote

$$\mathcal{F}_N(\delta_k) := \{f \in \mathcal{F}_N \mid \|f - f_N\|_2^2 \leq \delta_k^2\} \tag{5.38}$$

We first combine Proposition 5.2.8 with the entropy bound we established in Lemma 5.2.4 to derive

$$\mathbb{E} \sup_{f \in \mathcal{F}_N(\delta_k)} \left| \sum_{i=1}^{k} e_i f(X_i) \right| \leq C \delta_k k^{\frac{d}{2(2\alpha+d)} + \frac{1}{2}} \sqrt{\log k} \tag{5.39}$$

where $\delta_k = k^{-\frac{\alpha}{2\alpha+d}} \vee k^{-\frac{1}{2}+\frac{1}{2m}}$.

When $m \geq 2\alpha/d + 1$ (recall m is the moment index for ϵ_i's), $k^{-\frac{\alpha}{2\alpha+d}} > k^{-\frac{1}{2}+\frac{1}{2m}}$, so the above bound becomes

$$\mathbb{E} \sup_{f \in \mathcal{F}_N(\delta_k)} \left| \sum_{i=1}^{k} e_i f(X_i) \right| \leq C k^{\frac{d}{2\alpha+d}} \sqrt{\log k} \tag{5.40}$$

Using (5.40) we see that the conditions of Proposition 5.2.9 are satisfied, thus giving us

$$\begin{aligned} \mathbb{E} \sup_{f \in \mathcal{F}_N(\delta_k)} \left| \sum_{i=1}^{n} \epsilon_i f(X_i) \right| &\leq C \int_0^\infty \left(\sum_{i=1}^{n} \mathbb{P}(|\epsilon_i| > t) \right)^{\frac{d}{2\alpha+d}} \sqrt{\log \left(\sum_{i=1}^{n} \mathbb{P}(|\epsilon_i| > t) \right)} \mathrm{d}t \\ &= C n^{\frac{d}{2\alpha+d}} \sqrt{\log n} (1 \vee \|\epsilon_1\|_{2\alpha+1,1}) \end{aligned} \tag{5.41}$$

Note that we used ϵ_i's are i.i.d. random variables.

When $1 < m < 2\alpha/d + 1$, (5.39) becomes

$$\mathbb{E} \sup_{f \in \mathcal{F}_N(\delta_k)} \left| \sum_{i=1}^{k} e_i f(X_i) \right| \leq C k^{\frac{1}{m}} \sqrt{\log k}. \tag{5.42}$$

Plugging this in to Proposition 5.2.9 we get

$$\mathbb{E} \sup_{f \in \mathcal{F}_N(\delta_k)} \left| \sum_{i=1}^{n} \epsilon_i f(X_i) \right| \leq C n^{\frac{1}{m}} \sqrt{\log n} (1 \vee \|\epsilon_1\|_{m,1}) \tag{5.43}$$

This completes the proof. □

5.3 Online Projection Estimator and Functional Stochastic Gradient Descent

The computational expense of **Algorithm 2** is a dramatic improvement compared with SGD based algorithms, whose expense is $O(n)$ per updating. We also note that the computational expense of **Algorithm 2** depends on our assumption of the spectrum of operator T_K. The larger α is, the stronger our statistical assumption is, the faster our algorithm is. However, the expense of SGD-based algorithm is not sensitive to the statistical assumptions.

In this section we use the same notation as in Section 3 in the main text. We define $\hat{\boldsymbol{\theta}}_{N,n}$ as the minimizer of the empirical loss

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^N} \sum_{i=1}^{n} (Y_i - \boldsymbol{\theta}^\top \boldsymbol{\psi}^N(X_i))^2 \tag{5.44}$$

Here we use double subscript to emphasize that $\hat{\boldsymbol{\theta}}_{N,n}$ is calculated with N basis function and n data. Similarly, we can define $\hat{\boldsymbol{\theta}}_{N,n-1}$ as the minimizer when there is one less sample (X_n, Y_n) (but keep the other samples the same). There is actually a recursive relationship between $\hat{\boldsymbol{\theta}}_{N,n}$ and $\hat{\boldsymbol{\theta}}_{N,n-1}$:

$$\hat{\boldsymbol{\theta}}_{N,n} = \hat{\boldsymbol{\theta}}_{N,n-1} + \Phi_n \boldsymbol{\psi}_n \left[Y_n - \hat{f}_{n-1,N}(X_n) \right] \tag{5.45}$$

See [73] p.18-20 for the derivation. This formula tells us how $\hat{\boldsymbol{\theta}}_{N,n}$ changes when one additional data-pair is observed. If we see $\hat{\boldsymbol{\theta}}_{N,n}$ as an update of $\hat{\boldsymbol{\theta}}_{N,n-1}$ with (X_n, Y_n), the step size will scale in proportion to the prediction error $|Y_n - \hat{f}_{n-1,N}(X_n)|$, and the direction is $\Phi_n \boldsymbol{\psi}_n$ (which, in general, is not equal to $\boldsymbol{\psi}_n$)

Similarly, we can derive a recursive relationship for how $\hat{\boldsymbol{\theta}}_{N,n}$ changes when one more basis function ψ_{N+1} is added in. Specifically,

$$\hat{\boldsymbol{\theta}}_{N+1,n} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_{N,n} \\ 0 \end{bmatrix} + \frac{\left(\boldsymbol{\psi}^{N+1}\right)^\top \boldsymbol{\Delta}_n}{\left\| \left(I - P_n\right) \boldsymbol{\psi}^{N+1} \right\|^2} \begin{bmatrix} -P_n \boldsymbol{\psi}^{N+1} \\ 1 \end{bmatrix} \tag{5.46}$$

Where $\boldsymbol{\Delta}_n$ is the residual vector, whose i-th component is defined by:

$$\boldsymbol{\Delta}_n^{(i)} = Y_i - \hat{f}_{n,N}(X_i) \tag{5.47}$$

and $P_n = (\Psi_n^\top \Psi_n)^{-1} \Psi_n^\top$ is the projection matrix of the column space of design matrix Ψ_n with N features. We give the derivation in the later part of this section.

The influence of a new feature on the regression coefficients is quantitatively associated with how much the residual can be explained by the new feature (represented by the term $\left(\boldsymbol{\psi}^{N+1}\right)^\top \boldsymbol{\Delta}_n$) and how orthogonal the new feature is to the old features (represented by $P_n \boldsymbol{\psi}^{N+1}$).

However, if we use parametric stochastic gradient descent to solve the problem (2.15), then the updating rule should be:

$$\hat{\boldsymbol{\theta}}_{N,n} = \hat{\boldsymbol{\theta}}_{N,n-1} + \epsilon_n \boldsymbol{\psi}_n \left[Y_n - \hat{f}_{n-1,N}(X_n) \right] \tag{5.48}$$

where we usually choose $\epsilon_n \asymp \frac{1}{n}$.

Comparing (5.48) with (5.45), we see that it replaces the structured matrix Φ_n with a diagonal matrix $\epsilon_n I$. By doing so it omits the information of the correlation between features, this can help to illustrate why the SGD-based estimator (5.48) usually has a larger generalization error than the empirical risk minimizer (5.45).

5.3.1 Proof of recursive formula (5.46)

Proof. In this proof, we use a double subscript to indicate the dimension of the matrices. By definition of OLS estimator:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{N+1,n} &= \Phi_{(N+1)\times(N+1)} \cdot \Psi^{\top}_{n\times(N+1)} \cdot \boldsymbol{Y}_n \\
&= \Phi_{(N+1)\times(N+1)} \cdot \left(\sum_{i=1}^{n} Y_i \left[\psi_1 \left(X_i \right), \ldots, \psi_{N+1} \left(X_i \right) \right]^{\top} \right) \\
&= \Phi_{(N+1)\times(N+1)} \cdot \begin{bmatrix} \sum_{i=1}^{n} \boldsymbol{\psi}_N \left(X_i \right) Y_i \\ \sum_{i=1}^{n} \psi_{N+1} \left(X_i \right) Y_i \end{bmatrix} \\
&\overset{(1)}{=} \Phi_{(N+1)\times(N+1)} \cdot \begin{bmatrix} \Phi^{-1}_{N\times N} \cdot \hat{\boldsymbol{\theta}}_{N,n} \\ \sum_{i=1}^{n} \psi_{N+1} \left(X_i \right) Y_i \end{bmatrix} \\
&\overset{(2)}{=} \left(\begin{bmatrix} \Phi_{N\times N} & 0 \\ 0 & 0 \end{bmatrix} + A \right) \cdot \begin{bmatrix} \Phi^{-1}_{N\times N} \cdot \hat{\boldsymbol{\theta}}_{N,n} \\ \sum_{i=1}^{n} \psi_{N+1} \left(X_i \right) Y_i \end{bmatrix}
\end{aligned}$$

where

$$A = \begin{bmatrix} \frac{1}{k} \Phi_{n-1} \, \boldsymbol{b}\boldsymbol{b}^T \, \Phi_{n-1} & -\frac{1}{k} \Phi_{n-1} \, \boldsymbol{b} \\ -\frac{1}{k} \boldsymbol{b}^T \, \Phi_{n-1} & \frac{1}{k} \end{bmatrix}$$

$$\boldsymbol{b} = \Psi^T_{n-1} \boldsymbol{\psi}_{N+1}$$

$$k = \boldsymbol{\psi}^T_{N+1} \boldsymbol{\psi}_{N+1} - \boldsymbol{b}^T \Phi_{n-1} \, \boldsymbol{b}$$

In (1) we use the definition of $\hat{\boldsymbol{\theta}}_{N,n}$ and in (2) use the block matrix inversion formula.

$$\hat{\boldsymbol{\theta}}_{N+1,n} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_{N,n} \\ 0 \end{bmatrix} + \frac{1}{k} \cdot \begin{bmatrix} \Phi_{N\times N} \boldsymbol{b} \left(\boldsymbol{b}^T \hat{\boldsymbol{\theta}}_{N,n} - \sum_{i=1}^{n} \psi_{N+1} \left(X_i \right) Y_i \right) \\ \left(\sum_{i=1}^{n} \psi_{N+1} \left(X_i \right) Y_i - \boldsymbol{b}^{\top} \hat{\boldsymbol{\theta}}_{N,n} \right) \end{bmatrix} \tag{5.49}$$

Note that

$$\boldsymbol{b}^{\top} \hat{\boldsymbol{\theta}}_{N,n} = \sum_{i=1}^{n} \psi_{N+1} \left(X_i \right) \sum_{j=1}^{N} \psi_j(X_i) \hat{\boldsymbol{\theta}}^{(j)}_{N,n} = \sum_{i=1}^{n} \psi_{N+1} \left(X_i \right) \hat{f}_{n,N}(X_i) \tag{5.50}$$

So

$$\sum_{i=1}^{n} \psi_{N+1}(X_i) Y_i - \boldsymbol{b}^\top \hat{\boldsymbol{\theta}}_{N,n} = \sum_{i=1}^{n} \psi_{N+1}(X_i) (Y_i - f_{n,N}(X_i)) \tag{5.51}$$

Continuing, we see that

$$\hat{\boldsymbol{\theta}}_{N+1,n} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_{N,n} \\ 0 \end{bmatrix} + \frac{\boldsymbol{\psi}_{N+1}^\top \boldsymbol{\Delta}_n}{k} \cdot \begin{bmatrix} -\Phi_{N\times N}\boldsymbol{b} \\ 1 \end{bmatrix}$$

Now we expand k:

$$\begin{aligned} k &= \boldsymbol{\psi}_{N+1}^\top \boldsymbol{\psi}_{N+1} - \boldsymbol{\psi}_{N+1}^\top \Psi_{n\times N} \Phi_{N\times N} \Psi_{n\times N}^\top \boldsymbol{\psi}_{N+1} \\ &= \boldsymbol{\psi}_{N+1}^\top \left(I - \Psi_{n\times N} \left(\Psi_{n\times N}^\top \Psi_{n\times N} \right)^{-1} \Psi_{n\times N} \right) \boldsymbol{\psi}_{N+1} \\ &= \left\| (I - P_n) \boldsymbol{\psi}_{N+1} \right\|^2 \end{aligned}$$

And use the definition of b:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{N+1,n} = &\begin{bmatrix} \hat{\boldsymbol{\theta}}_{N,n} \\ 0 \end{bmatrix} \\ &+ \frac{\boldsymbol{\psi}_{N+1}^\top \boldsymbol{\Delta}_n}{\left\| (I - P_n) \boldsymbol{\psi}_{N+1} \right\|^2} \begin{bmatrix} -\Phi_{N\times N} \begin{bmatrix} \boldsymbol{\psi}_1^\top \boldsymbol{\psi}_{N+1} \\ \vdots \\ \boldsymbol{\psi}_N^\top \boldsymbol{\psi}_{N+1} \end{bmatrix} \\ 1 \end{bmatrix} \\ = &\begin{bmatrix} \hat{\boldsymbol{\theta}}_{N,n} \\ 0 \end{bmatrix} + \frac{\boldsymbol{\psi}_{N+1}^\top \boldsymbol{\Delta}_n}{\left\| (I - P_n) \boldsymbol{\psi}_{N+1} \right\|^2} \begin{bmatrix} -P_n \boldsymbol{\psi}_{N+1} \\ 1 \end{bmatrix} \end{aligned} \tag{5.52}$$

□

5.4 Regression in Additive Models

In the main text we discussed estimation in multivariate RKHS and how it suffers from the curse of dimensionality. For $X_i \in \mathbb{R}^d$, it is also quite common to impose an extra additive structure on the model, in other words, we assume

$$f_\rho(x_i) = \sum_{k=1}^{d} f_{\rho,k}\left(x_i^{(k)} \right) \tag{5.53}$$

where the component functions $f_{\rho,i}$ belong to a RKHS $\mathcal{H}$ (in general they can belong to different spaces), and $x_i^{(k)}$ is the k-th entry of x_i. Such a model is a generalization of the multivariate linear model. It balances modeling flexibility with tractability of estimation. See eg. [52] and [141] for further discussion.

The projection estimator for an additive model is obtained by solving the following least-squares problem in Euclidean space (which is essentially the same as solving the problem (2.15)).

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{N\times d}} \sum_{i=1}^{n} (Y_i - \sum_{k=1}^{d}\sum_{j=1}^{N} \theta_{jk}\psi_j(x_i^{(k)}))^2 \tag{5.54}$$

here N still needs to be chosen of order $n^{\frac{1}{2\alpha+1}}$, when $\lambda_j = \Theta(j^{-2\alpha})$. The online projection estimator in an additive model is

$$\hat{f}_{n,N} = \sum_{k=1}^{d}\sum_{j=1}^{N} \hat{\theta}_{jk}\psi_j \tag{5.55}$$

For a fixed d, the minimax rate for estimating an additive model is identical (losing a constant d) to the minimax rate in the analogous one-dimension nonparametric regression problem working with the same hypothesis space $\mathcal{H}$ [92].

The design matrix of (5.54) now is of dimension $n \times (Nd)$. When a new data point is collected, our design matrix grows by one row. When we need to increase the model capacity however, we need to add one feature for each dimension (in total d columns). Updating such estimators when $X_i \in \mathbb{R}^d$ has a computational expense of order $O(d^2 n^{\frac{2}{2\alpha+1}})$, by a argument similar to that presented in Section 3.4. To clarify, in Section 3.4 we are assuming the eigenvalue $\lambda_j = \Theta(j^{-2\alpha/d})$ (for example, the RKHS is d-dimension, α-th order Sobolev space); however in this section we are discussing d-dimension additive model, *each component* lies in a 1-dimension RKHS whose $\lambda_j = \Theta(j^{-2\alpha})$. The additive model is more restrictive, therefore we have better statistical and computational guarantee when the model is well-specified.

5.4.1 Additive Model Application

We chose a 10-dimension additive function to illustrate the efficacy of our method for fitting additive models. In this example, the components of the f_ρ in each dimension are Doppler-like functions. For $x \in \mathbb{R}^{10}$,

$$\begin{aligned} f_\rho(x) &= \sum_{k=1}^{10} f_{\rho,k}(x^{(k)}) \\ &= \sum_{k=1}^{10} \left\{ \sin\left(\frac{2\pi}{(x^{(k)}+0.1)^{k/20}} \right) - \sin\left(\frac{2\pi}{0.1^{k/20}} \right) \right\} \end{aligned} \tag{5.56}$$

Similar functions are used in [97]. The kernel (for each dimension) we consider is

$$K(s,t) = \sum_{m=1}^{2} s^m t^m + B_4(\{s-t\}) \tag{5.57}$$

In Figure 5.1, we compare the method in this chapter with the additive smoothing spline estimator calculated with back fitting using R package 'gam' [51]. Both of the methods achieve rate-optimal convergence, but we note the smoothing spline method takes dramatically more time as an offline estimator.

5.5 Details of simulation studies

In the main text we gave important details on of the settings of our simulation studies. To help our readers replicate our result, we now list all details for our simulations.

5.5.1 Notation and general setting

The $\|\hat{f}_{n,N} - f_\rho\|_2^2$ on the y-axis of Figure 2 is estimated with 1,000 X generated from ρ_X. The estimator based on kernel ridge regression (KRR) is defined as the minimizer of penalized mean-square error

$$\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - f\left(X_i\right) \right)^2 + \lambda_{n,KRR} \|f\|_{\mathcal{H}}^2 \tag{5.58}$$

for a closed form solution and theoretical optimal selection of $\lambda_{n,KRR}$, see 12.5.2 and Theorem 13.7 of [128].

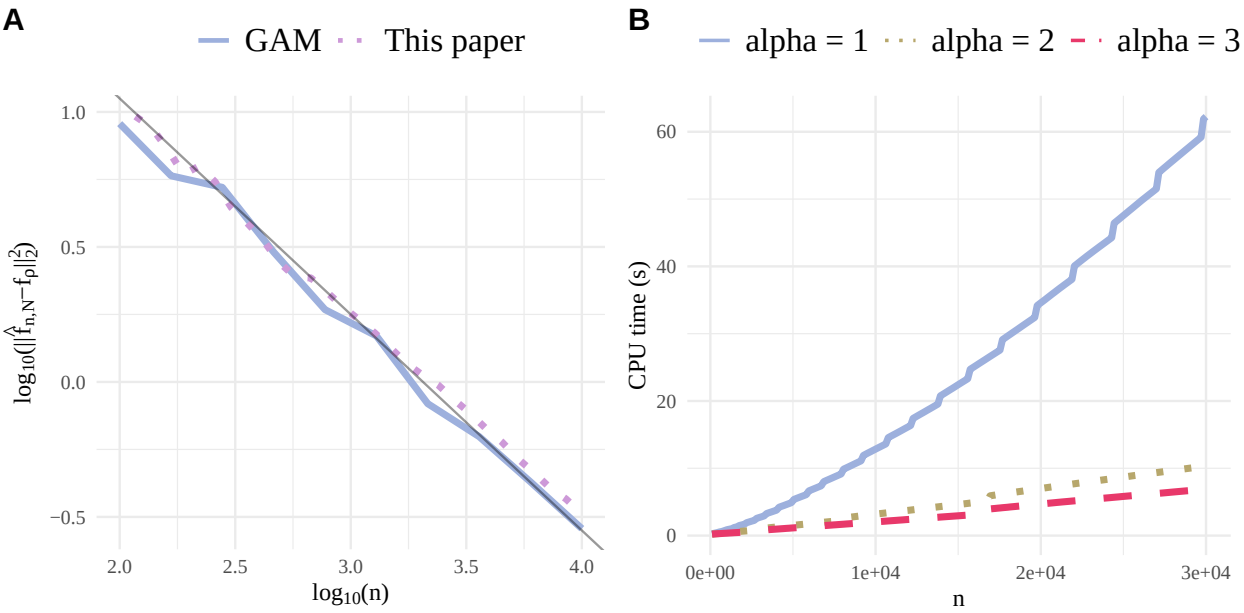

Figure 5.1: Additive model: generalization error and CPU time. **(A)** Both smoothing spline and online projection estimator achieve the optimal rate $O(n^{-4/5})$. The black line has slope $-4/5$. Each curve is based on 15 independent runs. **(B)** The CPU time decreases as α becomes larger (repetitions=10).

In the main text, we slightly simplify the update rule for nonparametric SGD estimator without losing the essential principles. In all the simulation study of this section, the SGD estimator we use is the version with Polyak averaging (p.1375-1376 of [25])

$$\tilde{f}_n = \tilde{f}_{n-1} + \gamma_{n,SGD}\left[Y_n - \tilde{f}_{n-1}\left(X_n\right)\right] K_{X_n} \tag{5.59}$$

$$\hat{f}_n = \frac{1}{n+1}\sum_{k=0}^{n} \tilde{f}_k \tag{5.60}$$

The nonparametric SGD estimator we use is $\hat{f}_n$. To update such an estimator, the computational cost is also $O(n)$.

All the simulation study examples are coded in R version 3.5.1.

5.5.2 One Dimension Example Settings

We give the details of example 1 (resp. example 2) in Table 5.1 (resp. Table 5.2).

Table 5.1: Settings of example 1. See [126] and [25]

f_ρ	$B_4(x) = x^4 - 2x^3 + x^2 - \frac{1}{30}$
ϵ	Unif([-0.02,0.02])
$p_X(x)$	$1_{[0,1]}(x)$
$K(s,t)$	$\frac{-1}{24}B_4(\{s-t\}) = \sum_{j=1}^{\infty} \frac{2}{(2\pi j)^4}[\cos(2\pi js)\cos(2\pi jt)$ $+ \sin(2\pi js)\sin(2\pi jt)]$
RKHS $\mathcal{H}$	$W_2^{per} = \Big\{ f \in L^2([0,1]) \mid \int_0^1 f(u)du = 0,$ $f(0) = f(1), f'(0) = f'(1), \int_0^1 \left(f^{(2)}(u)\right)^2 du < \infty \Big\}$
λ_j	$\frac{2}{(2\pi j)^4} = O(j^{-4})$
$\psi_j(x)$	$\sin(2\pi jx)$ and $\cos(2\pi jx)$
basis adding step	$n = \lfloor 0.2N^5 \rfloor$
Hyperparameter KRR $\lambda_{n,KRR}$	$\lambda_{n,KRR} = 10^{-3}n^{-4/5}$
Learning rate $\gamma_{n,SGD}$	$\gamma_{n,SGD} = 128n^{-0.5}$

5.5.3 Additive Model Example

We use the function gam() in R package **gam** [51] to fit the additive model with smoothing spline. The degrees of freedom parameter used in the `s()` function were selected to increase with n. The details for the additive model example (including parameter selection) are given in Table 5.3.

Table 5.2: Settings of example 2. See [128, Chap. 12] for more discussion on the kernel space W_1^0.

f_ρ	$(6x-3)\sin(12x-6)+\cos^2(12x-6)$
ϵ	Normal(0,5)
$p_\rho(x)$	$(x+0.5)1_{[0,1]}(x)$
$K(s,t)$	$\min\{s,t\}=\sum_{j=1}^{\infty}\frac{8}{(2j-1)^2\pi^2}\sin\left(\frac{(2j-1)\pi s}{2}\right)\sin\left(\frac{(2j-1)\pi t}{2}\right)$
RKHS $\mathcal{H}$	$W_1^0=\left\{f\in L^2([0,1])\vert f(0)=0,\int_0^1(f'(u))^2du<\infty\right\}$
λ_j	$\frac{2}{(2j-1)^2\pi^2}=O(j^{-2})$
$\psi_j(x)$	$2\sin\left(\frac{(2j-1)\pi x}{2}\right)$
basis adding step	$n=\lfloor 0.5N^3\rfloor$
Hyperparameter KRR $\lambda_{n,KRR}$	$\lambda_{n,KRR}=0.1n^{-2/3}$
Learning rate $\gamma_{n,SGD}$	$\gamma_{n,SGD}=5n^{-0.5}$

Table 5.3: Settings of Additive model example.

f_ρ	$\sum_{k=1}^{10}\left\{\sin\left(\frac{2\pi}{(X^{(k)}+0.1)^{k/20}}\right)-\sin\left(\frac{2\pi}{0.1^{k/20}}\right)\right\}$
ϵ	Normal(0,5)
$p_\rho(X^{(1)},...,X^{(10)})$	$\Pi_{k=1}^{10}1_{[0,1]}(X^{(k)})$
$K(s,t)$ (for each dimension)	$\sum_{m=1}^{2}s^mt^m+B_4(\{s-t\})$
RKHS $\mathcal{H}$	$W_2=\left\{f\in L^2([0,1])\mid\int_0^1(f''(u))^2du<\infty\right\}$
λ_j	$\frac{2}{(2\pi j)^4}=O\left(j^{-4}\right)$
$\psi_j(x)$	$x,x^2,\sin(2\pi jx),\cos(2\pi jx)$
basis adding step	$n=\lfloor 0.2N^5\rfloor$
df for smoothing spline	$2\lfloor n^{1/5}\rfloor$

5.6 A Note for Application and Additional Examples

The hypothesis spaces used so far in this chapter have been well-studied in previous work, and are relatively easy to engage with: Their kernel functions have a closed form, and their eigenfunctions can also be explicitly written out with respect to some special measures $\bar{\rho}$.

However, they are usually equipped with some undesirable boundary conditions. For example, in example 2, it is more interesting to consider the space

$$W_1 = \left\{ f \in L^2([0,1]) | \int_0^1 \left(f'(u)\right)^2 du < \infty \right\} \tag{5.61}$$

rather than the one we use in our simulation study

$$W_1^0 = \left\{ f \in L^2([0,1]) | f(0) = 0, \int_0^1 \left(f'(u)\right)^2 du < \infty \right\} \tag{5.62}$$

Although it is known that W_1 is also an RKHS [128] with kernel $\tilde{K}(s,t) = 1 + \min\{s,t\}$, it takes extra analytical work to get the form of eigenfunctions for $\tilde{K}$.

For practical purposes, it is enough to consider functions of the following form as estimator:

$$\hat{f}_{n,N}(x) = \theta_0 \cdot 1 + \sum_{j=1}^{N} \theta_j \psi_j(x) \tag{5.63}$$

where $\psi_j = \sqrt{2}\sin\left(\frac{(2j-1)\pi x}{2}\right)$ as stated in Table 1. Because the difference between W_1^0 and W_1 is merely a constant function in the sense that

$$W_1 = \{1\} \oplus W_1^0 \tag{5.64}$$

When a new sample comes in, we update $\hat{f}_{n,N}$ (and potentially add a new basis function) in an online manner as in Algorithm 2. Similarly, in example 1, the more interesting space is

$$W_2 = \left\{ f \in L^2([0,1]) | \int_0^1 \left(f^{(2)}(u)\right)^2 du < \infty \right\} \tag{5.65}$$

Note that

$$W_2 = \{1\} \oplus \{x\} \oplus \{x^2\} \oplus W_2^{per} \tag{5.66}$$

So the projection estimator can be of the form

$$\hat{f}_{n,N}(x) = \sum_{k=0}^{2} \tilde{\theta}_k x^k + \sum_{j=1}^{N} \theta_j \psi_j(x) \tag{5.67}$$

where ψ_j's are the trigonometric functions listed in Table 1.

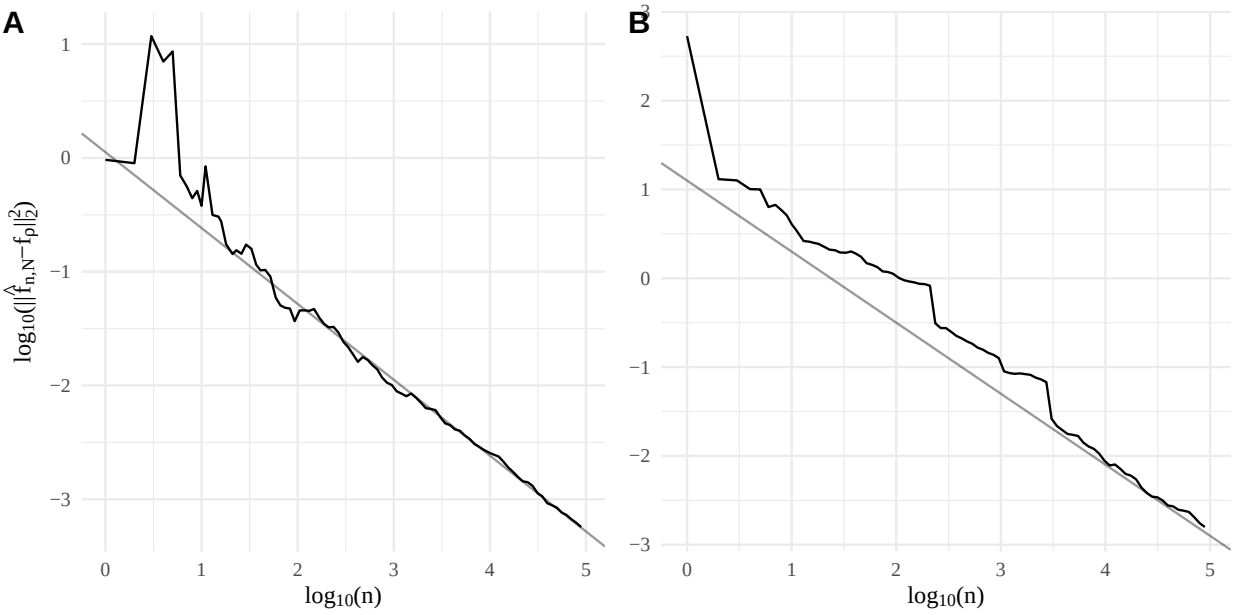

Figure 5.2: Generalization error for additional examples.**(A)** Example A.1, black line has slope $-2/3$ **(B)** Example A.2, the black line has slope $-4/5$. Both estimators achieve the minimax rates in W_1 and W_2. Each curve is based on 15 independent repetitions.

The settings for our two additional examples are given in Table 5.4

Table 5.4: Settings of additional examples.

	Example A.1	Example A.2
f_ρ	$1+(x-0.5)1_{[0.5,1]}(x)$ $+2(x-0.2)1_{[0.2,1]}(x)$	$1+(6x-3)\sin(12x-6)+\cos^2(12x-6)$ $+10(x-0.5)^2 1_{[0.5,1]}(x)$
ϵ	Normal(0,1)	Unif(-5,5)
$p_\rho(x)$	$(x+0.5)1_{[0,1]}(x)$	$1_{[0,1]}(x)$
RKHS	W_1	W_2
basis function	$1, \sin\left(\frac{(2j-1)\pi x}{2}\right), j=1,2,...$	$1, x, x^2, \sin(2\pi jx), \cos(2\pi jx), j=1,2,...$
basis adding step	$n=\lfloor 0.5N^3 \rfloor$	$n=\lfloor \frac{1}{30}N^5 \rfloor$

Chapter 6

SUPPLEMENTARY MATERIALS FOR CHAPTER 3

6.1 Algorithm of Sieve-SGD, Numerical Version

In the main text, section 3.5.1, we presented a functional form of the proposed Sieve-SGD algorithm. To facilitate comprehension and application, we also attach an equivalent numerical version of it.

Proposed Algorithm: Sieve Stochastic Gradient Descent (Sieve-SGD)

Set $\alpha, \omega > 0$, step size $\{\gamma_i\}$ and basis functions $\{\psi_j\}$.

Initialize $\bar{\beta}_j, \hat{\beta}_j = 0$ for all $j \in \mathbb{N}^+$.

For $i = 1, 2, ...$:

1. Calculate $J_i = \lfloor i^\alpha \rfloor$, collect data pair (X_i, Y_i).

2. Update $\hat{\beta}_j$

$$\text{res}_i \leftarrow Y_i - \sum_{j=1}^{J_{i-1}} \hat{\beta}_j \psi_j(X_i) \tag{6.1}$$

For $j = 1, ..., J_i$:

$$\hat{\beta}_j \leftarrow \hat{\beta}_j + \gamma_i \text{res}_i \left(j^{-2\omega} \psi_j(X_i) \right) \tag{6.2}$$

3. Update $\bar{\beta}_j$

For $j = 1, ..., J_i$:

$$\bar{\beta}_j \leftarrow \frac{i}{i+1}\bar{\beta}_j + \frac{1}{i+1}\hat{\beta}_j \tag{6.3}$$

6.2 Multivariate Regression Problems

In this section, we will give additional discussion of the technical details for multivariate regression using the Sieve-SGD estimator.

6.2.1 Hyperbolic cross and Sieve-SGD

In main text Section 3.5.4, we discussed using a tensor product sieve-basis to solve multivariate feature problems. Assume $\mathbf{X} \in \mathbb{R}^p$ and $Y \in \mathbb{R}$. It is known that for an univariate orthonormal basis $\{\psi_j, j \in \mathbb{N}^+\}$, the set of tensor product functions $\{\psi_{\mathbf{j}}(\mathbf{x}) = \prod_{k=1}^p \psi_{\mathbf{j}^{(k)}}(\mathbf{x}^{(k)}), \mathbf{j} \in (\mathbb{N}^+)^p\}$ is also an orthonormal basis ($\mathbf{v}^{(k)}$ is the k-th component of $\mathbf{v} \in \mathbb{R}^p$). However, there are more choices of the *order* in which we arrange the tensor product basis functions when estimating an unknown regression function. We propose using the index product $c_{prod}(\mathbf{j}) = \prod_{k=1}^p \mathbf{j}^{(k)}$ to determine such an ordering. That is, basis functions with smaller index product will be used earlier when constructing Sieve-SGD – such a tensor product basis is called hyperbolic cross in the literature [28, 101]. Before we discuss the intuition of such an ordering, we present some numerical examples of applying Sieve-SGD in multivariate regression problems.

We consider two dimensional settings of the feature variable $\mathbf{X}$: $p = 2 and 10$. The feature vector is generated as: $\mathbf{X}^{(1)} = U_1$, $\mathbf{X}^{(k)} = (U_k - U_{k-1} + 1)/2$ for $k = 2, ..., p$. Here U_k are independent Unif$[0, 1]$ variables. The true regression function is defined as

$$f_\rho(\mathbf{x}) = \sum_{k=1}^{p}\sum_{l=k}^{p}(0.5 - |\mathbf{x}^{(k)} - 0.5|)(0.5 - |\mathbf{x}^{(l)} - 0.5|), \tag{6.4}$$

And the outcome $Y = f_\rho(\mathbf{x}) + \epsilon$ is contaminated by a normal distributed noise, SNR = 3.

The main update rule of Sieve-SGD we applied here is

$$\hat{f}_i = \hat{f}_{i-1} + \gamma_0 i^{-1/(2s+1)} \left(Y_i - \hat{f}_{i-1}\left(\mathbf{X}_i\right)\right) \sum_{\mathbf{j}: c_{prod}(\mathbf{j}) \le cpi^{1/(2s+1)}} \left(\prod_{k=1}^{p} \mathbf{j}^{(k)}\right)^{-2\omega} \psi_{\mathbf{j}}\left(\mathbf{X}_i\right) \psi_{\mathbf{j}}, \qquad (6.5)$$

where $\omega = 0.51$ and $s = 2$. We use $\gamma_0 \in \{0.1, 0.5\}$ and $c \in \{4, 8\}$: the latter two parameters may be different in each replication (as tuning parameters). The index set $\{\mathbf{j} : c_{prod}(\mathbf{j}) \le cpi^{1/(2s+1)}\}$ contains the p-dimension index vectors of smallest product. For example, when $p = 2$, $\{\mathbf{j} : c_{prod}(\mathbf{j}) \le 5\} = \{(1,1),(1,2),(2,1),(1,3),(3,1)\}$. Arbitrary choice is used when there is a tie. The working basis functions we use is generated from the univariate basis:

$$\psi_1(x) = 1, \quad \psi_j = \sqrt{2}\cos((j-1)\pi x), \text{ for } j \ge 2. \qquad (6.6)$$

In Figure 6.1, we compare the statistical performance of Sieve-SGD with several popular benchmark learning methods in statistics and computer science communities. The Sobolev tensor product kernel we use there is $K(\mathbf{s}, \mathbf{t}) = \prod_{k=1}^{p} \left(1 + \min\{\mathbf{s}^{(k)}, \mathbf{t}^{(k)}\}\right)$. The RKHS corresponding to this kernel is the tensor product of univariate Sobolev spaces on $[0, 1]$.

Like many other learning methods trained with stochastic gradient descent, it is possible to have several pass over the data set to achieve better generalization ability. We choose to continue increasing the number of basis function of Sieve-SGD while processing the data multiple times. That is, after 5 epochs we are using $cp(5 \times 10^5)^{1/(2s+1)}$ basis functions. This strategy is not feasible for kernel SGD methods or fixed-dimension SGD, so we include relevant results for reference.

Now we present some intuition behind our choice of ordering the multivariate tensor product basis functions $\psi_{\mathbf{j}}$, using some basic theory of RKHS. Our readers can check section 6.3.1 and 6.3.2 for more background.

A ball in the RKHS of kernel $K(x, z) = \sum_{j=1}^{\infty} j^{-2s} \psi_j(x) \psi_j(z)$ can be identified (Theorem 6.3.3) as the ellipsoid

$$W(s, Q, \{\psi_j\}) = \left\{ f \in L_\nu^2 \mid f = \sum_{j=1}^{\infty} \theta_j \psi_j, \sum_{j=1}^{\infty} (\theta_j j^s)^2 \le Q^2 \right\}. \qquad (6.7)$$

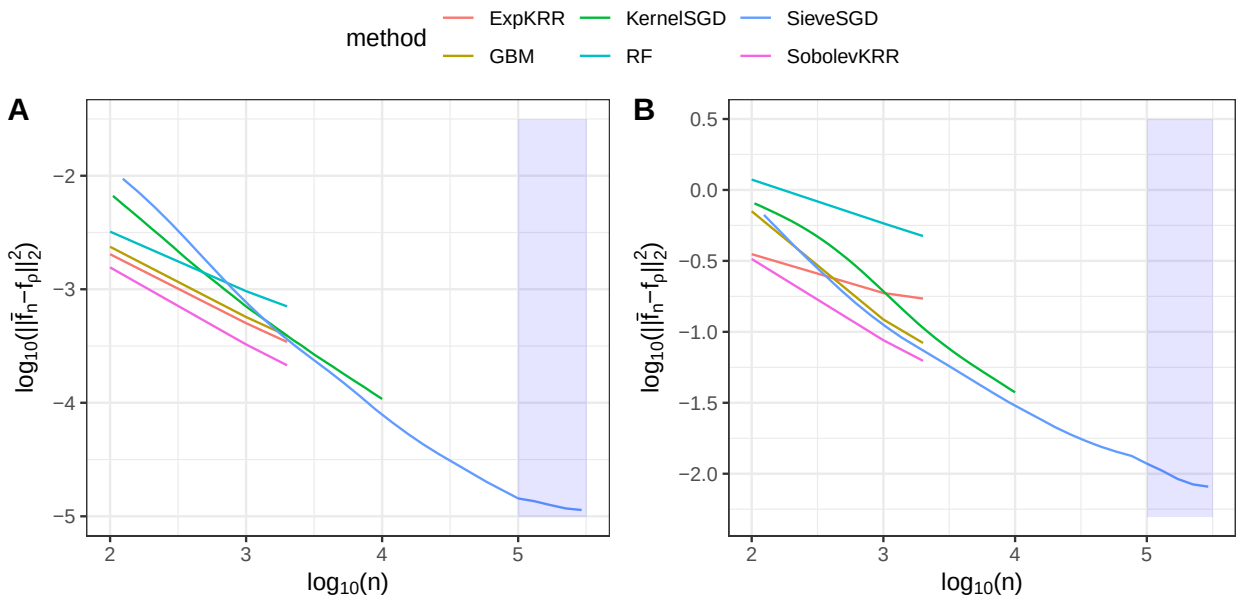

Figure 6.1: Multivariate numerical examples of applying Sieve-SGD. Other benchmark methods: ExpKRR, kernel ridge regression with Gaussian kernel; KernelSGD, [25] with tensor product Sobolev kernel; GBM, gradient boosting machine; RF, random forest; SobolevKRR, kernel ridge regression with tensor product Sobolev kernel. We present the result of each method under oracle hyperparameters (best testing error). The shaded area corresponds to the second to fifth pass of Sieve-SGD over the same training data (10^5 unique observations). (A) $p = 2$ (B) $p = 10$.

If we consider the two-dimensional tensor product kernel $\tilde{K} : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ constructed from K, that is

$$\tilde{K}(\mathbf{x}, \mathbf{z}) = K(\mathbf{x}^{(1)}, \mathbf{z}^{(1)})K(\mathbf{x}^{(2)}, \mathbf{z}^{(2)}). \tag{6.8}$$

It is known ([128] Section 12.4.2) that we have the Mercer expansion

$$\tilde{K}(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{\infty}\sum_{k=1}^{\infty}(jk)^{-2s}\psi_j(\mathbf{x}^{(1)})\psi_j(\mathbf{z}^{(1)})\psi_k(\mathbf{x}^{(2)})\psi_k(\mathbf{z}^{(2)}) \tag{6.9}$$

and a ball in the RKHS of $\tilde{K}$ takes the form

$$\tilde{W} = \left\{ f \in L^2_\nu \otimes L^2_\nu \mid f(x) = \sum_{j,k=1}^{\infty} \theta_{jk}\psi_j\left(\mathbf{x}^{(1)}\right)\psi_k\left(\mathbf{x}^{(2)}\right), \sum_{j,k=1}^{\infty}\left(\theta_{jk}(jk)^s\right)^2 \leq Q^2 \right\}. \tag{6.10}$$

We can also read out that the eigenvalues are $\prod_{k=1}^{2}\left(\mathbf{j}^{(k)}\right)^{-2s}$, $\mathbf{j} \in (\mathbb{N}^+)^2$. According to (6.10), we would intuitively expect θ_{jk} to be smaller when the *product* of the index vector is larger.

When the univariate RKHS is a Sobolev space, estimating with a tensor product kernel is essentially assuming the true regression function is in the tensor product space of Sobolev space or can be well-approximated by a function from such a space. The tensor product Sobolev space is also characterized as a Sobolev space with (dominating) mixed derivatives [99]. In statistics, such a model has been studied under the name of nonparametric Tensor product ANOVA [70] (where the regression function is estimated using kernel ridge regression). Although methods engaging with hyperbolic cross have been studied in numerical analysis in the past decade, there are few works adopting such an idea into statistics. Sobolev spaces with mixed derivatives are not homogeneous spaces in the sense that they contain functions of different smoothness in different directions. Specifically, functions in such spaces can be less smooth along the directions of coordinate axis than other directions. This can be useful in the case when the features X as a strong "main effect" on the outcome Y and a weaker "interaction effect" (in the language of [70]).

6.2.2 Additive model and Sieve-SGD

In the main text Section 3.5.4, we described the nonparametric additive model and how to use Sieve-SGD to estimate it. We simplified things by omitting the intercept term to

streamline exposition. The additive model with intercept is given by

$$f_\rho(\mathbf{x}) = \beta^0 + \sum_{k=1}^{p} f_{\rho,k}\left(\mathbf{x}^{(k)}\right) \tag{6.11}$$

for some $\beta^0 \in \mathbb{R}$ and $f_{\rho,k} \in W_k(s_k, Q_k, \{\psi_{jk}\})$ for some centered ψ_{jk} $\left(\int \psi_{jk}(x)d\nu(x) = 0\right)$ (for example the functions in (3.32)). For the additive model with intercept (6.11), the updating rule (3.26) of Sieve-SGD could be replaced by a two-step procedure:

$$\begin{aligned}
\hat{f}_i &= \hat{f}_{i-1} + \gamma_i \left(Y_i - \hat{\beta}^0_{i-1} - \sum_{k=1}^{p} \hat{f}_{i-1,k}\left(\mathbf{X}_i^{(k)}\right)\right) \sum_{k=1}^{p}\sum_{j=1}^{J_{ik}} j^{-2s_k}\psi_{jk}\left(\mathbf{X}_i^{(k)}\right)\psi_{jk} \\
\hat{\beta}^0_i &= \hat{\beta}^0_{i-1} + \gamma_i \left(Y_i - \hat{\beta}^0_{i-1} - \sum_{k=1}^{p} \hat{f}_{i-1,k}\left(\mathbf{X}_i^{(k)}\right)\right)
\end{aligned} \tag{6.12}$$

here J_{ik} is the truncation level of the k-th covariate when the sample size is equal to i; and $\hat{f}_{i-1,k}$ is the estimate of $f_{\rho,k}$. After applying Polyak averaging (averaging $\hat{\beta}^0_i + \hat{f}_i$ with previous estimates), we will get the Sieve-SGD estimate of f_ρ.

6.3 Proof of Lemma 6.2

In this section, we will prove Lemma 3.6.2, together with results regarding the spectrum of some related operators that we will use in the proof of Theorem 3.6.1 & 3.6.3. To this end, we need to prepare our readers by reminding them about some established results and ideas in the literature. In this section we will

- Define a Reproducing Kernel Hilbert Space (RKHS) formally;
- Define covariance operators characterized by a kernel and discuss related geometric properties, and;
- Define the entropy of a compact operator and relate it to the eigenvalues of the operator.

After all these, we will be ready to give a proof of Lemma 3.6.2.

In this section, we need to distinguish continuous functions and their L^2-equivalent class for a more rigorous discussion. For a given measure μ on $\mathcal{X} \subset \mathbb{R}^p$, we use $\mathcal{L}^2_\mu$ to denote the Hilbert space of all μ-square integrable functions. The L^2_μ spaces should be understand as the quotient spaces of $\mathcal{L}^2_\mu$ under the equivalence relation:

$$f = g \Leftrightarrow \int_{\mathcal{X}} (f(\tau) - g(\tau))^2 \, d\mu(\tau) = 0. \tag{6.13}$$

For a function $g \in \mathcal{L}^2_\mu$, we use $[g]_\mu \in L^2_\mu$ to denote its equivalence class. The mathematical framework [109] we present in the following subsections allows a rigorous discussion when the measure μ does not have a full-support over $\mathcal{X}$ (which is weaker than our Assumption A2), or when the RKHS is not dense in $\mathcal{L}^2_\mu$ (e.g. when RKHS is of finite dimension).

6.3.1 Mercer Kernel and RKHS

We first introduce the definition of a Mercer kernel and its corresponding RKHS.

Definition 6.3.1 (Mercer kernel)**.** *A symmetric bivariate function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive semi-definite (PSD) if for any $n \geq 1$ and $(x_i)_{i=1}^n \subset \mathcal{X}$, the $n \times n$ matrix $\mathbb{K}$ whose elements are $\mathbb{K}_{ij} = k(x_i, x_j)$ is always a PSD matrix.*

A continuous, bounded, PSD kernel function k is called a Mercer kernel*.*

In Assumption A3 of the main text, we assumed $\{\psi_j\}$ to be a set of bounded, continuous functions in $\mathcal{L}^2_\nu$.

For each $J \in \mathbb{N}^+ \cup \{\infty\}$ and $\omega > 0.5$, it is known that the bivariate functions

$$K^\omega_J(s,t) := \sum_{j=1}^{J} j^{-2\omega} \psi_j(s)\psi_j(t), \tag{6.14}$$

are Mercer kernels. We use

$$K = K(s,t) = K^s_\infty(s,t)$$

to denote the canonical (untruncated) kernel in our analysis.

It is well-known that for any Mercer kernels, there is an unique associated Hilbert space $(\mathcal{H}_K, \langle \cdot, \cdot \rangle_K)$ which has the so-called reproducing property. The following theorem formally defines such a Hilbert space and states its uniqueness.

Theorem 6.3.2 ([22]). *For a Mercer Kernel $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$, there exists an unique Hilbert Space $(\mathcal{H}_k, \langle \cdot, \cdot \rangle_k)$ of functions on $\mathbb{X}$ satisfying the following conditions. Let $k_x : z \mapsto k(x, z)$:*

1. *For all $x \in \mathbb{X}$, $k_x \in \mathcal{H}_k$.*

2. *The linear span of $\{k_x \mid x \in \mathbb{X}\}$ is dense (w.r.t $\| \cdot \|_k$) in $\mathcal{H}_k$.*

3. *For all $f \in \mathcal{H}_k, x \in \mathbb{X}$, $f(x) = \langle f, k_x \rangle_k$ (reproducing property).*

We call this Hilbert space the Reproducing kernel Hilbert space (RKHS) *associated with kernel k.*

Note that in the above theorem, we did not mention any measures on $\mathbb{X}$. The RKHS $\mathcal{H}_k$ of a kernel k is defined independently as a Hilbert space (a complete linear space equipped with an inner product).

The inner product in Theorem 6.3.2 is implicitly defined and appear to be quite abstract. However, there is an equivalent, more tangible definition of the RKHS. For example, the RKHS corresponding to the canonical kernel $K = K_\infty^s$ can be characterized as the following ellipsoid with "infinite-radius".

Theorem 6.3.3 (p.37,Theorem 4 in [22]). *Under the assumptions A2, A3, the Hilbert space $\mathcal{H}_K$ of the kernel K is identical (same function class with the same inner product) to the following Hilbert space $\mathbb{H}_K$.*

$$\mathbb{H}_K = \left\{ f \in \mathcal{L}_\nu^2(X) \mid f = \sum_{j=1}^{\infty} a_j \psi_j \quad \textit{with} \quad \sum_{j=1}^{\infty} \left(j^s a_j\right)^2 < \infty \right\} \tag{6.15}$$

Equipped with the inner product:

$$\langle f, g \rangle_K = \sum_{j=1}^{\infty} j^{2s} a_j b_j \tag{6.16}$$

for $f = \sum_j a_j \psi_j$, and $g = \sum_j b_j \psi_j$.

It is direct to check our assumption A3 is the same as assuming the conditional mean f_ρ belongs to a ball of radius Q (measure in the $\|\cdot\|_K$-norm) in the above constructed RKHS.

Under assumptions A2 and A3, the functions in $\mathcal{H}_K$ are all square-integrable.

Therefore the identity mapping (w.r.t. measure μ) id_μ is also well-defined on $\mathcal{H}_K$:

$$\begin{aligned} \mathrm{id}_\mu : \mathcal{H}_K &\to L^2_\mu \\ g &\mapsto [g]_\mu. \end{aligned} \tag{6.17}$$

6.3.2 Covariance Operators

In Section 6.3.2 and 6.3.3, we engage with the covariance operator of the canonical kernel K. Once the properties of this operator is clear, we can directly generalize our analysis techniques to other truncated kernels K_J^ω as well. Recall our definitions of T_X (covariance operator) and T_ν in the main text:

$$\begin{aligned} T_X : L^2_{\rho_X} &\to L^2_{\rho_X} \\ g &\mapsto \int_{\mathcal{X}} g(\tau) K(\tau, \cdot), d\rho_X(\tau) \end{aligned} \tag{6.18}$$

and

$$\begin{aligned} T_\nu : L^2_\nu &\to L^2_\nu \\ g &\mapsto \int g(\tau) K(\tau, \cdot) d\nu(\tau). \end{aligned} \tag{6.19}$$

Now we state several basic properties of T_X. Similar properties also hold for T_ν and can be verified much easier without abstract analysis. For proofs of Lemma 6.3.4 and other properties not listed, see [109, Section 2 & 3].

Lemma 6.3.4. *Under the assumptions A2, A3 in the main text:*

- *The operator T_X is bounded, self-adjoint, and positive.*
- *There exists a countable set of functions $\{\phi_j\} \subset \mathcal{H}_K$, $j \in \mathcal{J}$ and a countable sequence of positive numbers λ_j (decreasingly ordered) such that*

$$T_X(g) = \sum_{j \in \mathcal{J}} \lambda_j \langle g, [\phi_j]_{\rho_X} \rangle_{L^2_{\rho_X}} [\phi_j]_{\rho_X}, \quad \text{for } g \in L^2_{\rho_X}. \tag{6.20}$$

- *The* $\{[\phi_j]_{\rho_X}\}$ *above is an orthonormal system in* $L^2_{\rho_X}$ *and* $\{\sqrt{\lambda_j}\phi_j\}$ *is an orthonormal system in* $\mathcal{H}_K$*. Therefore,* $(\lambda_j, [\phi]_j)$ *is an eigensystem of* T_X*, that is:*

$$T_X([\phi_j]_{\rho_X}) = \lambda_j[\phi_j]_{\rho_X} \tag{6.21}$$

- T_X *is a trace class operator, i.e.* $\sum_{j\in\mathcal{J}} \lambda_j < \infty$*.*

Under the assumptions A2, A3, we can actually say more about ϕ_j. But the properties presented in the following lemma are not necessarily true when we discuss truncated kernels later, so we list them separately.

Lemma 6.3.5 (Theorem 3.1, [109])**.** *Under the same assumptions as in Lemma 6.3.4. Let* (λ_j, ϕ_j) *denote the eigensystem in Lemma 6.3.4. Then*

- *The family* $\{[\phi_j]_{\rho_X}\}$ *is an orthonormal* basis *of* $L^2_{\rho_X}$*.*

- *The family* $\{\sqrt{\lambda_j}\phi_j\}$ *is an orthonormal* basis *of* $\mathcal{H}_K$*.*

Now we define several operators related to T_X that we will engage with in our analysis of the spectrum of T_X.

Definition 6.3.6. *Under the same assumptions as in Lemma 6.3.4, with the same* $\{\lambda_j\}$ *and* $\{\phi_j\}$*:*

- *We define the* r*-th power of* T_X *as:*

$$\begin{aligned} T_X^r : L^2_{\rho_X} &\to L^2_{\rho_X} \\ g &\mapsto \sum_j \lambda_j^r \langle g, [\phi_j]_{\rho_X}\rangle_{L^2_{\rho_X}} [\phi_j]_{\rho_X}, \quad \text{for } g \in L^2_{\rho_X}. \end{aligned} \tag{6.22}$$

  In this chapter, we are most interested in the square-root of T_X*, i.e.* $T_X^{1/2}$*.*

- *Define the operator* $S_X^{1/2}$ *as:*

$$\begin{aligned} S_X^{1/2} : L^2_{\rho_X} &\to \mathcal{H}_K \\ g &\mapsto \sum_j \lambda_j^{1/2} \langle g, [\phi_j]_{\rho_X}\rangle_{L^2_{\rho_X}} \phi_j, \quad \text{for } g \in L^2_{\rho_X} \end{aligned} \tag{6.23}$$

The operator $S_X^{1/2}$ has a very important geometric property: it preserves distance between $L^2_{\rho_X}$ and $\mathcal{H}_K$, as stated in the following lemma.

Lemma 6.3.7 ([109],Theorem 2.11)**.** *Under the same assumptions as in Lemma 6.3.4, let $S_X^{1/2}$ be the operator defined in* (6.23)*. Then*

- *$S_X^{1/2}$ is bijective between $\overline{span\{[\phi_j]_{\rho_X}, j \in \mathcal{J}\}} = L^2_{\rho_X}$ and $\overline{span\{\phi_j, j \in \mathcal{J}\}} = \mathcal{H}_K$;*
- *$\left\|S_X^{1/2}(g)\right\|_K = \|g\|_{L^2_{\rho_X}}$, for $g \in \overline{span\{[\phi_j]_{\rho_X}, j \in \mathcal{J}\}}$.*

That is, $S_X^{1/2}$ is an isometric isomorphism between $L^2_{\rho_X}$ and $\mathcal{H}_K$.

It is direct to check the following lemma by the equivalent definition of the RKHS (Theorem 6.3.3).

Lemma 6.3.8. *Under the assumptions A2, A3 in the main text. Define $S_\nu^{1/2}$ similarly for the operator T_ν:*

$$\begin{aligned} S_\nu^{1/2} : L^2_\nu &\to \mathcal{H}_K \\ g &\mapsto \sum_j j^{-s} \langle g, [\psi_j]_\nu \rangle_{L^2_\nu} \psi_j, \quad \text{for } g \in L^2_\nu \end{aligned} \tag{6.24}$$

Then

- *$S_\nu^{1/2}$ is bijective between L^2_ν and $\mathcal{H}_K$;*
- *$\left\|S_\nu^{1/2}(g)\right\|_K = \|g\|_{L^2_\nu}$, for $g \in L^2_\nu$.*

6.3.3 Entropy of an operator and its spectrum

The above Lemma 6.3.7 and Lemma 6.3.8 is one set of the elements we are going to use in the proof of Lemma 3.6.2. Another important part of our proof is the correspondence between the spectrum of an operator and its metric entropy. We believe that these results might help show connections between proof methods regarding nonparametric problems that use the spectrum of operators [25, 140] and those using metric entropy [128, 120].

We first define the metric entropy of an operator. There is a correspondence between our definition and the "standard" definition of metric entropy for compact sets. In the following we will use B_E to denote the unit ball of a function space E.

Definition 6.3.9 (Entropy of an operator)**.** *For $k \geq 1$ we define the k-th entropy number of a metric space S to be*

$$e_k(S) = \inf\{\epsilon > 0 | \exists \text{ closed balls } D_1, \ldots, D_{2^k-1} \text{ with radius } \epsilon \text{ covering } S\} \tag{6.25}$$

If $T : E \to F$ is a linear map, then we define the k-th entropy number of T as

$$e_k(T) = e_k\left(T\left(B_E\right)\right) \tag{6.26}$$

The first result we are going to present gives a bound on the entropy of an operator using its eigenvalues.

Theorem 6.3.10. *Let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_j \geq \ldots \geq 0$ be a sequence of real numbers, (ξ_j) be an element of l^2 (space of square-summable sequences). Consider the operator defined by*

$$\begin{gathered} T : l^2 \to l^2 \\ (\xi_1, \xi_2, ..., \xi_j, ...) \mapsto (\lambda_1\xi_1, \lambda_2\xi_2, ..., \lambda_j\xi_j, ...) \end{gathered} \tag{6.27}$$

If $\lambda_j \leq Cj^{-s}$ for some C, s and all $j \geq 1$, then for all $j \geq 2$

$$e_j(T) \leq 12Cs^s j^{-s} \tag{6.28}$$

For proof, see Proposition 9 in [22, Appendix A]. The proof there uses Proposition 1.3.2 in [18], which engages with l^2 spaces as well. Theorem 6.3.10 is good enough for our purpose because we can diagonalize the covariance operators of interest (Lemma 6.3.4). For a general result of the same nature as Proposition 1.3.2, one can use Proposition 3.4.2 combined with Proposition 4.4.1 in [18].

We also state a result in the other direction: Carl's inequality, see [17, Theorem 4].

Theorem 6.3.11. *Let E be a Banach space and let $T : E \to E$ be a linear compact operator. Then*

$$|\lambda_j(T)| \leq \sqrt{2} e_j(T) \tag{6.29}$$

where $\lambda_j(T)$ is the non-increasing sequence of the eigenvalues of T.

Now we are ready to prove Lemma 3.6.2.

Proof of Lemma 3.6.2. We use B_X to denote the unit ball in $L^2_{\rho_X}$ and B_ν for the unit ball in L^2_ν. The proof of this lemma can be divided into three steps.

Step 1: bound entropy by spectrum. By the definition of the power of an operator, the eigenvalue of $T_\nu^{1/2}$ is j^{-s} (because the eigenvalue of T_ν is j^{-2s}). And the corresponding eigenfunction is ψ_j. For any function $f \in L^2_\nu$, there is a unique sequence $(f_j)_{j=1}^\infty \in l^2$ such that $f = \sum_{j=1}^\infty f_j \psi_j$.

Applying Theorem 6.3.10 to $T_\nu^{1/2}$, we get that:

$$e_j\left(T_\nu^{1/2}\right) \leq 12 s^s j^{-s} \tag{6.30}$$

Step 2: relate the entropy of operators. Now we are going to show the entropy of $T_X^{1/2}$ can be bounded by that of $T_\nu^{1/2}$ times a constant. To investigate the entropy of an operator, we only need to study the entropy of

$$\begin{aligned} &T_X^{1/2}(B_X) = \mathrm{id}_{\rho_X} \circ S_X^{1/2}(B_X), \text{ and} \\ &T_\nu^{1/2}(B_\nu) = \mathrm{id}_\nu \circ S_\nu^{1/2}(B_\nu) \end{aligned} \tag{6.31}$$

By Lemma 6.3.7 and Lemma 6.3.8, we know $S_X^{1/2}(B_X) = S_\nu^{1/2}(B_\nu) = B_{\mathcal{H}_K}$. When measuring the entropy of $T_X^{1/2}(B_X)$ we need to use id_{ρ_X} to map it to $L^2_{\rho_X}$, but for $T_\nu^{1/2}(B_\nu)$ we need to use id_ν to map it to L^2_ν.

By the assumption A2 of ρ_X, for any $f \in \mathcal{L}^2_{\rho_X} = \mathcal{L}^2_\nu$, we have

$$\int [f]^2_{\rho_X}(x) d\rho_X(x) \leq u \int [f]^2_\nu(x) d\nu(x) \tag{6.32}$$

This means we can use the center of an ϵ-cover of $T_\nu^{1/2}(B_\nu)$ to construct a $\sqrt{u}\epsilon$-cover of $T_X^{1/2}(B_X)$ using the same center points. By the definition of entropy,

$$e_j(T_X^{1/2}) \leq \sqrt{u} e_j(T_\nu^{1/2}) \tag{6.33}$$

Step 3: bound spectrum by entropy. We use Theorem 6.3.11 to translate the bound on entropy of the operator to its spectrum. For the eigenvalue λ_j of T_X, we have

$$\lambda_j = \left(\lambda_j^{1/2}\right)^2 \leq \left(\sqrt{2} e_j(T_X^{1/2})\right)^2 \leq \left(\sqrt{2u} e_j(T_\nu^{1/2})\right)^2 \leq \left(12 s^s \sqrt{2u} j^{-s}\right)^2 \tag{6.34}$$

This concludes our claim that the eigenvalues of T_X satisfy $\lambda_j = O(j^{-2s})$.

Similarly, we can show $j^{-2s} \leq C\lambda_j$ by going in the other direction (using the lower bound of the density assumed in assumption A2): Suppose $\lambda_j < \left(12 s^s \sqrt{2\ell^{-1}}\right)^{-2} j^{-2s}$, we can get the eigenvalues of T_ν are strictly less than j^{-2s}, which leads to a contradiction.

We conclude that $\lambda_j = \Theta(j^{-2s})$. □

6.3.4 Technical Results

In this subsection we will present some results that we will use in the proof of Theorem 3.6.1 and 3.6.3. Since they are related to the spectrum of covariance operators, we present them here rather than in the technical section of Appendix 6.4.

For a fixed $1 \leq J \leq \infty$ and $\omega > \frac{1}{2}$, we can define a Mercer kernel $K_J^\omega(s,t) = \sum_{j=1}^J j^{-2\omega} \psi_j(s)\psi_j(t)$. According to Theorem 6.3.2, there is a unique corresponding Hilbert space with the reproducing property. Using Theorem 6.3.3, we can also get the hierarchy relation that $\mathcal{H}_{K_J^\omega} \subset \mathcal{H}_{K_I^\omega}$ for $I \geq J$. We can also define the covariance operators of K_J^ω w.r.t. ρ_X and ν:

$$\begin{aligned} T_{X,J}^\omega : L^2_{\rho_X} &\to L^2_{\rho_X} \\ g &\mapsto \int_{\mathcal{X}} g(\tau) K_J^\omega(\tau, \cdot) d\rho_X(\tau) \end{aligned} \tag{6.35}$$

and

$$\begin{aligned} T_{\nu,J}^\omega : L^2_{\nu} &\to L^2_{\nu} \\ g &\mapsto \int g(\tau) K_J^\omega(\tau, \cdot) d\nu(\tau). \end{aligned} \tag{6.36}$$

Similar to Lemma 6.3.4, we can diagonalize $T^{\omega}_{X,J}$ with an eigensystem $(\lambda^{\omega}_{J,j}, \phi^{\omega}_{J,j})$ satisfying $T^{\omega}_{X,J}\left(\phi^{\omega}_{J,j}\right) = \lambda^{\omega}_{J,j}\phi^{\omega}_{J,j}$. However, these differ from $\{\phi_j\}$ (the eigenfunctions of T_X): $\{\phi^{\omega}_{J,j}\}$ is a basis of $\mathcal{H}_{K^{\omega}_J}$ but $\{[\phi^{\omega}_{J,j}]_{\rho_X}\}$ is not a basis of $L^2_{\rho_X}$. We formally state that in the following lemma.

Lemma 6.3.12 (Theorem 3.1, [109])**.** *Under the assumptions A2, A3. Let $(\lambda^{\omega}_{J,j}, \phi^{\omega}_{J,j})$ denote the eigensystem of $T^{\omega}_{X,J}$ similarly defined as in Lemma 6.3.4. Then*

- *The family $[\phi^{\omega}_{J,j}]_{\rho_X}$ is an orthonormal system of $L^2_{\rho_X}$.*
- *The family $\sqrt{\lambda^{\omega}_{J,j}}\phi^{\omega}_{J,j}$ is an orthonormal basis of $\mathcal{H}_{K^{\omega}_J}$.*

Related to Lemma 6.3.12, the mapping $\left(S^{\omega}_{X,J}\right)^{1/2}$ is not an isomorphism between $L^2_{\rho_X}$ and $\mathcal{H}_{K^{\omega}_J}$ – it has a non-trivial kernel space.

Lemma 6.3.13 ([109],Theorem 2.11)**.** *Under the same assumptions as in Lemma 6.3.12, define $\left(S^{\omega}_{X,J}\right)^{1/2}$ as*

$$\begin{aligned} \left(S^{\omega}_{X,J}\right)^{1/2} : L^2_{\rho_X} &\to \mathcal{H}_{K^{\omega}_J} \\ g &\mapsto \sum_j \left(\lambda^{\omega}_{J,j}\right)^{1/2} \langle g, \left[\phi^{\omega}_{J,j}\right]_{\rho_X}\rangle_{L^2_{\rho_X}} \phi^{\omega}_{J,j}, \quad \text{for } g \in L^2_{\rho_X} \end{aligned} \tag{6.37}$$

Then

- $\left(S^{\omega}_{X,J}\right)^{1/2}$ *is bijective between* $\overline{span\{[\phi^{\omega}_{J,j}]_{\rho_X}, j \in \mathcal{J}\}} \subset L^2_{\rho_X}$ *and* $\overline{span\{\phi^{\omega}_{J,j}, j \in \mathcal{J}\}} = \mathcal{H}_{K^{\omega}_J}$ *and,*
- $\left\| \left(S^{\omega}_{X,J}\right)^{1/2}(g) \right\|_{K^{\omega}_J} = \|g\|_{L^2_{\rho_X}}$, *for* $g \in \overline{span\{[\phi^{\omega}_{J,j}]_{\rho_X}, j \in \mathcal{J}\}}$.

Now we state and prove the main result of this section:

Lemma 6.3.14. *Let $1 \le J \le \infty$, $\omega > \frac{1}{2}$, assume A2 and A3 in the main text. We use $(\lambda^{\omega}_{J,j}, \phi^{\omega}_{J,j})$ to denote the eigensystem of $T^{\omega}_{X,J}$ (similarly defined as in Lemma 6.3.4). Then*

$$C_1(\ell) j^{-2\omega} \le \lambda^{\omega}_{J,j} \le C_2(u) j^{-2\omega}, \quad \text{for } 1 \le j \le J, \tag{6.38}$$

where C_1, C_2 are real numbers that only depend on u and ℓ.

Note: The constants that show up in (6.38) do not depend on the truncation level J. Therefore, for a given set of ω, ℓ, u, we can treat the result in Lemma 6.3.14 as a uniform bound.

Proof. The proof resembles that of Lemma 3.6.2. To investigate the eigenvalues of $\left(T^{\omega}_{X,J}\right)^{1/2}$, we just need to compare the entropy of $\left(T^{\omega}_{X,J}\right)^{1/2}(B_X)$ and $\left(T^{\omega}_{\nu,J}\right)^{1/2}(B_\nu)$. Since we know the $\left(T^{\omega}_{X,J}\right)^{1/2}$ operators can be further decomposed as $\left(T^{\omega}_{X,J}\right)^{1/2} = \mathrm{id}_{\rho_X} \circ \left(T^{\omega}_{X,J}\right)^{1/2}$ (similarly, $\left(T^{\omega}_{\nu,J}\right)^{1/2} = \mathrm{id}_{\nu} \circ \left(T^{\omega}_{\nu,J}\right)^{1/2}$). We need to first compare $\left(S^{\omega}_{X,J}\right)^{1/2}(B_X)$ and $\left(S^{\omega}_{\nu,J}\right)^{1/2}(B_\nu)$. We know that

$$\left(S^{\omega}_{X,J}\right)^{1/2}(B_X) \overset{(1)}{=} \text{unit ball in } \mathcal{H}_{K^{\omega}_J} = \left(S^{\omega}_{\nu,J}\right)^{1/2}(B_\nu) \tag{6.39}$$

In (1) we used the distribution assumption A2, which ensures $\{\phi^{\omega}_{J,j}\}$ is a basis of $\mathcal{H}_{K^{\omega}_J}$ (Lemma 6.3.13). For this step, we also note that for any $f \in B_X$, we can decompose it as $f = g + g^{\perp}$ where $g \in \overline{\mathrm{span}\{[\phi^{\omega}_{J,j}]_{\rho_X}, j \in \mathcal{J}\}}$ and $g^{\perp} \perp \overline{\mathrm{span}\{[\phi^{\omega}_{J,j}]_{\rho_X}, j \in \mathcal{J}\}}$. When applying $\left(T^{\omega}_{X,J}\right)^{1/2}$ to f, it is mapped to $\left(T^{\omega}_{X,J}\right)^{1/2}(g)$ since $\left(T^{\omega}_{X,J}\right)^{1/2}(g^{\perp}) = 0$.

After embedding the unit ball in $\mathcal{H}_{K^{\omega}_J}$ back to the L^2 spaces, we know an ϵ-covering of $\left(T^{\omega}_{\nu,J}\right)^{1/2}(B_\nu)$ can induce a $\sqrt{u}\epsilon$-covering of $\left(T^{\omega}_{X,J}\right)^{1/2}(B_X)$, which gives the above upper bound by similar argument as in the proof of Lemma 3.6.2 (presented in Section 6.3.3). It is also similar to show the lower bound in (6.38), noting that the feature distribution is assumed in A2 to have a strictly positive density. □

6.4 Proof of Theorem 6.1

In this section we are going to prove the main performance guarantees, result Theorem 3.6.1. In our proof, we first split the error into two parts: one part is noiseless and depends only on the the initial bias, the other is due to the noise in our data. We will bound each term separately and choose the learning rate γ_n to balance the trade-off. The last part of this section will give some technical lemmas that will be referred to in the proofs of Theorem 3.6.1. Some of the proof techniques are taken from [6] and [25].

6.4.1 Notation

In this section, the RKHSs we are considering are those associated with kernels $K^s_{J_n}(s,t) = \sum_{j=1}^{J_n} j^{-2s}\psi_j(s)\psi_j(t)$. They can be treated as subspaces of the RKHS associated with kernel $K = K^s_\infty$. We use $\|\cdot\|_K$ and $\langle\cdot,\cdot\rangle_K$ to denote the RKHS-norm and RKHS-inner product of this larger space (and the subspaces are equipped with the same inner product), which has an explicit form stated in Theorem 6.3.3. For any elements $g, h \in \mathcal{H}_K$, we define the operator $g \otimes h$ as a mapping from $\mathcal{H}_K$ to $\mathcal{H}_K$ such that $(g \otimes h)f = \langle f, h\rangle_K\, g$.

We also use K_{X_n,J_n} to denote the function of kernel evaluated at the feature vector X_n: $K_{X_n,J_n}(\cdot) = K^s_{J_n}(X_n,\cdot)$, omitting the parameter s since it is fixed in this section. As we will show in Lemma 6.4.1, the quantity $\|K_{X_n,J_n}\|^2_K$ is bounded (and this bound only depends on s). We use R^2 to denote the smallest bound for $\|K_{X_n,J_n}\|^2_K$. And any γ_n in this section is assumed to satisfy: $\gamma_0 R^2 < 1/2$ and $\gamma_n = \gamma_0 n^{-\frac{1}{2s+1}}$.

We consider a filtration of σ-algebras $\{\mathcal{F}_n\}$, where $\mathcal{F}_n$ is the σ-algebra generated by $(X_i, Y_i)_{i=1}^n$.

The sign $\preccurlyeq$ denotes the order between self-adjoint operators over the RKHS. That is, for self-adjoint operators $A, B : E \to \mathcal{H}_K$, $A \preccurlyeq B$ means $\langle f, (B-A)f\rangle_K \geq 0$ for any $f \in \mathcal{H}_K$. Intuitively we can think of $B - A$ as a positive semi-definite matrix in a finite-dimensional space. The expectation of random function/operator should be understood as the Bochner integral, a generalization of the Lebesgue integral where the random element takes value in a Banach space, see [80, 21] or Chapter 4 of [10]. The $\zeta(s)$ function that shows up in this section is the Riemann-zeta function $\zeta(s) := \sum_{k=1}^\infty k^{-s}$. We use it for simplifying the notation — it's neat that it shows up, but we do not need any of the exciting (and difficult to show) properties that number theorists/combinatorists are concerned with.

6.4.2 Separation of the error

In our theoretical analysis of the generalization error, rather than study how $\bar{f}_n$ converge to f_ρ, we can equivalently study how the difference shrinks to zero instead. We define

$$\begin{aligned} \Delta_i &= \hat{f}_i - f_\rho \\ \bar{\Delta}_i &= \frac{1}{i}\sum_{j=1}^{i} \Delta_j = \bar{f}_i - f_\rho \end{aligned} \tag{6.40}$$

for $i = 1, 2, ..., n$. We can relate the $\|\cdot\|_2$-norm (the natural norm shows up in the regression problem) and the $\|\cdot\|_K$-norm (which facilitates our theoretical analysis) using $T_X^{1/2}$. We will be repeatedly using the following equivalence in our proof:

$$\|g\|_2^2 = \left\|T_X^{1/2} g\right\|_K^2 = \langle g, T_X g\rangle_K \qquad \text{for } g \in \mathcal{H}_K \tag{6.41}$$

Similarly,

$$\|g\|_2^2 = \left\|T_{X,J_i}^{1/2} g\right\|_K^2 = \langle g, T_{X,J_i} g\rangle_K \qquad \text{for } g \in \mathcal{H}_K \cap \operatorname{span}\{\psi_1, ..., \psi_{J_i}\} \tag{6.42}$$

And we have a recursive relationship for Δ_i based on the recursive relationship for $\hat{f}_i$:

$$\begin{aligned} \hat{f}_i &= \hat{f}_{i-1} - \gamma_i (Y_i - \hat{f}_{i-1}(X_i)) K_{X_i,J_i} \\ \Rightarrow \hat{f}_i &= \left(I - \gamma_i T_{X_i,J_i}\right) \hat{f}_{i-1} + \gamma_i Y_i K_{X_i,J_i} \\ \Rightarrow \hat{f}_i - f_\rho &= \left(I - \gamma_i T_{X_i,J_i}\right) \left(\hat{f}_{i-1} - f_\rho\right) + \gamma_i \Xi_i \end{aligned} \tag{6.43}$$

where

$$\begin{aligned} T_{X_i,J_i}(f) &= f(X_i) K_{X_i,J_i} \\ \Xi_n &= \left(Y_i - f_\rho\left(X_i\right)\right) K_{X_i,J_i} \end{aligned} \tag{6.44}$$

Thus, we have a recursive formula for Δ_i:

$$\begin{aligned} \Delta_0 &= -f_\rho \\ \Delta_i &= \left(I - \gamma_i T_{X_i,J_i}\right) \Delta_{i-1} + \gamma_i \Xi_i \end{aligned} \tag{6.45}$$

We further decompose Δ_i into two parts $\Delta_i = \eta_i + \vartheta_i$. The sequence η_i is defined as

$$\begin{aligned} \eta_0 &= -f_\rho \\ \eta_i &= \left(I - \gamma_i T_{X_i,J_i}\right) \eta_{i-1} \end{aligned} \tag{6.46}$$

It is the part of Δ_i due to an initial value not equal to f_ρ. We note that it does not contain the noise term Ξ_i, so the only randomness comes from features X_i. The pure noise component ϑ_i is defined as:

$$\begin{aligned} \vartheta_0 &= 0 \\ \vartheta_i &= (I - \gamma_i T_{X_i,J_i})\,\vartheta_{i-1} + \gamma_i \Xi_i \end{aligned} \tag{6.47}$$

We can directly show that $\bar{\Delta}_i = \bar{\eta}_i + \bar{\vartheta}_i$. By Minkowski's inequality:

$$\left(E\left[\left\|\bar{\Delta}_i\right\|_2^2\right]\right)^{1/2} \leq \left(E\left[\|\bar{\eta}_i\|_2^2\right]\right)^{1/2} + \left(E\left[\left\|\bar{\vartheta}_i\right\|_2^2\right]\right)^{1/2} \tag{6.48}$$

Our job now is just to bound the two terms separately and then choose the correct γ_i, J_i to minimize the combined bound.

6.4.3 Bound on initial condition sub-process

In this section, we will engage with bounding η_n, which is the part of error due to the imperfect initialization.

proof of bound on initial value. By definition,

$$\eta_i = \eta_{i-1} - \gamma_i \eta_{i-1}(X_i) K_{X_i,J_i} \tag{6.49}$$

We square both sides w.r.t. the RKHS inner product

$$\|\eta_i\|_K^2 = \|\eta_{i-1}\|_K^2 - 2\gamma_i \eta_{i-1}(X_i)\langle \eta_{i-1}, K_{X_i,J_i}\rangle_K + \gamma_i^2 \eta_{i-1}^2(X_i)\left\|K_{X_i,J_i}\right\|_K^2 \tag{6.50}$$

We take the expectation on both sides: conditioned on $\mathcal{F}_{i-1}$ first, then unconditionally.

$$\begin{aligned} E\|\eta_i\|_K^2 =& E\|\eta_{i-1}\|_K^2 - 2\gamma_i E\left[\eta_{i-1}(X_i)\langle \eta_{i-1}, K_{X_i,J_i}\rangle_K\right] \\ &+ \gamma_i^2 E\left[\eta_{i-1}^2(X_i)\left\|K_{X_i,J_i}\right\|_K^2\right] \\ \overset{(1)}{\leq}& E\|\eta_{i-1}\|_K^2 - 2\gamma_i E\left[\eta_{i-1}(X_i)\langle \eta_{i-1}, K_{X_i,J_i}\rangle_K\right] + \gamma_i E\|\eta_{i-1}\|_2^2 \end{aligned} \tag{6.51}$$

in (1) we use the fact that for any n, $\gamma_i \left\|K_{X_i,J_i}\right\|_K^2 \leq \gamma_0 R^2 < 1$. This is actually how we choose our (γ_i). If the K_{X_i,J_i} in the middle term above is actually K_{X_i}, then the whole

middle term will become just $E\|\eta_{i-1}\|_2^2$, which makes the following algebra easier. However, since we do not use exactly K_{X_i} but truncate at level J_i: Extra care is required to deal with this.

$$\begin{aligned} E\|\eta_i\|_K^2 &\leq E\|\eta_{i-1}\|_K^2 - 2\gamma_i E\left[\eta_{i-1}(X_i)\langle\eta_{i-1}, K_{X_i} + K_{X_i,J_i} - K_{X_i}\rangle_K\right] + \gamma_i E\|\eta_{i-1}\|_2^2 \\ &= E\|\eta_{i-1}\|_K^2 - \gamma_i E\|\eta_{i-1}\|_2^2 + 2\gamma_i E\left[\eta_{i-1}(X_i)\langle\eta_{i-1}, K_{X_i} - K_{X_i,J_i}\rangle_K\right] \\ &\overset{(1)}{\leq} E\|\eta_{i-1}\|_K^2 - \gamma_i E\|\eta_{i-1}\|_2^2 + \frac{1}{2}\gamma_i E\|\eta_{i-1}\|_2^2 + 2\gamma_i E\left[\langle\eta_{i-1}, K_{X_i} - K_{X_i,J_i}\rangle_K^2\right] \end{aligned} \tag{6.52}$$

In (1) we use Young's inequality. Now we bound the last term:

$$\begin{aligned} E\left[\langle\eta_{i-1}, K_{X_i} - K_{X_i,J_i}\rangle_K^2\right] &= E\left[\langle\eta_{i-1}, \sum_{j=J_i+1}^{\infty} j^{-2s}\psi_j(X_i)\psi_j\rangle_K^2\right] \\ &= E\left[\left(\sum_{j=J_i+1}^{\infty} \eta_{i-1,j}\psi_j(X_i)\right)^2\right] \text{ where } \eta_{i-1,j} = \langle\eta_{i-1}, \psi_j\rangle_{L^2_\nu} \\ &\leq uE\left[\int_{\mathcal{X}}\left(\sum_{j=J_i+1}^{\infty} \eta_{i-1,j}\psi_j(x)\right)^2 d\nu(x)\right] \\ &\leq uJ_i^{-2s}E\left[\sum_{j=J_i+1}^{\infty}(j^s\eta_{i-1,j})^2\right] \leq uJ_i^{-2s}E\|\eta_{i-1}\|_K^2 \end{aligned} \tag{6.53}$$

Continuing from (6.52):

$$E\|\eta_i\|_K^2 \leq E\|\eta_{i-1}\|_K^2 - \frac{1}{2}\gamma_i E\|\eta_{i-1}\|_2^2 + 2u\gamma_i J_i^{-2s}E\|\eta_{i-1}\|_K^2 \tag{6.54}$$

Now for each i we have such a recursive relationship for $\|\eta_i\|_K^2$, $\|\eta_{i-1}\|_K^2$ and $\|\eta_{i-1}\|_2^2$. We can

sum this from $i = 1$ to n.

$$
\begin{aligned}
E\|\eta_n\|_K^2 &\leq E\|\eta_0\|_K^2 - \frac{1}{2}\sum_{i=1}^{n}\gamma_i E\|\eta_{i-1}\|_2^2 + 2u\sum_{i=1}^{n}\gamma_i J_i^{-2s} E\|\eta_{i-1}\|_K^2 \\
\Rightarrow \frac{1}{2}\sum_{i=1}^{n}\gamma_i E\|\eta_{i-1}\|_2^2 &\leq \|\eta_0\|_K^2 + 2u\sum_{i=1}^{n}\gamma_i J_i^{-2s} E\|\eta_{i-1}\|_K^2 \\
\Rightarrow \frac{\gamma_n}{n}\sum_{i=1}^{n} E\|\eta_{i-1}\|_2^2 &\leq 2\|\eta_0\|_K^2/n + 4u\sum_{i=1}^{n}\gamma_i J_i^{-2s} E\|\eta_{i-1}\|_K^2/n \\
\overset{(1)}{\Rightarrow} E\left\|\frac{1}{n}\sum_{i=1}^{n}\eta_{i-1}\right\|_2^2 &\leq \frac{2\|f_\rho\|_K^2}{n\gamma_n} + \frac{4u\sum_{i=1}^{n}\gamma_i J_i^{-2s} E\|\eta_{i-1}\|_K^2}{n\gamma_n} \\
\Rightarrow \left(E[\|\bar{\eta}_n\|_2^2]\right)^{1/2} &\leq \left(\frac{2\|f_\rho\|_K^2 + 4u\sum_{i=1}^{n}\gamma_i J_i^{-2s} E\|\eta_{i-1}\|_K^2}{n\gamma_n}\right)^{1/2}
\end{aligned}
\tag{6.55}
$$

Under assumptions A1-A3, we use Lemma 6.4.2 to show that for $J_i \geq i^\alpha \log^2 i \vee 1$ for some $\alpha \geq \frac{1}{2s+1}$, then the series $\sum_{i=1}^{n}\gamma_i J_i^{-2s} E\|\eta_{i-1}\|_K^2$ is convergent and uniformly bounded, that is, it can be bounded by a constant that does not depend on n. Recall that we chose $\gamma_n = \gamma_0 n^{-\frac{1}{2s+1}}$, so we conclude

$$
\left(E[\|\bar{\eta}_n\|_2^2]\right)^{1/2} = O\left(n^{-\frac{s}{2s+1}}\right) \tag{6.56}
$$

□

6.4.4 Bound on noise sub-process

In this section we bound the part of error that is due to the noise in the stochastic gradient. We remind the reader of the definition of our noise sub-process:

$$
\begin{aligned}
\vartheta_0 &= 0 \\
\vartheta_i &= \left(I - \gamma_i T_{X_i,J_i}\right)\vartheta_{i-1} + \gamma_i \Xi_i
\end{aligned}
\tag{6.57}
$$

where $\Xi_i = \left(Y_i - f_\rho\left(X_i\right)\right) K_{X_i,J_i}$ (we also remind our reader K_{X_i,J_i} is the "truncated kernel" at level J_i). Also, we recall the definition of T_{X,J_i} and T_{X_i,J_i}:

$$
\begin{aligned}
T_{X,J_i}(f) &= \int_{\mathcal{X}} \langle f, K_{x,J_i}\rangle_K K_{x,J_i} d\rho_X(x) \quad \text{population operator} \\
T_{X_i,J_i}(f) &= \langle f, K_{X_i,J_i}\rangle_K K_{X_i,J_i} \quad \text{random operator}
\end{aligned}
\tag{6.58}
$$

proof of bound on noise. We need to define several sequences that are related to ϑ_i for our technical analysis. The first sequence is:

$$\begin{aligned} \eta_0^0 &= 0 \\ \eta_i^0 &= (I - \gamma_i T_{X,J_i})\eta_{i-1}^0 + \gamma_i \Xi_i^0 \end{aligned} \tag{6.59}$$

where $\Xi_i^0 = \Xi_i = (Y_i - f_\rho(X_i)) K_{X_i,J_i}$.We also define additional sequences, η_i^r, for each integer $r \geq 1$, by

$$\begin{aligned} \eta_0^r &= 0 \\ \eta_i^r &= (I - \gamma_i T_{X,J_i})\eta_{i-1}^r + \gamma_i \Xi_i^r \end{aligned} \tag{6.60}$$

where $\Xi_i^r = (T_{X,J_i} - T_{X_i,J_i})\, \eta_{i-1}^{r-1}$. These sequences are easier to analyze than ϑ_i because the operator in the recursive relationship $(I - \gamma_i T_{X,J_i})$ is a deterministic (population) operator. In contrast the operator in the original ϑ_i is random. We show in Lemma 6.4.3 that as r increases, the amplitudes of "noise" Ξ_i^r get smaller. Additionally, using the fact that all the sequences η_i^r start with $\eta_0^r = 0$, we can show that η_i^r becomes more concentrated about 0 for larger r.

We split the noise process ϑ_i into two parts:

$$\vartheta_i = \left(\vartheta_i - \sum_{k=0}^{r} \eta_i^k\right) + \sum_{k=0}^{r} \eta_i^k. \tag{6.61}$$

So its average (over the iteration trajectory) satisfies

$$\bar{\vartheta}_i = \left(\bar{\vartheta}_i - \sum_{k=0}^{r} \bar{\eta}_i^k\right) + \sum_{k=0}^{r} \bar{\eta}_i^k. \tag{6.62}$$

Here $\bar{\eta}_i^k$ is also an averaged sequence of η_i^k (i.e. $\bar{\eta}_i^k = \frac{1}{i}\sum_{j=1}^{i} \eta_j^k$).

Applying Minkowski' s inequality gives us

$$\left(E\|\bar{\vartheta}_n\|_2^2\right)^{1/2} \leq \sum_{k=0}^{r} \left(E\|\bar{\eta}_n^k\|_2^2\right)^{1/2} + \left(E\|\bar{\vartheta}_n - \sum_{k=0}^{r} \bar{\eta}_n^k\|_2^2\right)^{1/2}. \tag{6.63}$$

Now we define

$$\alpha_n^r = \vartheta_n - \sum_{k=0}^{r} \eta_n^k \tag{6.64}$$

We will show (Lemma 6.4.7) that for $r \geq n$, we have $\alpha_n^r = \eta_n^r = 0$. We can see the second term in (6.63) is exactly zero when we choose $r \geq n$, that is:

$$\bar{\vartheta}_n - \sum_{k=0}^{r} \bar{\eta}_n^k = \frac{1}{n}\sum_{i=1}^{n}\left(\vartheta_n - \sum_{k=0}^{r}\eta_n^k\right) = \frac{1}{n}\sum_{i=1}^{n}\alpha_n^r = 0 \tag{6.65}$$

Now our task is just bounding the first term $\sum_{k=0}^{r}(E\|\bar{\eta}_n^k\|_2^2)^{1/2}$, we analyze the summation term by term. In Lemma 6.4.4, we show that:

$$E\left[\left\|\bar{\eta}_n^k\right\|_2^2\right] = O\left(\gamma_0^k R^{2k} C_\epsilon^2 n^{-\frac{2s}{2s+1}}\right). \tag{6.66}$$

To get this result, we first show in Lemma 6.4.3 that the Ξ_j^k variables are centered and satisfy some moment bounds. With these properties in hand, we prove the above result in Lemma 6.4.4, with the help of some technical results (Lemma 6.4.5 and 6.4.6).

Following (6.66), we have

$$\begin{aligned}\sum_{k=0}^{r}\left(E\|\bar{\eta}_n^k\|_2^2\right)^{1/2} &\leq \sum_{k=0}^{r} C(\gamma_0 R^2)^{k/2} C_\epsilon n^{-\frac{s}{2s+1}} \\ &\leq C C_\epsilon n^{-\frac{s}{2s+1}} \sum_{k=0}^{\infty}(\gamma_0 R^2)^{k/2} \\ &= C C_\epsilon n^{-\frac{s}{2s+1}} \frac{1}{1-\sqrt{\gamma_0 R^2}} = O\left(n^{-\frac{s}{2s+1}}\right)\end{aligned} \tag{6.67}$$

we note that $\gamma_0 R^2 < 1$, which allows us to sum up the geometric series.

Combining the results in (6.67) with $\alpha_n^r = 0$ for $r \geq n$, we can conclude from (6.63) that, for $r \geq n$:

$$\left(E\left[\left\|\bar{\vartheta}_n\right\|_2^2\right]\right)^{1/2} \leq \sum_{k=0}^{r}\left(E\|\bar{\eta}_n^k\|_2^2\right)^{1/2} + 0 = O\left(n^{-\frac{s}{2s+1}}\right) \tag{6.68}$$

□

6.4.5 Combining the bounds

Plugging the final bounds (6.56) and (6.68) back into (6.48), we have the desired result:

$$E\left[\left\|\bar{f}_n - f_\rho\right\|_2^2\right] = O\left(n^{-\frac{2s}{2s+1}}\right) \tag{6.69}$$

6.4.6 Technical Results

Lemma 6.4.1. *There exists $R < \infty$, such that:*

$$\|K_{X_n,J_n}\|_K^2 \leq R^2 \tag{6.70}$$

for any $J_n \geq 1, X_n \in \mathcal{X}$.

Proof. By the definition of $\|\cdot\|_K$ we have:

$$\begin{aligned} \|K_{X_n,J_n}\|_K^2 &= \left\| \sum_{j=1}^{J_n} j^{-2s} \psi_j(X_n) \psi_j \right\|_K^2 \\ &= \sum_{j=1}^{J_n} \frac{j^{-4s} \psi_j^2(X_n)}{j^{-2s}} \\ &\leq M^2 \zeta(2s) =: R^2 \end{aligned} \tag{6.71}$$

where $\zeta(\cdot)$ is the Riemann-zeta function. □

We use R^2 rather than R, in the bounds in this lemma because it simplifies calculation where this lemma is applied

Lemma 6.4.2. *Under A1-A3, and if we choose $J_i \geq i^\alpha \log^2 i \vee 1$ for some $\alpha \geq \frac{1}{2s+1}$, $\gamma_i = \gamma_0 i^{-\frac{1}{2s+1}}$, then there exists a number C that does not depend on n such that*

$$\sum_{i=1}^{n} \gamma_i J_i^{-2s} E\|\eta_{i-1}\|_K^2 \leq C \tag{6.72}$$

for all n.

Proof. We first show the expectation of $\|\eta_n\|_K^2$ can be uniformly bounded for J_n that increases fast enough. Recall we have the following recursive relationship (6.54):

$$\begin{aligned} E\|\eta_n\|_K^2 &\leq E\|\eta_{n-1}\|_K^2 - \frac{1}{2}\gamma_n E\|\eta_{n-1}\|_2^2 + 2u\gamma_n J_n^{-2s} E\|\eta_{n-1}\|_K^2 \\ &\leq (1 + 2u\gamma_n J_n^{-2s}) E\|\eta_{n-1}\|_K^2 \\ \Rightarrow E\|\eta_n\|_K^2 &\leq \prod_{i=1}^{n} (1 + 2u\gamma_i J_i^{-2s}) \|\eta_0\|_K^2 \end{aligned} \tag{6.73}$$

When we take $J_i \geq i^\alpha \log^2 i \vee 1$, for some $\alpha \geq \frac{1}{2s+1}$ we have $\gamma_i J_i^{-2s} \leq \gamma_0 \left((i \log^2 i)^{-1} \wedge 1\right)$. Therefore $\prod_{i=1}^n (1 + 2u\gamma_i J_i^{-2s})$ converges as $n \to \infty$:

$$\begin{aligned}\prod_{i=1}^n (1 + 2u\gamma_i J_i^{-2s}) &\leq \prod_{i=1}^n (1 + C(i \log^2 i)^{-1} \wedge 1) \\ &= \exp\left(\log(\prod_{i=1}^n (1 + C(i \log^2 i)^{-1} \wedge 1)\right) \\ &= \exp\left(\sum_{i=1}^n \log((1 + C(i \log^2 i)^{-1} \wedge 1)\right) \\ &\leq \exp\left(\sum_{i=1}^n C(i \log^2 i)^{-1} \wedge 1\right)\end{aligned} \tag{6.74}$$

We note that the series in the exponent of the last line converges. So at this point we know, $E\|\eta_n\|_K^2$ can be uniformly bounded by $K\|\eta_0\|_K = K\|f_\rho\|_K$, with a number K that does not depend on n. Now it is direct to control the quantity of interest:

$$\sum_{i=1}^n \gamma_i J_i^{-2s} E\|\eta_{i-1}\|_K^2 \leq K\|f_\rho\| \sum_{i=1}^n \gamma_i J_i^{-2s} < \infty, \tag{6.75}$$

when $J_i \geq i^\alpha \log^2 i \vee 1$ with $\alpha \geq \frac{1}{2s+1}$. □

Lemma 6.4.3. *Assume A1-A4, for any integer $r, n \geq 0$ we have:*

- Ξ_n^r *is* $\mathcal{F}_n$ *measurable and* $\Xi_n^r \in span\{\psi_1, ..., \psi_{J_n}\}$,
- $E[\Xi_n^r \mid \mathcal{F}_{n-1}] = 0$, *and*
- $E\left[\Xi_n^r \otimes \Xi_n^r\right] \preccurlyeq \gamma_0^r R^{2r} C_\epsilon^2 T_{X,J_n}$.

Here $\mathcal{F}_n$ is the σ-algebra generated by $(X_i, Y_i)_{i=1}^n$.

Note: In the third line in Lemma 6.4.3, because $\gamma_0 R^2 < 1$ by our choice of γ_0, the upper bound on $E\left[\Xi_n^r \otimes \Xi_n^r\right]$ is smaller for larger r.

Proof. We remind our readers that the definitions of Ξ_n^r and η_n^r are given around (6.60). The first claim is direct. We first note that η_n^0, Ξ_n^0 are both $\mathcal{F}_n$-measurable and belong to $\text{span}\{\psi_1, ..., \psi_{J_n}\}$. Then we can show the corresponding properties of η_n^r, Ξ_n^r by induction.

For the second claim, we calculate directly:

$$\begin{aligned} E[\Xi_n^r|\mathcal{F}_{n-1}] &= E[(T_{X,J_n} - T_{X_n,J_n})\eta_{n-1}^{r-1}|\mathcal{F}_{n-1}] \\ &= E[(T_{X,J_n} - T_{X_n,J_n})|\mathcal{F}_{n-1}]\eta_{n-1}^{r-1} = 0 \end{aligned} \tag{6.76}$$

Now we show the last claim. We define

$$D_k^n = (I - \gamma_n T_{X,J_n})(I - \gamma_{n-1} T_{X,J_{n-1}}) \cdots (I - \gamma_k T_{X,J_k}). \tag{6.77}$$

Note that each of the element in D_k^n is self-adjoint, positive but they in general do not commute. Because T_{X,J_i} is a positive operator, we have $I - \gamma_i T_{X,J_i} \preccurlyeq I$ for our choice of γ_i. We are also going to use the following relationship in the rest of this proof. Recall that we denote the adjoint of operator A as A^*:

$$\begin{aligned} \sum_{k=1}^n D_{k+1}^n \gamma_k^2 T_{X,J_k} \left(D_{k+1}^n\right)^* &\preccurlyeq \gamma_0 \sum_{k=1}^n D_{k+1}^n \gamma_k T_{X,J_k} \left(D_{k+1}^n\right)^* \\ &\overset{(1)}{=} \gamma_0 \sum_{k=1}^n D_{k+1}^n \left(D_{k+1}^n\right)^* - D_k^n \left(D_{k+1}^n\right)^* \\ &= \gamma_0 \sum_{k=1}^n (I - \gamma_n T_{X,J_n}) \cdots \left(I - \gamma_{k+1} T_{X,J_{k+1}}\right) \left(I - \gamma_{k+1} T_{X,J_{k+1}}\right) \cdots (I - \gamma_n T_{X,J_n}) \\ &\qquad - D_k^n \left(D_{k+1}^n\right)^* \\ &\overset{(2)}{\preccurlyeq} \gamma_0 \sum_{k=1}^n (I - \gamma_n T_{X,J_n}) \cdots \left(I - \gamma_{k+1} T_{X,J_{k+1}}\right) \cdots (I - \gamma_n T_{X,J_n}) - D_k^n \left(D_{k+1}^n\right)^* \\ &= \gamma_0 \sum_{k=1}^n D_{k+1}^n \left(D_{k+2}^n\right)^* - D_k^n \left(D_{k+1}^n\right)^* = \gamma_0 \left(I - D_1^n \left(D_2^n\right)^*\right) \preccurlyeq \gamma_0 I \end{aligned} \tag{6.78}$$

In (1) we used $D_{k+1}^n \gamma_k T_{X,J_k} = D_{k+1}^n - D_k^n$. In (2) we used

$$ABBA^* = AB^{1/2}BB^{1/2}A^* \preccurlyeq ABA \tag{6.79}$$

for positive, self-adjoint $B \preccurlyeq I$.

Now we give an inductive argument (applying induction on r):

Initialisation:

When $r = 0$, recall that $\Xi_n^0 = (Y_n - f_\rho(X_n))K_{X_n,J_n}$. Thus, we have that

$$\begin{aligned}\langle f, E\left[\Xi_n^0 \otimes \Xi_n^0\right](f)\rangle_K &= \langle f, E\left[(Y_n - f_\rho(X_n))^2 \langle K_{X_n,J_n}, f\rangle_K K_{X_n,J_n}\right]\rangle_K \\ &\overset{(1)}{\leq} \langle f, C_\epsilon^2 E\left[\langle K_{X_n,J_n}, f\rangle_K K_{X_n,J_n}\right]\rangle_K \\ &= C_\epsilon^2 \langle f, T_{X,J_n} f\rangle_K\end{aligned} \tag{6.80}$$

In (1) we used our noise assumption (A4). So we have

$$E\left[\Xi_n^0 \otimes \Xi_n^0\right] \preccurlyeq C_\epsilon T_{X,J_n} \tag{6.81}$$

To perform induction over r, we also need a bound on $E\left[\eta_n^0 \otimes \eta_n^0\right]$ as well. Recall that $\eta_n^0 = \sum_{k=1}^n \gamma_k D_{k+1}^n \Xi_k^0$ as defined in (6.59). Thus we have:

$$\begin{aligned}E\left[\eta_n^0 \otimes \eta_n^0\right] &= \sum_{k=1}^n \sum_{j=1}^n \gamma_j \gamma_k D_{j+1}^n E\left[\Xi_j^0 \otimes \Xi_k^0\right]\left(D_{k+1}^n\right)^* \\ &\overset{(1)}{=} \sum_{k=1}^n \gamma_k^2 D_{k+1}^n E\left[\Xi_k^0 \otimes \Xi_k^0\right]\left(D_{k+1}^n\right)^* \\ &\preccurlyeq C_\epsilon \sum_{k=1}^n D_{k+1}^n \gamma_k^2 T_{X,J_k}\left(D_{k+1}^n\right)^* \\ &\preccurlyeq C_\epsilon \gamma_0 I\end{aligned} \tag{6.82}$$

In (1) the interaction terms vanish because the noise variables Ξ_j^0, Ξ_k^0 are mean-zero and independent when $j \neq k$.

Induction : If we assume for $r \geq 0$,

$$E\left[\Xi_n^r \otimes \Xi_n^r\right] \preccurlyeq \gamma_0^r R^{2r} C_\epsilon T_{X,J_n} \tag{6.83}$$

and

$$E\left[\eta_n^r \otimes \eta_n^r\right] \preccurlyeq \gamma_0^{r+1} R^{2r} C_\epsilon I \tag{6.84}$$

then for $r+1$:

$$\begin{aligned} E\left[\Xi_n^{r+1}\otimes\Xi_n^{r+1}\right] &\overset{(1)}{=} E\left[\left(T_{X,J_n}-T_{X_n,J_n}\right)\eta_{n-1}^r\otimes\eta_{n-1}^r\left(T_{X,J_n}-T_{X_n,J_n}\right)\right] \\ &= E\left[\left(T_{X,J_n}-T_{X_n,J_n}\right)E\left[\eta_{n-1}^r\otimes\eta_{n-1}^r\right]\left(T_{X,J_n}-T_{X_n,J_n}\right)\right] \\ &\preccurlyeq \gamma_0^{r+1}R^{2r}C_\epsilon E\left[\left(T_{X,J_n}-T_{X_n,J_n}\right)^2\right] \\ &= \gamma_0^{r+1}R^{2r}C_\epsilon\left(E\left[\left(T_{X_n,J_n}\right)^2\right]-T_{X,J_n}^2\right) \\ &\overset{(2)}{\preccurlyeq} \gamma_0^{r+1}R^{2r+2}C_\epsilon T_{X,J_n} \end{aligned} \tag{6.85}$$

Here (1) is the definition of Ξ_n^{r+1}. For (2), it is sufficient to show $E\left[\left(T_{X_n,J_n}\right)^2\right]\preccurlyeq R^2T_{X,J_n}$ (T_{X,J_n}^2 is non-negative). This is true because:

$$\langle f,E\left[\left(T_{X_n,J_n}\right)^2\right]f\rangle_K=E[\|T_{X_n,J_n}(f)\|_K^2]=E[\langle f,K_{X_n,J_n}\rangle_K^2\|K_{X_n,J_n}\|_K^2]\le R^2\langle f,T_{X,J_n}f\rangle_K \tag{6.86}$$

Recall that $\eta_n^{r+1}=\sum_{k=1}^n D_{k+1}^n\gamma_k\Xi_k^{r+1}$, then

$$\begin{aligned} E\left[\eta_n^{r+1}\otimes\eta_n^{r+1}\right] &= E\left[\sum_{k=1}^n\gamma_k^2D_{k+1}^n\Xi_k^{r+1}\otimes\Xi_k^{r+1}\left(D_{k+1}^n\right)^*\right] \\ &= \sum_{k=1}^n\gamma_k^2D_{k+1}^nE[\Xi_k^{r+1}\otimes\Xi_k^{r+1}]\left(D_{k+1}^n\right)^* \\ &\preccurlyeq C_\epsilon\gamma_0^{r+1}R^{2r}\sum_{k=1}^n D_{k+1}^n\gamma_kT_{X,J_k}\left(D_{k+1}^n\right)^* \\ &\preccurlyeq C_\epsilon\gamma_0^{r+2}R^{2r+2}I \end{aligned} \tag{6.87}$$

□

Lemma 6.4.4. *Under assumptions A1-A4, we have*

$$E\left[\|\bar{\eta}_n^r\|_2^2\right]=O\left(\gamma_0^rR^{2r}C_\epsilon^2n^{-\frac{2s}{2s+1}}\right) \tag{6.88}$$

Proof.

$$
\begin{aligned}
n^2 E\left[\|\bar{\eta}_n^r\|_2^2\right] &= E\left\|\sum_{j=1}^{n}\sum_{k=1}^{j}\left[\prod_{i=k+1}^{j}(I-\gamma_i T_{X,J_i})\right]\gamma_k\Xi_k^r\right\|_2^2 \\
&= E\left\|\sum_{k=1}^{n}\sum_{j=k}^{n}\left[\prod_{i=k+1}^{j}(I-\gamma_i T_{X,J_i})\right]\gamma_k\Xi_k^r\right\|_2^2 \\
&= \sum_{k=1}^{n}\gamma_k^2 E\left\|\sum_{j=k}^{n}\left[\prod_{i=k+1}^{j}(I-\gamma_i T_{X,J_i})\right]\Xi_k^r\right\|_2^2 \\
&= \sum_{k=1}^{n}\gamma_k^2 E\left\langle\sum_{j=k}^{n}\left[\prod_{i=k+1}^{j}(I-\gamma_i T_{X,J_i})\right]\Xi_k^r, T_{X,J_n}\sum_{j=k}^{n}\left[\prod_{i=k+1}^{j}(I-\gamma_i T_{X,J_i})\right]\Xi_k^r\right\rangle_K \\
&= \sum_{k=1}^{n}\gamma_k^2 E\,\mathrm{tr}\left(T_{X,J_n}M_k^n\Xi_k^r\otimes\Xi_k^r\left(M_k^n\right)^*\right) \quad \text{where } M_k^n=\sum_{j=k}^{n}\left[\prod_{i=k+1}^{j}(I-\gamma_i T_{X,J_i})\right] \\
&= \sum_{k=1}^{n}\gamma_k^2\sum_{l=k}^{n}\sum_{m=k}^{n}\mathrm{tr}\left(T_{X,J_n}\left[\prod_{i=k+1}^{l}(I-\gamma_i T_{X,J_i})\right]E[\Xi_k^r\otimes\Xi_k^r]\left[\prod_{i=k+1}^{m}(I-\gamma_i T_{X,J_i})\right]\right) \\
&\overset{(1)}{\leq} \sum_{k=1}^{n}\gamma_k^2\sum_{l=k}^{n}\sum_{m=k}^{n}\sum_{t=1}^{\infty}\left(\lambda_{n,t}\left[\prod_{i=k+1}^{l}(1-\gamma_i\lambda_{i,t})\right]\gamma_0^r R^{2r}C_\epsilon^2\lambda_{k,t}\left[\prod_{i=k+1}^{m}(1-\gamma_i\lambda_{i,t})\right]\right) \\
&\overset{(2)}{\leq} \gamma_0^r R^{2r}C_\epsilon^2\sum_{k=1}^{n}\gamma_k^2\sum_{t=1}^{J_k}\lambda_{k,t}\lambda_{n,t}\left(\sum_{j=k}^{n}\left[\prod_{i=k+1}^{j}(1-\gamma_i C\lambda_{k,t})\right]\right)^2.
\end{aligned}
\tag{6.89}
$$

Here we denote the t-th eigenvalue of T_{X,J_k} as $\lambda_{k,t}\geq 0$. In (1), we used the trace inequality $\mathrm{tr}(A_1\cdots A_m)\leq\sum_t\lambda_t(A_1)\cdots\lambda_t(A_m)$, where $\lambda_t(A)$ takes the t-th largest eigenvalue of A (p.342 of [76]). In this step we also used $E\left[\Xi_n^r\otimes\Xi_n^r\right]\preccurlyeq\gamma_0^r R^{2r}C_\epsilon^2 T_{X,J_n}$, stated in Lemma 6.4.3. In step (2) we used the rank of T_{X,J_k} is at most J_k. We also apply the uniform bound on the eigenvalues stated in Lemma 6.3.14.

We claim we can further extend the inequality as follows, the gap will be bridged as a

technical lemma in Lemma 6.4.5

$$
\begin{aligned}
(\gamma_0^r R^{2r} C_\epsilon^2)^{-1} n^2 E\left[\|\bar{\eta}_n^r\|_2^2\right] &\leq \sum_{k=1}^{n} \gamma_k^2 \sum_{t=1}^{J_k} \lambda_{k,t}\lambda_{n,t}\left((n-k)^2 \wedge C\left(\lambda_{k,t}^{-2/(1-\zeta)} + \lambda_{k,t}^{-2} k^{2\zeta}\right)\right) \\
&\leq \underbrace{\sum_{k=1}^{n} \gamma_k^2 \sum_{t=1}^{J_k} \lambda_{k,t}\lambda_{n,t}\left((n-k)^2 \wedge C\lambda_{k,t}^{-2/(1-\zeta)}\right)}_{S_1} \\
&\quad + \underbrace{\sum_{k=1}^{n} \gamma_k^2 \sum_{t=1}^{J_k} \lambda_{k,t}\lambda_{n,t}\left((n-k)^2 \wedge \lambda_{k,t}^{-2} k^{2\zeta}\right)}_{S_2} \overset{(1)}{=} O\left(n^{2-\frac{2s}{2s+1}}\right)
\end{aligned}
\tag{6.90}
$$

In step (1), we can show that both S_1 and S_2 are of order $O\left(n^{2-\frac{2s}{2s+1}}\right)$. These results are provided in Lemma 6.4.6, which concludes our proof. □

Lemma 6.4.5. *Using the same notation as the last line of* (6.89)*. We have*

$$
\sum_{j=k}^{n}\left[\prod_{i=k+1}^{j}(1-\gamma_i\lambda_{k,t})\right] \leq (n-k) \wedge C\left(\lambda_{k,t}^{-1/(1-\zeta)} + \lambda_{k,t}^{-1} k^{\zeta}\right) \tag{6.91}
$$

where $\zeta = \frac{1}{2s+1}$.

Proof. We first bound the inside term:

$$
\begin{aligned}
\prod_{i=k+1}^{j}(1-\gamma_i\lambda_{k,t}) &= \prod_{i=k+1}^{j} \exp\left(\log(1-\gamma_i\lambda_{k,t})\right) \\
&\leq \exp\left(-\sum_{i=k+1}^{j}(\gamma_i\lambda_{k,t})\right) \\
&\leq \exp\left(-\lambda_{k,t}\int_{u=k+1}^{j+1}\left(\frac{1}{u^\zeta}du\right)\right) \quad \left(\gamma_i = \gamma_0 i^{-\zeta}\right) \\
&\leq \exp\left(-\lambda_{k,t}\frac{(j+1)^{1-\zeta} - (k+1)^{1-\zeta}}{1-\zeta}\right)
\end{aligned}
\tag{6.92}
$$

Then we have

$$
\begin{aligned}
\sum_{j=k}^{n}\prod_{i=k+1}^{j}(1-\gamma_i\lambda_{k,t}) &\leq \sum_{j=k}^{n} \exp\left(-\lambda_{k,t}\frac{(j+1)^{1-\zeta} - (k+1)^{1-\zeta}}{1-\zeta}\right) \\
&\leq \int_{k}^{n} \exp\left(-\lambda_{k,t}\frac{(u+1)^{1-\zeta} - (k+1)^{1-\zeta}}{1-\zeta}\right) du
\end{aligned}
\tag{6.93}
$$

We provide two upper bounds for this quantity: The first one is simply $n-k$, because $\zeta < 1/2$ and $n, k, t \geq 1$.

Now we derive the second bound:

$$\int_k^n \exp\left(-\lambda_{k,t}\frac{(u+1)^{1-\zeta}-(k+1)^{1-\zeta}}{1-\zeta}\right)du = \int_{k+1}^{n+1} \exp\left(-\lambda_{k,t}\frac{u^{1-\zeta}-(k+1)^{1-\zeta}}{1-\zeta}\right)du \tag{6.94}$$

Now we perform a change of variables, denote $\rho = 1-\zeta$:

$$\begin{aligned}
v^\rho &= \rho^{-1}\lambda_{k,t}\left((u)^\rho - (k+1)^\rho\right) \\
v &= \rho^{-1/\rho}\lambda_{k,t}^{1/\rho}\left((u)^\rho - (k+1)^\rho\right)^{1/\rho} \\
dv &= \rho^{-1/\rho}\lambda_{k,t}^{1/\rho}\left(1-\left(\frac{k+1}{u}\right)^\rho\right)^{1/\rho-1}du \\
du &= \rho^{1/\rho}\lambda_{k,t}^{-1/\rho}\left(1-\left(\frac{k+1}{u}\right)^\rho\right)^{1-1/\rho}dv \\
&= \rho^{1/\rho}\lambda_{k,t}^{-1/\rho}\left(1-\frac{(k+1)^\rho}{v^\rho\rho\lambda_{k,t}^{-1}+(k+1)^\rho}\right)^{1-1/\rho}dv \\
&= \rho^{1/\rho}\lambda_{k,t}^{-1/\rho}\left(1+\frac{(k+1)^\rho}{v^\rho\rho\lambda_{k,t}^{-1}}\right)^{1/\rho-1}dv
\end{aligned} \tag{6.95}$$

Plug this into (6.94):

$$\begin{aligned}
\sum_{j=k}^{n}\prod_{i=k+1}^{j}(1-\gamma_i\lambda_{k,t}) &\leq \int_0^\infty \rho^{1/\rho}\lambda_{k,t}^{-1/\rho}\left(1+\frac{(k+1)^\rho}{v^\rho\rho\lambda_{k,t}^{-1}}\right)^{1/\rho-1}\exp\left(-v^\rho\right)dv \\
&\leq 2^{1/\rho-1}\rho^{1/\rho}\lambda_{k,t}^{-1/\rho}\int_0^\infty\left(1\vee\frac{(k+1)^\rho}{v^\rho\rho\lambda_{k,t}^{-1}}\right)^{1/\rho-1}\exp\left(-v^\rho\right)dv \\
&\leq 2^{1/\rho-1}\rho^{1/\rho}\lambda_{k,t}^{-1/\rho}\left(I_1\vee(k+1)^{1-\rho}\lambda_{k,t}^{1/\rho-1}I_2\right) \\
&= 2^{1/\rho-1}\rho^{1/\rho}I_1\lambda_{k,t}^{-1/\rho}\vee 2^{1-2\rho+1/\rho}\rho^{1/\rho}I_2k^{1-\rho}\lambda_{k,t}^{-1}
\end{aligned} \tag{6.96}$$

which concludes the lemma. □

Lemma 6.4.6. *For both S_1 and S_2 in* (6.90)*, we have*

$$S_i = O\left(n^{2-\frac{2s}{2s+1}}\right) \quad i=1,2 \tag{6.97}$$

Proof. First we derive the bound for S_1, denote $\rho = 1 - \zeta$:

$$\begin{aligned}
S_1 &= \sum_{k=1}^{n} \gamma_k^2 \sum_{t=1}^{J_k} \lambda_{k,t}\lambda_{n,t} \left((n-k)^2 \wedge C\lambda_{k,t}^{-2/\rho} \right) \\
&\overset{(1)}{\leq} C \sum_{k=1}^{n} \gamma_k^2 \sum_{t=1}^{\infty} t^{-4s} \left((n-k)^2 \wedge Ct^{4s/\rho} \right) \\
&= C \sum_{k=1}^{n} \gamma_k^2 \left(\sum_{t=1}^{(n-k)^{\rho/2s}} t^{4s(1/\rho-1)} + (n-k)^2 \sum_{t=(n-k)^{\rho/2s}}^{\infty} t^{-4s} \right) \\
&\leq C \sum_{k=1}^{n} \gamma_k^2 \left((n-k)^{2-2\rho+\rho/2s} + (n-k)^2 (n-k)^{(-4s+1)\rho/2s} \right) \\
&\leq C \sum_{k=1}^{n} \gamma_k^2 (n-k)^{3\zeta} = C \sum_{k=1}^{n} k^{-2\zeta} (n-k)^{3\zeta} \\
&= C \sum_{k=1}^{n} \left(\frac{n}{k} - 1 \right)^{3\zeta} k^{\zeta} = n^{\zeta} \sum_{k=1}^{n} \left(\frac{1}{k/n} - 1 \right)^{3\zeta} \left(\frac{k}{n} \right)^{\zeta} \\
&= Cn^{1+\zeta} \left(\frac{1}{n} \sum_{k=1}^{n} \left(\frac{1}{k/n} - 1 \right)^{3\zeta} \left(\frac{k}{n} \right)^{\zeta} \right) \\
&= Cn^{1+\zeta} \left(\frac{1}{n} \sum_{k=1}^{n} \left(\frac{1}{k/n} - 1 \right)^{2\zeta} \left(1 - \frac{k}{n} \right)^{\zeta} \right)
\end{aligned} \tag{6.98}$$

In (1) we used the bound for $\lambda_{k,t}$ and $\lambda_{n,t}$ proved in Lemma 6.3.14. Next, we use

$$\int_0^1 \left(\frac{1}{x} - 1 \right)^{2\zeta} (1-x)^{\zeta} dx \leq \int_0^1 \left(\frac{1}{x} - 1 \right)^{2\zeta} dx < \infty \tag{6.99}$$

So for S_1 we conclude

$$S_1 \leq Cn^{1+\zeta} = O(n^{2-\frac{2s}{2s+1}}) \tag{6.100}$$

Now we bound S_2:

$$\begin{aligned}
S_2 &\le C\sum_{k=1}^{n}\gamma_k^2\sum_{t=1}^{\infty}\left(t^{-4s}(n-k)^2\wedge k^{2\zeta}\right)\\
&\le C\sum_{k=1}^{n}\gamma_k^2\left(\sum_{t=1}^{(n-k)^{\frac{1}{2s}}/k^{\frac{\zeta}{2s}}}k^{2\zeta}+\sum_{t=(n-k)^{\frac{1}{2s}}/k^{\frac{\zeta}{2s}}}^{\infty}t^{-4s}(n-k)^2\right)\\
&\le C\sum_{k=1}^{n}\gamma_k^2\left(k^{2\zeta}\sum_{t=1}^{(n-k)^{\frac{1}{2s}}/k^{\frac{\zeta}{2s}}}1+(n-k)^2\sum_{t=(n-k)^{\frac{1}{2s}}/k^{\frac{\zeta}{2s}}}^{\infty}t^{-4s}\right)\\
&\le C\sum_{k=1}^{n}\gamma_k^2\left(k^{2\zeta}\frac{(n-k)^{\frac{1}{2s}}}{k^{\frac{\zeta}{2s}}}+(n-k)^2\left(\frac{(n-k)^{\frac{1}{2s}}}{k^{\frac{\zeta}{2s}}}\right)^{1-4s}\right) \qquad (6.101)\\
&= C\sum_{k=1}^{n}\gamma_k^2\left(k^{2\zeta-\frac{\zeta}{2s}}(n-k)^{\frac{1}{2s}}+(n-k)^{\frac{1}{2s}}k^{\frac{\zeta}{2s}(4s-1)}\right)\\
&= C\sum_{k=1}^{n}\frac{1}{k^{2\zeta}}(n-k)^{\frac{1}{2s}}k^{\frac{\zeta}{2s}(4s-1)} = C\sum_{k=1}^{n}k^{-\frac{\zeta}{2s}}(n-k)^{\frac{1}{2s}}\\
&= Cn^{\left(1-\frac{\zeta}{2s}+\frac{1}{2s}\right)}\left(\frac{1}{n}\sum_{k=1}^{n}\left(\frac{k}{n}\right)^{-\frac{\zeta}{2s}}\left(1-\frac{k}{n}\right)^{\frac{1}{2s}}\right)\\
&\overset{(1)}{\le} Cn^{\left(1+\frac{1-\zeta}{2s}\right)} = O\left(n^{2-\frac{2s}{2s+1}}\right),
\end{aligned}$$

in (1) we use

$$\begin{aligned}
\int_0^1 x^{-\zeta/2s}(1-x)^{1/(2s)}dx &\le \int_0^1 x^{-\zeta/(2s)}dx\\
&= \int_1^{\infty} u^{\zeta/(2s)-2}du < \infty
\end{aligned} \qquad (6.102)$$

□

Lemma 6.4.7. *Let η_n^r be the sequences defined in* (6.60)*, then for any $r \ge n$ we have*

$$\eta_n^r = 0. \qquad (6.103)$$

As a further consequence, for α_n^r defined in (6.64)*, for any $r \ge n$, we have*

$$\alpha_n^r = 0. \qquad (6.104)$$

Proof. We prove both results by induction (over n). We recall the definition of Ξ_k^r, for $n, r \geq 1$:

$$\Xi_n^r = (T_{X,J_n} - T_{X_n,J_n})\eta_{n-1}^{r-1} \tag{6.105}$$

Let's first show $\eta_n^r = 0$ for any $r \geq n$.

When $n = 0$, by definition for any $r \geq 0$, $\eta_0^r = 0$.

Now assume for k and any $r \geq k$ we have $\eta_k^r = 0$, then for any $r \geq k+1$

$$\begin{aligned} \eta_{k+1}^r &= (I - \gamma_{k+1}T_{X,J_{k+1}})\eta_k^r + \gamma_{k+1}\Xi_{k+1}^r \\ &= 0 + \gamma_{k+1}\left(T_{X,J_{k+1}} - T_{X_{k+1},J_{k+1}}\right)\eta_k^{r-1} \\ &= 0 + 0 \end{aligned} \tag{6.106}$$

This shows that $\eta_{k+1}^r = 0$ for any $r \geq k+1$.

Now we prove the second part. Here we need to use the following recursive relationship of α_n^r (proof is postponed later):

$$\alpha_n^r = (I - \gamma_n T_{X_n,J_n})\,\alpha_{n-1}^r + \gamma_n \Xi_n^{r+1} \tag{6.107}$$

When $n = 0$, by definition $\vartheta_0 = \sum_{k=0}^r \eta_0^k$ for any $r \geq 0$. Therefore $\alpha_0^r = 0$ for any $r \geq 0$.

Then assume for k we have $\forall r \geq k$, $\alpha_k^r = 0$, then for $r \geq k+1$

$$\begin{aligned} \alpha_{k+1}^r &= \left(I - \gamma_{k+1}T_{X_{k+1},J_{k+1}}\right)\alpha_k^r + \gamma_{k+1}\Xi_{k+1}^{r+1} \\ &= 0 + \gamma_{k+1}\left(T_{X,J_{k+1}} - T_{X_{k+1},J_{k+1}}\right)\eta_k^r \\ &= 0 + 0 \end{aligned} \tag{6.108}$$

Now we just need to verify the claimed recursive formula (6.107).

$$\begin{aligned}
(I-\gamma_n T_{X_n,J_n})\alpha_{n-1}^r+\gamma_n\Xi_n^{r+1} &= (I-\gamma_n T_{X_n,J_n})\,\alpha_{n-1}^r+\gamma_n(T_{X,J_n}-T_{X_n,J_n})\eta_{n-1}^r\\
&= (I-\gamma_n T_{X_n,J_n})\,\vartheta_{n-1}-\sum_{k=0}^{r}(I-\gamma_n T_{X_n,J_n})\eta_{n-1}^k+\gamma_n(T_{X,J_n}-T_{X_n,J_n})\eta_{n-1}^r\\
&= \vartheta_n-\gamma_n\Xi_n-\sum_{k=0}^{r}(I-\gamma_n T_{X_n,J_n}+\gamma_n T_{X,J_n}-\gamma_n T_{X,J_n})\eta_{n-1}^k\\
&\quad+\gamma_n(T_{X,J_n}-T_{X_n,J_n})\eta_{n-1}^r\\
&= \vartheta_n-\gamma_n\Xi_n-\sum_{k=0}^{r}(I-\gamma_n T_{X,J_n})\eta_{n-1}^k-\sum_{k=0}^{r}\gamma_n\Xi_n^{k+1}+\gamma_n\Xi_n^{r+1}\\
&= \vartheta_n-\sum_{k=0}^{r}\eta_n^k=\alpha_n^r
\end{aligned}\tag{6.109}$$

□

6.5 Proof of Theorem 6.3

In this section we will show Sieve-SGD achieves a near-optimal convergence rate under the parameter regime specified in Theorem 3.6.3 in the main text. The proof is similar to that of Theorem 3.6.1. But in the section, we need to consider the RKHSs associated with kernels

$$K_{J_n}^{\omega}(s,t)=\sum_{j=1}^{J_n}j^{-2\omega}\psi_j(s)\psi_j(t),\quad \text{with } J_n=\lfloor n^{\frac{1}{2s+1}}\log^2 n\rfloor,\omega\in\left(\frac{1}{2},s\right).\tag{6.110}$$

To clarify, our reader should treat ω as a fixed value and J_n is a deterministic sequence that increases with n. The aforementioned series of RKHSs are subspaces of the RKHS (denoted as $\mathcal{H}_{K_\infty^\omega}$) spanned by the kernel

$$K_{\infty}^{\omega}(s,t)=\sum_{j=1}^{\infty}j^{-2\omega}\psi_j(s)\psi_j(t),\tag{6.111}$$

equipped with the same inner product

$$\langle f,g\rangle_{K_\infty^\omega}=\sum_{j=1}^{\infty}j^{2\omega}\langle f,\psi_j\rangle_{L_\nu^2}\langle g,\psi_j\rangle_{L_\nu^2}.\tag{6.112}$$

Note that, the above inner products no longer have a direct correspondence with our ellipsoid assumptions.

6.5.1 Separation of the error

Similar to section 6.4.2, we consider the following stochastic sequences. The first one is the "total deviation" sequence Δ_i:

$$\begin{aligned} \Delta_0 &= -f_\rho \\ \Delta_i &= \left(I - \gamma_i T^{\omega}_{X_i,J_i}\right) \Delta_{i-1} + \gamma_i \Xi_i, \end{aligned} \tag{6.113}$$

where

$$\begin{aligned} T^{\omega}_{X_i,J_i}(f) &= f\left(X_i\right) K^{\omega}_{X_i,J_i} = f\left(X_i\right) \left(\sum_{j=1}^{J_i} j^{-2\omega} \psi_j(X_i)\psi_j\right) \\ \Xi_i &= \left(Y_i - f_\rho\left(X_i\right)\right) K^{\omega}_{X_i,J_i} = \left(Y_i - f_\rho\left(X_i\right)\right) \left(\sum_{j=1}^{J_i} j^{-2\omega} \psi_j(X_i)\psi_j\right). \end{aligned} \tag{6.114}$$

The average of Δ_i is the difference between Sieve-SGD and f_ρ:

$$\bar{\Delta}_i = \frac{1}{i}\sum_{j=1}^{i} \Delta_j = \bar{f_i} - f_\rho \tag{6.115}$$

Similarly, we decompose the Δ_i into two parts $\Delta_i = \eta_i + \vartheta_i$, where

$$\begin{aligned} \eta_0 &= -f_\rho \\ \eta_i &= \left(I - \gamma_i T^{\omega}_{X_i,J_i}\right) \eta_{i-1}, \end{aligned} \tag{6.116}$$

and

$$\begin{aligned} \vartheta_0 &= 0 \\ \vartheta_i &= \left(I - \gamma_i T^{\omega}_{X_i,J_i}\right) \vartheta_{i-1} + \gamma_i \Xi_i \end{aligned} \tag{6.117}$$

We give bounds on $E\left[\|\bar{\eta}_i\|_2^2\right]$ and $E\left[\left\|\bar{\vartheta}_i\right\|_2^2\right]$ separately and combine them to get one for $E\left[\left\|\bar{\Delta}_i\right\|_2^2\right]$.

6.5.2 Bound on initial condition sub-process.

Proof sketch of bound on initial value. The proof formally resembles section 6.4.3 very closely. But we will engage with kernels $K^{\omega}_{X_n,J_n}$ and the RKHS inner products here are different. To

ensure $\|\eta_0\|^2_{K^\omega_\infty} = \sum_{j=1}^{\infty} \left(j^\omega \langle f_\rho, \psi_j\rangle_{L^2_\nu}\right)^2$ (or in general $\|\eta_n\|^2_{K^\omega_\infty}$) finite, we need $\omega \leq s$. Define $R^2 = M^2\zeta(2\omega)$, it is also direct to verify that $\gamma_i \|K_{X_i,J_i}\|^2_{K^\omega_\infty} \leq \gamma_0 R^2 < 1$ by our choice of γ_n. The conclusion from this is:

$$\left(E\left[\|\bar{\eta}_n\|_2^2\right]\right)^{1/2} = \left(E\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\eta_i\right\|_2^2\right]\right)^{1/2} = O\left(n^{-\frac{s}{2s+1}}\right) \tag{6.118}$$

□

6.5.3 Bound on noise sub-process

The basic structure of proof is similar to the corresponding part of Theorem 3.6.1. But the details are different: In Lemma 6.4.4, we used the fact that $t_j = j^{-2s}$ decreases quickly enough to control the magnitude of the noise; however, here we will leverage the finiteness of operators to give a different (and technically slightly simpler) bound, which is unique to sieve-type SGD.

proof of bound on noise. We still need the following working sequences to facilitate the analysis:

$$\begin{aligned} \eta_0^0 &= 0 \\ \eta_i^0 &= \left(I - \gamma_i T^\omega_{X,J_i}\right)\eta_{i-1}^0 + \gamma_i \Xi_i^0 \end{aligned} \tag{6.119}$$

where $\Xi_i^0 = \Xi_i = \left(Y_i - f_\rho\left(X_i\right)\right) K^\omega_{X_i,J_i}$ and

$$T^\omega_{X,J_i}(f) = \int_{\mathcal{X}} \left\langle f, K^\omega_{x,J_i}\right\rangle_{K^\omega_\infty} K^\omega_{x,J_i} d\rho_X(x) = \int_{\mathcal{X}} f(x)\left(\sum_{j=1}^{J_n} j^{-2\omega}\psi_j(x)\psi_j\right) d\rho_X(x) \tag{6.120}$$

For each $r > 0$, we define

$$\begin{aligned} \eta_0^r &= 0 \\ \eta_i^r &= \left(I - \gamma_i T^\omega_{X,J_i}\right)\eta_{i-1}^r + \gamma_i \Xi_i^r \end{aligned} \tag{6.121}$$

where $\Xi_i^r = \left(T^\omega_{X,J_i} - T^\omega_{X_i,J_i}\right)\eta_{i-1}^{r-1}$. Then we have

$$\begin{aligned}\left(E\left\|\bar{\vartheta}_n\right\|_2^2\right)^{1/2} &\leq \sum_{k=0}^{r}\left(E\left\|\bar{\eta}_n^k\right\|_2^2\right)^{1/2}+\left(E\left\|\bar{\vartheta}_n-\sum_{k=0}^{r}\bar{\eta}_n^k\right\|_2^2\right)^{1/2}\\ &\overset{(1)}{=}\sum_{k=0}^{r}\left(E\left\|\bar{\eta}_n^k\right\|_2^2\right)^{1/2}+0, \quad \text{when } r\geq n\\ &\overset{(2)}{\leq}\sum_{k=0}^{r} C\left(\gamma_0 R^2\right)^{k/2} C_\epsilon n^{-s/(2s+1)}\log n, \quad \text{with } R^2=M^2\zeta(2\omega)\\ &= O\left(n^{-s/(2s+1)}\log n\right).\end{aligned} \tag{6.122}$$

In (1) we used Lemma 6.4.7 (after taking another average). Step (2) leveraged the finiteness of the rank of T^{ω}_{X,J_n}, which is given in Lemma 6.5.1. Our choice of $\omega > \frac{1}{2}$ ensures that R is a finite number which does not depend on n. □

Lemma 6.5.1. *Under assumptions A1-A3, we have*

$$E\left[\left\|\bar{\eta}_n^r\right\|_2^2\right] = O\left(\gamma_0^r R^{2r} C_\epsilon^2 n^{-2s/(2s+1)}\log^2 n\right) \tag{6.123}$$

Proof. Denote $\zeta = \frac{1}{2s+1}, \rho = 1-\zeta$. According to the proof of Lemma 6.4.4 (equation (6.90)), we have (now $\lambda_{k,t}$ is the t-th largest eigenvalue of T^{ω}_{X,J_k}):

$$\begin{aligned}(\gamma_0^r R^{2r} C_\epsilon^2)^{-1} n^2 E\left[\left\|\bar{\eta}_n^r\right\|_2^2\right] &\leq \sum_{k=1}^{n}\gamma_k^2\sum_{t=1}^{J_k}\lambda_{k,t}\lambda_{n,t}\left(C\lambda_{k,t}^{-2/\rho}\right)+\lambda_{k,t}\lambda_{n,t}\left(\lambda_{k,t}^{-2}k^{2\zeta}\right)\\ &\overset{(1)}{\leq} C\sum_{k=1}^{n}\gamma_k^2\sum_{t=1}^{J_k}\left(t^{-4\omega+4\omega/\rho}+k^{2\zeta}\right)\\ &\leq C\sum_{k=1}^{n}k^{-2\zeta}(J_k)^{4\omega/\rho-4\omega+1}+J_k\\ &\overset{(2)}{\leq} C\sum_{k=1}^{n}k^{\zeta}+k^{\zeta}\log^2 k = O\left(n^{1+\zeta}\log^2 n\right)\end{aligned} \tag{6.124}$$

In step (1) we used the result of Lemma 6.3.14. For step (2) we note $\omega < s$. □

6.6 Space Expense Analysis

In this section, we are going to formally model how round-off errors appear in the process of collecting data and constructing the Sieve-SGD estimator. We are also going to characterize

how to optimally asymptotically increase space expense to ensure that round-off error does not affect model performance (beyond a multiplicative log term). Under minor simplification, we will show in Section 6.6.1 that $O(\log(n))$ times more space resources (counted in bits) is enough to make the influence of round-off error on statistical performance negligible. On the other hand, in Section 6.6.2 we will give the minimal space expense (also counted in bits) required for constructing a statistically rate-optimal estimator (using any procedures). Notably, the optimal space expense of Sieve-SGD only differs from this lower bound by a polylog term, therefore we claim the space expense of Sieve-SGD is almost optimal.

Notation: the left subscript ${}_r\cdot$ will be used to denote quantities that are directly related to round-off error.

6.6.1 Sieve-SGD under round-off error

In this subsection we are going to give an analysis of how a $O(\log^3(n)n^{\frac{1}{2s+1}})$-sized version of Sieve-SGD can achieve the optimal rate for estimating f_ρ, under assumptions A1 - A4 and some extra assumptions (A5,A6) regarding round-off error. We focus on the case when $\omega = s$ (Theorem 3.6.1). Very similar argument can be applied to the case when $\omega \neq s$ to proof Sieve-SGD can achieve near-optimality with the save space expense (Theorem 3.6.3).

We note that the size of the estimators above can be decomposed as

$$\log^3(n)n^{\frac{1}{2s+1}} = \log^2(n)n^{\frac{1}{2s+1}} \cdot \log(n), \tag{6.125}$$

where the $\log^2(n)n^{\frac{1}{2s+1}}$ term corresponds to the minimal number of basis functions needed to construct Sieve-SGD as stated in Theorem 3.6.1, and the extra logarithm term is due to the precision loss when storing a real number as a float point number.

Modern statistical estimation procedures are performed exclusively with the help of digital computers. Although computers cannot store general real numbers with arbitrary precision, statisticians usually do not count in such ubiquitous round-off errors when analyzing statistical procedures due to their tiny magnitude (for an example when it may cause some troubles, see [115]). However, we need to model and analyze in a finer scale because our

space expense is calculated in the unit of bit (rather than *number* of basis function or *number* of float point numbers). Let's be more specific about the round-off error in our estimation setting:

Recall the Sieve-SGD updating rule:

$$\hat{f}_i = \hat{f}_{i-1} + \gamma_i(Y_i - \hat{f}_{i-1}(X_i))K_{X_i,J_i} \tag{6.126}$$

and we denote $\hat{f}_i = \sum_{j=1}^{J_i} \hat{\beta}_{ij}\psi_j$. The above function update can be reduced to a simutanous update of J_i regression coefficients $\hat{\beta}_{ij}$ (as stated in Appendix 6.1):

$$\hat{\beta}_{ij} = \hat{\beta}_{(i-1)j} + \gamma_i(Y_i - \hat{f}_{i-1}(X_i))j^{-2s}\psi_j(X_i) \tag{6.127}$$

However, because general real numbers cannot be stored in a computer with infinite precision, the right-hand-side quantity of the above update rule cannot be evaluated perfectly. What is calculated and stored in the computer is a round-off version instead:

$$\text{round}_i\left(\hat{\beta}_{(i-1)j} + \gamma_i(Y_i - \hat{f}_{i-1}(X_i))j^{-2s}\psi_j(X_i)\right) \tag{6.128}$$

Here $\text{round}_i(z)$ rounds/truncates the decimal expansion of z after some digit (which we allow to be a function of i). Thus, there is round-off error between the rounded version and the exact version, which we denote as

$${}_r\epsilon_{ij} := \hat{\beta}_{(i-1)j} + \gamma_i(Y_i - \hat{f}_{i-1}(X_i))j^{-2s}\psi_j(X_i) - \text{round}_i\left(\hat{\beta}_{(i-1)j} + \gamma_i(Y_i - \hat{f}_{i-1}(X_i))j^{-2s}\psi_j(X_i)\right) \tag{6.129}$$

The round-off error is due, both, to the inexact storage of data X_i, Y_i, and potentially inexact evaluation of the intermediate quantities such as $\hat{f}_{i-1}(X_i), \psi_j(X_i)$. Even in the case when all the above is done without round-off, once we store the coefficients in computer memory, an inevitable precision loss will be introduced because only a finite length of memory is assigned to each $\hat{\beta}_{ij}$.

In assumption A5 we formalize a sequence of Sieve-SGD estimates contaminated by round-off errors and specify how small we require the errors to be to maintain statistical rate-optimality of our estimator:

A5 (Iteration with round-off error) The recursive relation of Sieve-SGD (3.26) is given under round-off error. That is

$$\hat{f}_i = \underbrace{\hat{f}_{i-1} + \gamma_i(Y_i - \hat{f}_{i-1}(X_i))K_{X_i,J_i}}_{\text{exact value we should have assigned to } \hat{f}_i} + \underbrace{\sum_{j=1}^{J_i} {}_r\epsilon_{ij}\psi_j}_{\text{round-off error (in function form)}} \tag{6.130}$$

Moreover, for each $j = 1, ..., J_i$, we assume the round-off error sequence (indexed by i) ${}_r\epsilon_{ij}$ is of order $o(i^{-2})$.

There is an equivalent way to express our assumption: Let $\hat{f}_i = \sum_{j=1}^{J_i} \hat{\beta}_{ij}\psi_j$, we assume the updating of coefficient $\hat{\beta}_{ij}$ is under round-off error ${}_r\epsilon_{ij}$, i.e.

$$\hat{\beta}_{ij} = \hat{\beta}_{(i-1)j} + \gamma_i(Y_i - \hat{f}_{i-1}(X_i))j^{-2s}\psi_j(X_i) + {}_r\epsilon_{ij} \tag{6.131}$$

where the round-off errors ${}_r\epsilon_{ij} \in \mathbb{R}, j = 1, ..., J_i$ are of order $o(i^{-2})$.

Note 1: As our readers will see very soon, we propose to assign more digits to store each $\hat{\beta}_{ij}$ as more data is collected. This will result in round-off errors that decrease as the sample size i increases.

Note 2: We assumed the round-off error of updating each coefficient $\hat{\beta}_{ij}$ is of order $o(i^{-2})$. There are many other options that people have to model the size of the round-off error: Maybe the upper bound ($o(i^{-2})$) should not only depend on n, but also depend on j (for each j, ${}_r\epsilon_{ij} = o(a_{ij})$ with some decreasing sequence a_{ij}); Alternatively, we could have not put assumptions on the *difference* between the exact value and the rounded one, but assume their *ratio* is not too far away from 1. The treatment we present in this study could be extended: Further discussion of other candidate assumptions is left to future work.

Theorem 6.6.1. *Under the same assumptions as Theorem 3.6.1, if we further assume the round-off error satisfy the assumption A5. Then the Sieve-estimator* $\bar{f}_i = \frac{1}{i}\sum_{k=1}^{i} \hat{f}_k$*, where* $\hat{f}_k$*'s are contaminated by the round-off error, is still rate-optimal for estimating* f_ρ*.*

Proof. The proof of this theorem is basically the same as that of Theorem 3.6.1. The only difference now is there is an extra round-off error term in the recursion. Recall in the proof of Theorem 3.6.1 in Appendix 6.4 we define the difference between our estimates and f_ρ as Δ_i:

$$\begin{aligned} \Delta_i &= \hat{f}_i - f_\rho \\ \bar{\Delta}_i &= \bar{f}_i - f_\rho \end{aligned} \tag{6.132}$$

And we have a recursive formula for Δ_i under A5:

$$\begin{aligned} \Delta_0 &= -f_\rho \\ \Delta_i &= \left(I - \gamma_i T_{X_i,J_i}\right) \Delta_{i-1} + \gamma_i \Xi_i + {}_r\Xi_i \end{aligned} \tag{6.133}$$

where

$$\begin{aligned} T_{X_i,J_i}(f) &= f(X_i) K_{X_i,J_i} \\ \Xi_i &= \left(Y_i - f_\rho\left(X_i\right)\right) K_{X_i,J_i} \\ {}_r\Xi_i &= \sum_{j=1}^{J_i} {}_r\epsilon_{ij} \psi_j \end{aligned} \tag{6.134}$$

Here ${}_r\Xi_i$ represents the influence of the round-off error. All we are going to do is show this is a higher order error term. Similar to our previous proofs, we further decompose Δ_i into two parts: $\Delta_i = \eta_i + \vartheta_i$:

1. (η_i) is defined as :

$$\begin{aligned} \eta_0 &= -f_\rho \\ \eta_i &= \left(I - \gamma_i T_{X_i,J_i}\right) \eta_{i-1} \end{aligned} \tag{6.135}$$

   We note η_i is exactly the same sequence as in Section 6.4.2, which means we already have the optimal bound on it. We do not need to worry about it in the rest of this proof.

2. The pure noise part (ϑ_i) now has the round-off error noise:

$$\begin{aligned} \vartheta_0 &= 0 \\ \vartheta_i &= \left(I - \gamma_i T_{X_i,J_i}\right) \vartheta_{i-1} + \gamma_i \Xi_i + {}_r\Xi_i \end{aligned} \tag{6.136}$$

To control $E[\|\bar{\vartheta}_i\|_2^2]$, we introduce η_i^k for $k \geq 0$ as in Section 6.4.4, that is

$$\begin{aligned} \eta_0^0 &= 0 \\ \eta_i^0 &= (I - \gamma_i T_{X,J_i})\eta_{i-1}^0 + \gamma_i \Xi_i^0 \end{aligned} \tag{6.137}$$

where $\Xi_i^0 := \Xi_i = (Y_i - f_\rho(X_i)) K_{X_i,J_i}$. And for each integer $k > 0$:

$$\begin{aligned} \eta_0^k &= 0 \\ \eta_i^k &= (I - \gamma_i T_{X,J_i})\eta_{i-1}^k + \gamma_i \Xi_i^k \end{aligned} \tag{6.138}$$

where $\Xi_i^k = (T_{X,J_i} - T_{X_i,J_i})\, \eta_{i-1}^{k-1}$. And now we need define another sequence to count in the round-off error:

$$\begin{aligned} {}_r\eta_0^0 &= 0 \\ {}_r\eta_i^0 &= (I - \gamma_i T_{X,J_i}) \left({}_r\eta_{i-1}^0\right) + \gamma_i \Xi_i^0 + {}_r\Xi_i \end{aligned} \tag{6.139}$$

Similar to Lemma 6.4.7, we can verify that

$$\vartheta_n = {}_r\eta_n^0 + \sum_{k=1}^{m} \eta_n^k, \quad \text{for } m \geq n \tag{6.140}$$

Then we use the triangular inequality:

$$\begin{aligned} \left(E\|\bar{\vartheta}_n\|_2^2\right)^{1/2} &\leq \left(E\|{}_r\bar{\eta}_n^0\|_2^2\right)^{1/2} + \sum_{k=1}^{m} \left(E\|\bar{\eta}_n^k\|_2^2\right)^{1/2} + \left(E\|\bar{\eta}_n - {}_r\bar{\eta}_n^0 - \sum_{k=1}^{m} \bar{\eta}_n^k\|_2^2\right)^{1/2} \\ &\overset{(1)}{\leq} o(n^{-\frac{s}{2s+1}}) + \sum_{k=0}^{m} \left(E\|\bar{\eta}_n^k\|_2^2\right)^{1/2} + 0 \quad \text{for } m \geq n \\ &\leq O(n^{-\frac{s}{2s+1}}) \end{aligned} \tag{6.141}$$

Here is a more detailed calculation of step (1)

$$
\begin{aligned}
E\left[\left\|{}_r\bar{\eta}_n^0\right\|_2^2\right] &= E\left[\left\langle {}_r\bar{\eta}_n^0, T_{X,J_n}\, {}_r\bar{\eta}_n^0\right\rangle_K\right] = E\left[\left\|T_{X,J_n}^{1/2}\, {}_r\bar{\eta}_n^0\right\|_K^2\right] \\
&= \frac{1}{n^2} E\left[\left\| T_{X,J_n}^{1/2} \sum_{i=1}^{n}\sum_{j=1}^{i}\left[\prod_{l=j+1}^{i}\left(I-\gamma_l T_{X,J_l}\right)\right]\left(\gamma_j \Xi_j^0 + {}_r\Xi_j\right)\right\|_K^2\right] \\
&\le E\left[\left\|\bar{\eta}_n^0\right\|_2^2\right] + \frac{1}{n^2} E\left[\left\|\sum_{i=1}^{n}\sum_{j=1}^{i} {}_r\Xi_j\right\|_K^2\right] \\
&\le E\left[\left\|\bar{\eta}_n^0\right\|_2^2\right] + n\sum_{i=1}^{n}\left\|{}_r\Xi_i\right\|_K^2 \\
&\le E\left[\left\|\bar{\eta}_n^0\right\|_2^2\right] + n\sum_{i=1}^{n} {}_r\epsilon_i^2 \cdot i
\end{aligned}
\tag{6.142}
$$

When ${}_r\epsilon_i^2 = o(i^{-4})$, second round-off error term will become higher order, and we have the desired optimal rate. □

Now we specify how we model the decrease of the round-off errors as we use a longer binary sequence to store $\hat{\beta}_{nj}$

A6 An $(\alpha+1)\log(i)$-long binary sequence is needed for each of the coefficient $\hat{\beta}_{ij}$ (6.131) to ensure the round-off errors ${}_r\epsilon_{ij}$ to be of order $o(i^{-\alpha})$.

We state this as an assumption, rather than a result because our theoretical roundoff error model allows for potential error to be introduced at multiple places in our update. In the case that everything is calculated exactly, and the only error comes from a final truncation, then it is straightforward to show that A6 holds.

We now give some intuition for the assumption. If we have a $(\alpha+1)\log(i)$-long binary sequence in hand, we can use it to specify $2^{(\alpha+1)\log(i)} \sim i^{\alpha+1}$ numbers. Therefore, for any number a that belongs to a bounded interval $[-M, M]$, we can 1) specify an equally-spaced grid using this binary sequence (there are $\sim i^{\alpha+1}$ grid points); and 2) there must exist a grid point that can approximate any number with an error less than $\sim Mi^{-(\alpha+1)}$. This is the

basic intuition that how a $\alpha \log(i)$ length binary sequence should in general give us an $i^{-\alpha}$ accuracy.

Our assumption A6 does not perfectly match with how float point numbers are used in modern computer. The protocol of IEEE 754 standard of float point representation is significantly more complicated and technical [85] than the simplification we present in A6. However, our assumption still captures the main relationship between binary sequence length and round-off error in the sense that every one more digit will give us a doubled accuracy to represent a real number.

Now we state our main result of this section, which can be best understood when compared with Theorem 3.6.5.

Corollary 6.6.2. *Under assumptions A1-A6, there is a $O(\log^3(n)n^{\frac{1}{2s+1}})$-sized version of Sieve-SGD that can achieve the minimax optimal statistical convergence rate.*

Proof. We managed to show in Theorem 6.6.1 that when the round-off error ${}_r\epsilon_{nj}$ is of size $o(n^{-2})$, Sieve-SGD can still achieve the optimal convergence rate. Under A6, it means we need a $3\log(n)$-length binary sequence to specify the coefficients $\hat{\beta}_{nj}$. Because there are $J_n = O(\log^2(n)n^{\frac{1}{2s+1}})$ coefficients used when sample size is n (Theorem 3.6.1), we conclude a $O(\log^3(n)n^{\frac{1}{2s+1}})$-sized version Sieve-SGD can achieve the minimax bound. □

6.6.2 Proof of Theorem 6.5 and Discussion

In this section we will show there is no b_n-size estimator with $b_n = o(n^{\frac{1}{2s+1}})$ that can achieve the minimax rate when estimating $f_\rho \in W(s, Q, \{\psi_j\})$. We first recall the metric entropy of a Sobolev ellipsoid satisfies (see [128, Chapter 5]):

$$\log \mathcal{N}(\delta; W(s), \|\cdot\|_2) \asymp \left(\frac{1}{\delta}\right)^{1/s} \quad \text{for all suitably small } \delta > 0 \tag{6.143}$$

We introduce the notation of δ-net of a decoder $D_n : \{0,1\}^{b_n} \to \mathcal{F}$ (under $\|\cdot\|_2$-norm)

$$\text{net}(\delta, b_n; D_n, \mathcal{F}) = \left\{ f \in \mathcal{F} \mid \exists s_n \in \{0,1\}^{b_n}, \text{ such that } \|f - D_n(s_n)\|_2 \le \delta \right\} \tag{6.144}$$

We use the notation $\text{net}(\delta, b_n)$ when it is clear what $D_n, \mathcal{F}$ we are refering to.

Proof of Theorem 3.6.5. Let M_n be any b_n-sized estimator with $b_n = o\left(n^{\frac{1}{2s+1}}\right)$. We denote its decoder function as D_n. We also choose a sequence c_n such that $b_n = o(c_n), c_n = o(n^{\frac{1}{2s+1}})$.

Now, we plug $\delta = c_n^{-s}$ into (6.143)

$$\log_2 \mathcal{N}(c_n^{-s}; W(s), \|\cdot\|_2) \asymp c_n \tag{6.145}$$

Because there are at most 2^{b_n} elements in $D_n(\{0,1\}^{b_n})$ (D_n is a known function), we also note

$$2^{b_n} \leq C 2^{c_n} \tag{6.146}$$

for some constant C. So we know $D_n(\{0,1\}^{b_n})$ cannot be a c_n^{-s}-cover of $W(s)$ for large enough n. In other words, for large n,

$$W(s) \setminus \operatorname{net}(c_n^{-s}, b_n) \tag{6.147}$$

is not an empty set.

Then we know

$$\begin{aligned}
\sup_{f_\rho \in W(s)} E[\|M_n((X_i,Y_i)_{i=1}^n) - f_\rho\|_2^2] &= \sup_{f_\rho \in W(s)} E[\|D_n(s_n) - f_\rho\|_2^2] \quad \text{where } s_n = E_n((X_i,Y_i)_{i=1}^n) \\
&\geq \sup_{f_\rho \in W(s) \setminus \operatorname{net}(c_n^{-s}, b_n)} E[\|D_n(s_n) - f_\rho\|_2^2] \\
&\geq \sup_{f_\rho \in W(s) \setminus \operatorname{net}(c_n^{-s}, b_n)} \inf_{s_n \in \{0,1\}^{b_n}} \|D_n(s_n) - f_\rho\|_2^2 \\
&\geq c_n^{-s}
\end{aligned} \tag{6.148}$$

Because this is true for any b_n-sized estimator M_n, we have

$$\begin{aligned}
&\inf_{M_n} \sup_{f_\rho \in W(s)} E[c_n^s \|M_n((X_i,Y_i)_{i=1}^n) - f_\rho\|_2^2] \geq 1 \\
\Rightarrow &\inf_{M_n} \sup_{f_\rho \in W(s)} E[n^{\frac{2s}{2s+1}} \|M_n((X_i,Y_i)_{i=1}^n) - f_\rho\|_2^2] \geq n^{\frac{2s}{2s+1}} c_n^{-s} \\
\Rightarrow &\lim_{n\to\infty} \inf_{M_n} \sup_{f_\rho \in W(s)} E[n^{\frac{2s}{2s+1}} \|M_n((X_i,Y_i)_{i=1}^n) - f_\rho\|_2^2] \to \infty
\end{aligned} \tag{6.149}$$

where the last line follows from the definition of c_n. □

Now we give a little bit of discussion on applying the above argument in parametric learning problem. Suppose we have a very simple model

$$Y = \theta X + \epsilon \tag{6.150}$$

where $X \in [0,1]$, $\theta \in [0,1]$, ϵ is uniformly-bounded, centered noise. Using the above argument, we can show that for any b_n-sized estimator M_n with $b_n = o(\log n)$, we have

$$\lim_{n\to\infty} \inf_{M_n} \sup_{\theta\in[0,1]} E\left[n\|M_n((X_i,Y_i)_{i=1}^n) - \theta\|_2^2\right] = \infty \tag{6.151}$$

This seems to suggest that for any estimator that uses a constant amount of memory, we cannot get a rate-optimal estimator of θ. This feels counter-intuitive because $\hat{\theta}_n$ is just a number. However we need to emphasize that in our formalization, the memory usage is counted in the unit of bit, i.e. one bit is $O(1)$. In practice we usually give the estimator substantial available memory (> 64 bits) and do not require extreme estimation accuracy. Thus our theory does not contradict the common belief that "parametric problem can be solved within $O(1)$ memory", because normally the unit of counting memory is a single stored real-number, rather than a single bit.

Note: Instead of the covering number of $W(s)$, the above result needs the following metric entropy result of the interval $[0,1]$ (see Prop.4.2.12 in [125])

$$\log \mathcal{N}(\delta; [0,1], \|\cdot\|_2) \asymp \log\left(\frac{1}{\delta}\right) \tag{6.152}$$

6.7 Constant Learning Rate $\gamma_n = \gamma_0$

Although in Theorem 3.6.1 and Theorem 3.6.3 we require the learning rate to decrease slowly ($\gamma_n = \Theta(n^{-\frac{1}{2s+1}})$) in order to achieve (near) optimal statistical guarantee, it is also of interest to see how the algorithm performs under other choices of learning rates. In this section we present a numerical example where the learning rate is set to be a constant γ_0 throughout the learning progress.

In Figure 6.2, we present the generalization ability of the estimator. The simulation setting is identical to that of Example 1 presented in the main text and we use the same training/testing data and identical processing ordering. The only difference is for Sieve-SGD we change the learning rate γ_n to be $\gamma_n = \gamma_0 = 0.5$.

The results using a constant learning rate is very similar to the those presented in Figure 3.1 (although numerically they are not identical). This means in this specific example we observe some adaptivity of the proposed algorithm w.r.t. the learning rate γ_n. However, it is not completely clear to us if such an adaptivity is a genuine property of Sieve SGD or it is a finite sample phenomenon.

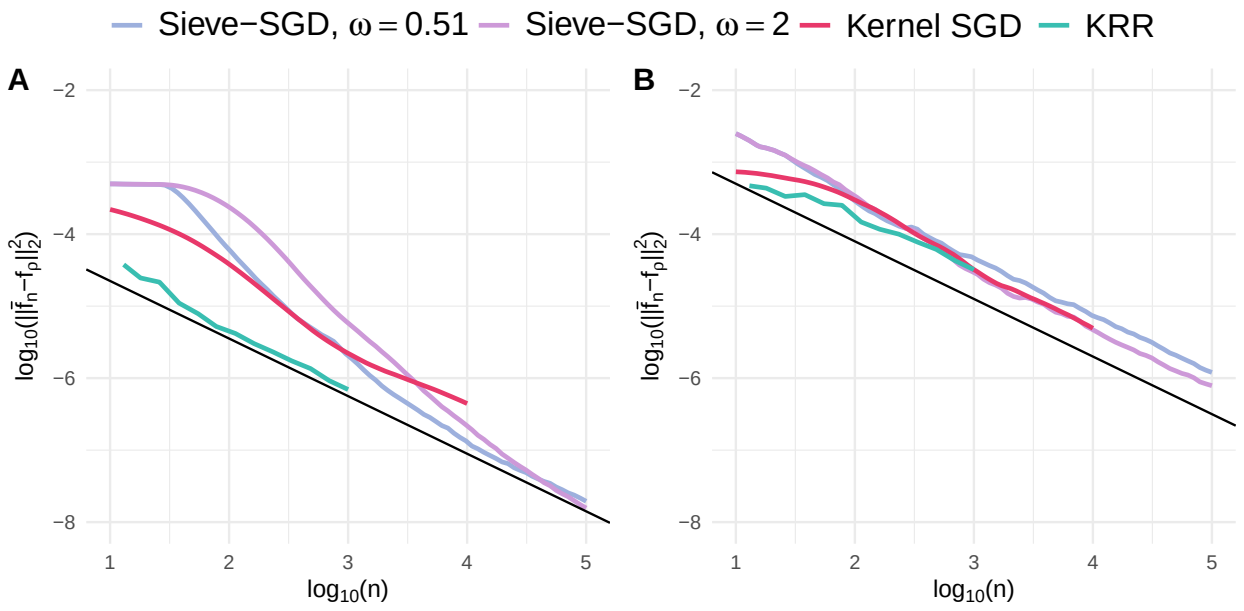

Figure 6.2: Simulation results with constant learning rate $\gamma_n = 0.5$. $\log_{10} \|\bar{f}_n - f_\rho\|_2^2$ against $\log_{10} n$. The Black line has slope $= -4/5$, which represents the optimal-rate. Each curve is calculated as the average of 100 repetitions. (**A**) X is uniformly distributed over $[0, 1]$. In this setting, SNR ~ 3. (**B**) X has a distribution in which ψ_j are not orthonormal. We present the results with very large noise, SNR~ 0.03. Due to different computational costs, we chose different maximum n for different methods.

Chapter 7

SUPPLEMENTARY MATERIALS FOR CHAPTER 4

7.1 More Numerical Examples and Method Implementation Discussion

7.1.1 Supplementary Numerical Results

In the main text, we present selected results from our simulation study. In this section we will provide more details together with another data generation setting that only has interaction terms.

In the simulation study, we have been using the oracle hyperparameters for each method under comparison, that is, those parameters that lead to minimal testing error. In Table 7.1 we present the hyper-parameters that are tuned for each method.

Table 7.1: Hyperparameters for each method

Method	Hyper-parameter
l_1-penalized sieve estimator (P-sieve)	number of basis functions, penalty parameter
gradient boosting machine (GBM)	number of iteration, tree depth
Gaussian kernel ridge regression	bandwidth, penalty parameter
least-square sieve estimator (LS-sieve)	number of basis functions
random forest (RF)	number of features randomly sampled, tree depth
highly adaptive lasso (HAL)	penalty parameter
sparse additive model (SpAdd)	number of basis functions, penalty parameter

In Figure 7.1 and 7.2, we present the simulation results under the same setting as in the

main text. The performance is evaluated using multiple metrics as in Figure 4.2 and 4.3.

We also present the simulation results from another data generating mechanism that does not have an additive component (results are in Figure 7.3 and 7.4). The data generating mechanism is defined as:

$$f^0_{interaction} = \sum_{k=1}^{D-1} Leg(2(\mathbf{x}^k - 0.5), 2) \cdot Leg(2(\mathbf{x}^{k+1} - 0.5), 3) \tag{7.1}$$

where the $Leg(x, j)$ function is the j-th Legendre polynomial

$$Leg(x, 2) = x, \quad Leg(x, 3) = (3x^2 - 1)/2 \tag{7.2}$$

This conditional mean has no main effects, meaning that

$$E[f^0_{interaction}(\mathbf{x}) \mid \mathbf{x}^k] = 0$$

for any $1 \leq k \leq d$. We can verify this by direct calculation (recall that $\mathbf{x} \sim Uniform([0, 1]^d)$). Although $f^0_{interaction}$ is a simple polynomial with nice smoothness properties, the lack of main effects (or additive components) messes up the performance of many methods. The almost zero testing R^2 of additive models demonstrates that in this setting they are no better than taking an unconditional mean of the outcome. Tree-based methods (gradient boosting and random forest) have more difficulties in this setting, especially when compared with their outstanding performance when the main effect components do exist. Tree-based methods cannot readily decide at which point to divide the feature space. For any binary cut only engaged with one feature, the mean of the outcome on one side of the division would be very similar to that of the other side under this specific setting.

7.1.2 Generating the Design Matrices

In this section we present more details on efficiently constructing the design matrix for multivariate sieve estimators. In the main text, we mention that the numerical implementation of sieve estimators is reduced to solving a least-square problem or a l_1-penalized optimization problem. In both cases we need to construct a design matrix $\hat{\Psi}$ whose elements are $\hat{\Psi}_{ij} = \psi_j(\mathbf{x}_i)$.

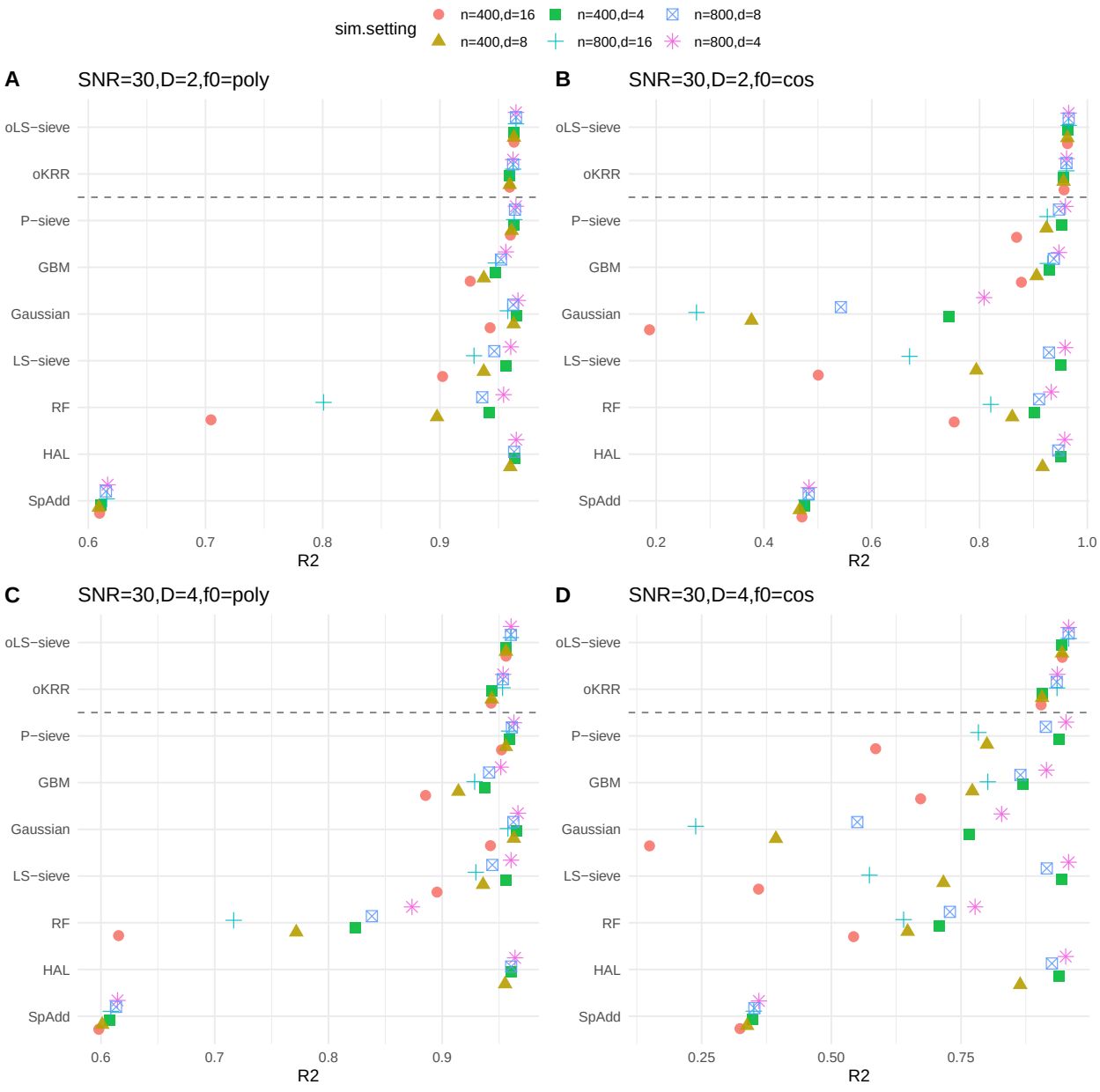

Figure 7.1: Simulation study results. SNR = 30.

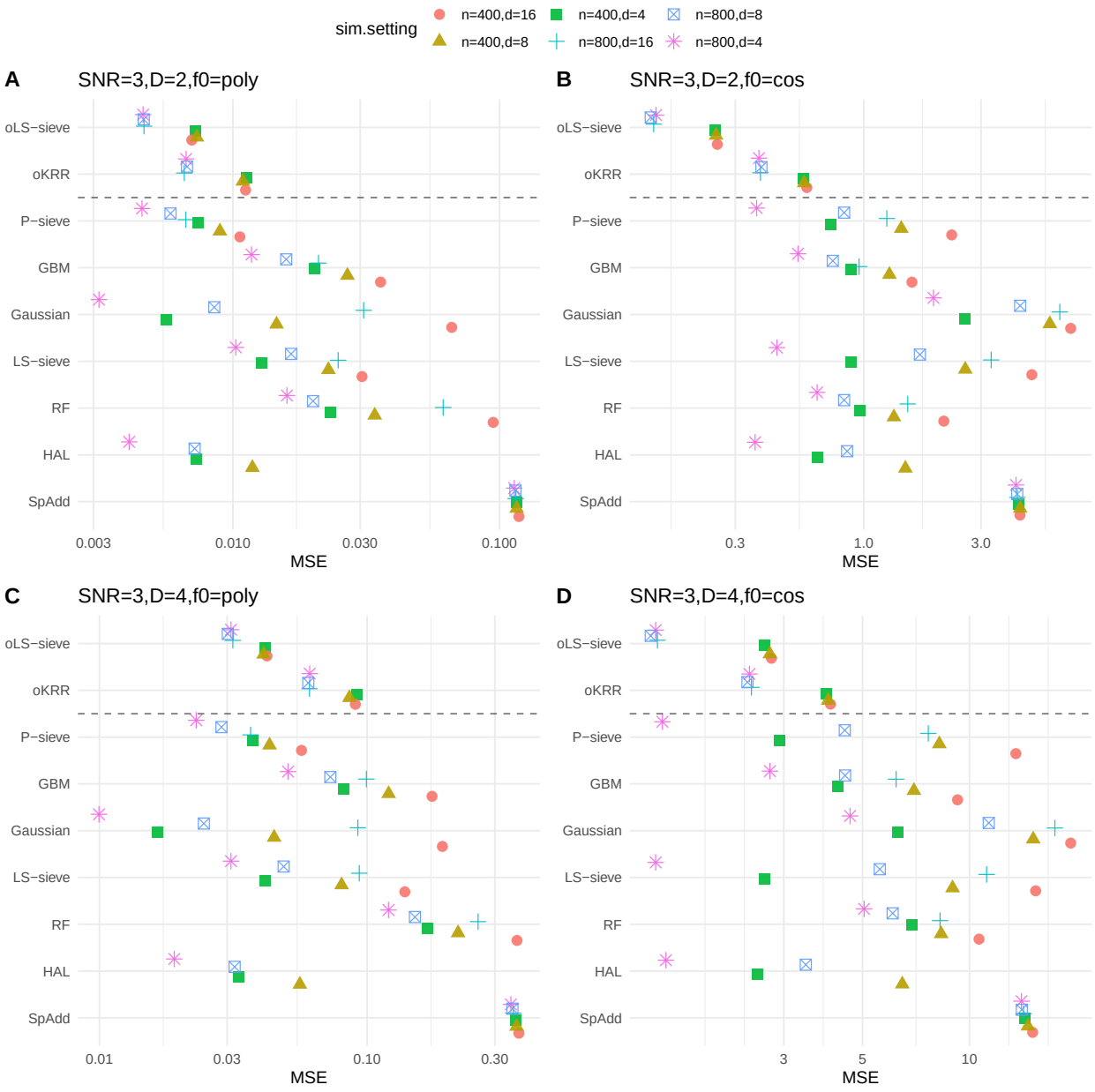

Figure 7.2: Simulation study results. SNR = 3.

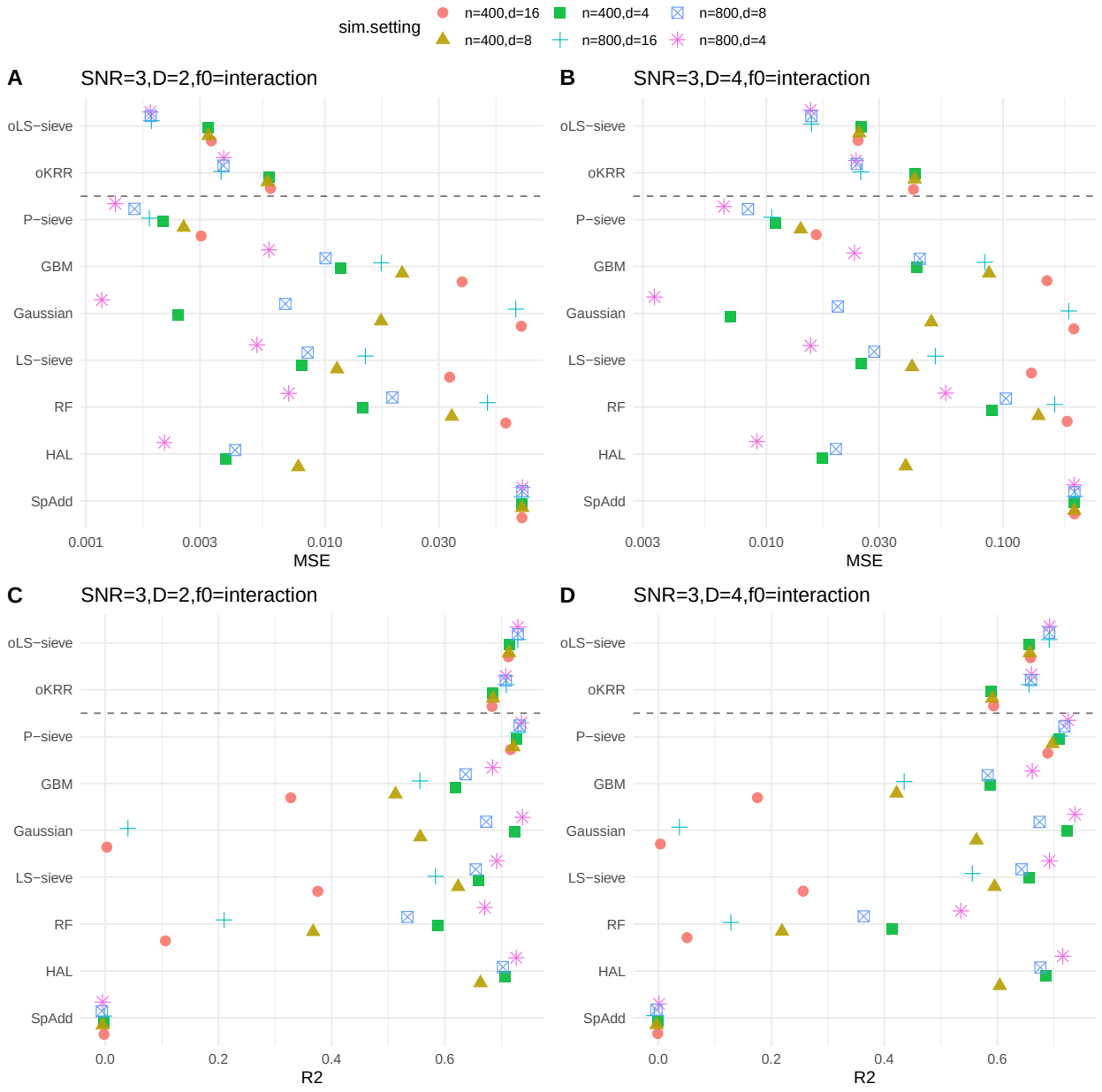

Figure 7.3: Additional settings, true regression function does not have main effect components. SNR = 3.

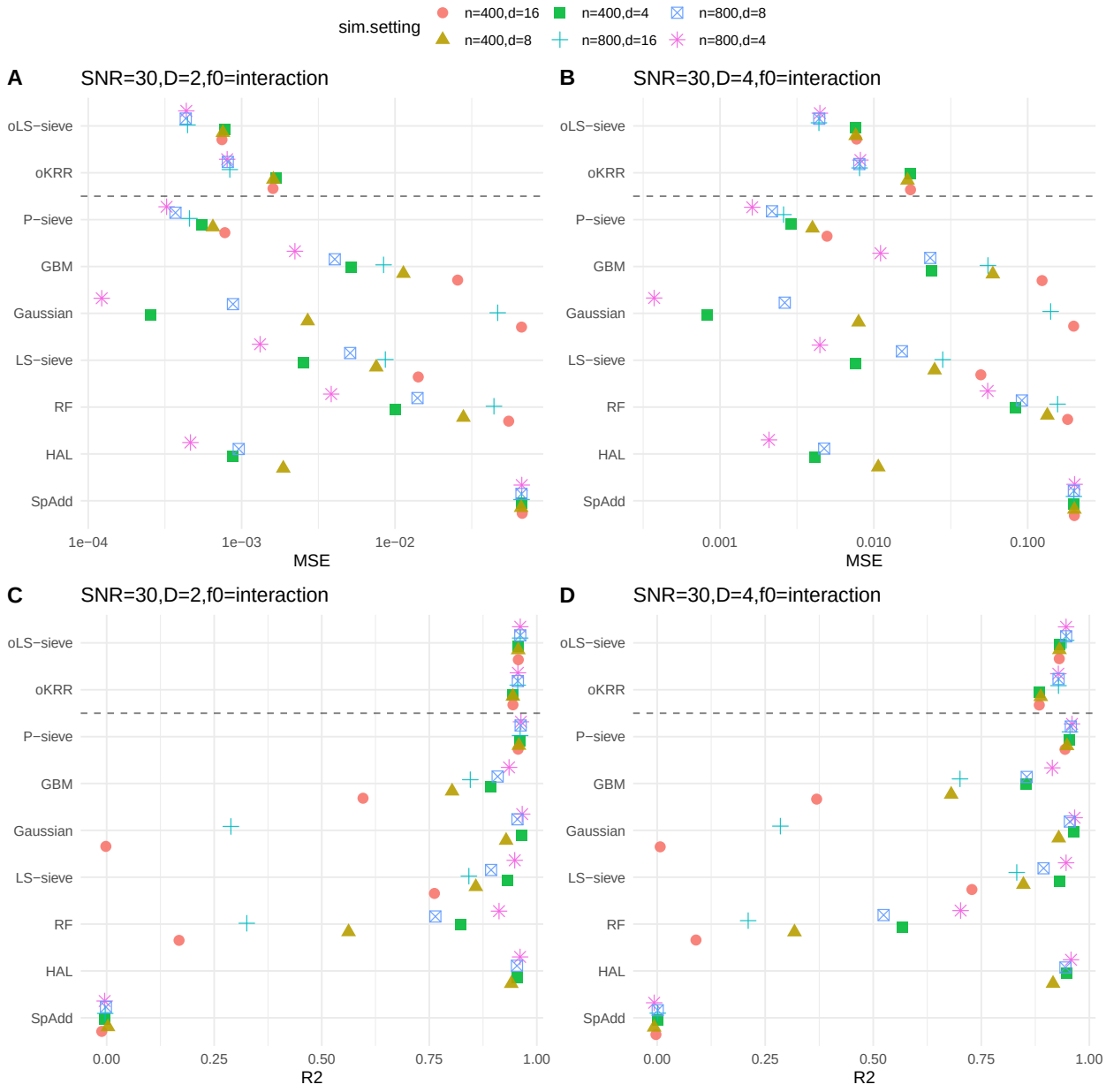

Figure 7.4: Additional settings, true regression function does not have main effect components. SNR = 30.

Given a set of multivariate product basis functions $\psi_{\mathbf{j}}(\mathbf{x}) = \prod_{k=1}^{d} \phi_{\mathbf{j}^k}(\mathbf{x}^k)$ indexed by $\mathbf{j} \in (\mathbb{N}^+)^d$, the unravelling rule $c_{\mathbf{j}} = \prod_{k=1}^{d} \mathbf{j}^k$ tells us how to sequentially use them to construct estimators. However, we only have a nonconstructive description of the elements in the unravelled sequence (ψ_j). To construct the design matrix $\hat{\Psi}$, we need to know the explicit form of each ψ_j. In practice, we need to first create an index matrix from which the algorithm identifies the analytical form of (ψ_j). For example, in the case $d = D' = 3$, we should construct an index matrix M of three columns (corresponding to the three dimensions). The first row has elements: $M_{11} = M_{12} = M_{13} = 1$, corresponding to the constant function $\psi_{(1,1,1)}$. And the following six rows are all 1 except for $M_{21} = M_{32} = M_{43} = 2$, and $M_{51} = M_{62} = M_{73} = 3$. They correspond to the second through seventh basis functions $\psi_{(2,1,1)}, \psi_{(1,2,1)}, \psi_{(1,1,2)}, \psi_{(3,1,1)}$, etc. By reading through this matrix, the algorithm can directly figure out the analytical form of the basis functions.

There are multiple ways to construct such an index matrix. When $D' = d$ (dense setting), one straightforward strategy is: 1) the user specifies the maximum index product C; 2) identify all the indices $\mathbf{j} \in (\mathbb{N}^+)^d$ whose maximum entry is smaller or equal to C; 3) sort the indices increasingly according to the index product $c_{\mathbf{j}} = \prod_{k=1}^{d} \mathbf{j}^k$; 4) keep only the earlier indices whose product is less than or equal to C. This algorithm is simple but is computationally wasteful. In step 2), C^d indices must be stored (this is memory intensive, even for moderate C and d). According to our theoretical results, we only need a subset of size $C \log^d(C)$. This issue is further exacerbated in the sparse case when $D' \ll d$. Therefore, we seek an alternative, computationally more efficient strategy, which includes some integer factorization, for generating the index matrix.

In Algorithm 7.1, we provide the details of the procedure. By factoring each positive integer as a product of D' numbers sequentially, we can fill out the matrix M. In the case when $d = D' = 3$, there is one row with row product equal to 1, two rows having row product equals to 2, and six rows having a product equal to 4. When D' is much smaller than d, as for the sparse sieve estimators, there should be many fewer rows corresponding to the same row product. The algorithm is presented below, followed by an example to explain some of

the steps.

Algorithm 7.1: Algorithm for generating the index matrix. For the definition of the $\tau_{D'}$ function mentioned in step *, see Definition 7.3.1. In ** step we use the notation from the `R` programming language to express our matrix update.

Set the maximum row product as `ProdMax`, feature dimension as `d`, working dimension as `D'`.

Define $C_m^d = d!/\{m!(d-m)!\}$, the combination number of "choosing m out of d elements".

`M` $\leftarrow$ An all 1 matrix of size $1 \times$ `d`.

FOR `Prod` $= 2$ TO `Prod` $=$ `ProdMax`

  Find all $\tau_{\texttt{D'}}$(`Prod`) ways to factorize `Prod` as a product of `D'` numbers. *

  Omit all values of "1" in the products and combine identical factorizations.

  `GreaterThanOne` $\leftarrow$ A list. Each element is an array, corresponding to one of the factorizations.

  FOR `i` $= 1$ TO `i` $=$ list length of `GreaterThanOne`

    `Gi` $\leftarrow$ The `i`-th element in `GreaterThanOne`.

    `m` $\leftarrow$ The length of the array `Gi`.

    `Position` $\leftarrow$ A matrix of size $C_{\texttt{m}}^{\texttt{d}} \times$ `m`.

      Each row corresponds to a unique way of choosing `m` elements from $\{1, 2, ..., \texttt{d}\}$.

    `NewIndexMatrix` $\leftarrow$ A matrix of size $C_{\texttt{m}}^{\texttt{d}} \times$ `d`. All elements are 1.

    FOR `j` $= 1$ TO `j` $=$ row number of `Position`

      `NewIndexMatrix[j, Position[j,]]` $\leftarrow$ `Gi` **

    ENDFOR

    `M` $\leftarrow$ Stack `M` above `NewIndexMatrix` to form a longer matrix.

  ENDFOR

ENDFOR

RETURN `M`.

We present some examples to better explain the compactly written algorithm above.

Let's assume $\texttt{d} = 3$, $\texttt{D'} = 2$. Suppose we are currently at $\texttt{Prod} = 6$ in the first layer of FOR loops. The $\tau_2(6) = 4$ ways to factorize 6 are:

$$6 = 6 \times 1 = 1 \times 6 = 2 \times 3 = 3 \times 2. \tag{7.3}$$

After the "Omit all values of 1 in the products and combine identical factorizations" step, we have three ways to factor 6 (the first two above are combined). Therefore, the `GreaterThanOne` list is

$$\texttt{GreaterThanOne} = \text{list}([6], [2, 3], [3, 2]). \tag{7.4}$$

The arrays in `GreaterThanOne` are of different lengths. Suppose we are at $\texttt{i} = 2$ in the second layer of the FOR loop. Then $\texttt{Gi} = [2, 3]$, $\texttt{m} = 2$. The `Position` matrix we constructed is

$$\texttt{Position} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 2 & 3 \end{bmatrix}. \tag{7.5}$$

This matrix specifies at which positions we are going to insert `Gi`. In the inner most FOR loop, we are going to update the all 1 matrix `NewIndexMatrix` using the information of `Position` and `Gi`: `Position`. In particular, `Gi`: `Position` specifies where to update, and `Gi` specifies what the elements are updated to. When $\texttt{i} = 2, \texttt{j} = 1$, we update the 1^{st} and 2^{nd} columns in the 1^{st} row of `NewIndexMatrix` to be $[2, 3]$, that is

$$\texttt{NewIndexMatrix}: \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \xrightarrow{\text{Update}} \begin{bmatrix} 2 & 3 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{7.6}$$

When $\texttt{i} = 2, \texttt{j} = 2$, we update the 1^{st} and 3^{rd} columns in the 2^{nd} row of `NewIndexMatrix` to be $[2, 3]$:

$$\texttt{NewIndexMatrix}: \begin{bmatrix} 2 & 3 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \xrightarrow{\text{Update}} \begin{bmatrix} 2 & 3 & 1 \\ 2 & 1 & 3 \\ 1 & 1 & 1 \end{bmatrix} \tag{7.7}$$

After looping through all the `j`, `i` and `Prod`, we have our desired index matrix `M`. Its first several rows are:

$$\text{row 1 to 10:} \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \\ 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \\ 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{bmatrix} \quad \text{row 11 to 20:} \begin{bmatrix} 2 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 2 \\ 5 & 1 & 2 \\ 1 & 5 & 1 \\ 1 & 1 & 5 \\ 6 & 1 & 1 \\ 1 & 6 & 1 \\ 1 & 1 & 6 \\ 2 & 3 & 1 \end{bmatrix} \quad \text{row 21 to 30:} \begin{bmatrix} 2 & 1 & 3 \\ 1 & 2 & 3 \\ 3 & 2 & 1 \\ 3 & 1 & 2 \\ 1 & 3 & 2 \\ 7 & 1 & 1 \\ 1 & 7 & 1 \\ 1 & 1 & 7 \\ 8 & 1 & 1 \\ 1 & 8 & 1 \end{bmatrix} \tag{7.8}$$

So we can read $\psi_1 = \psi_{(1,1,1)}$, $\psi_{11} = \psi_{(2,2,1)}$ and $\psi_{21} = \psi_{(2,1,3)}$, etc.

7.2 Product Kernels and Tensor Product Spaces

7.2.1 Univariate RKHS and Sobolev Ellipsoids

In Appendix 7.2, we will review the concept of Mercer kernels and reproducing kernel Hilbert spaces (RKHS). We will first engage with univariate RKHSs and their Sobolev ellipsoid representation in Appendix 7.2.1. By considering the tensor product kernel, we can extend our discussion to multivariate tensor product models (Appendix 7.2.2). Later in this section, we will arrive at some multivariate Sobolev ellipsoid models. These models can be seen as abstractions of the example function spaces (such as $S_1([0,1]^d)$) discussed in the main text.

There is a vast literature on univariate nonparametric regression problem. We list a few of them here: Sobolev space and smoothing spline estimators [126]; reproducing kernel Hilbert space and kernel ridge regression estimators [108]; Sobolev ellipsoid and sieve-type projection estimators [119]. These function spaces are closely related to each other: Sobolev spaces can sometimes be treat as a special case of RKHS and there is usually an equivalence

between a ball in an RKHS and a Sobolev ellipsoid. We will try to give a brief review of this part of nonparametric learning through some examples.

First we are going to present the concept of Mercer-kernels and their related reproducing kernel Hilbert spaces (on the real line).

Definition 7.2.1. *A symmetric bivariate function $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is positive semi-definite (PSD) if for any $n \geq 1$ and $(x_i)_{i=1}^n \subset \mathbb{R}$, the $n \times n$ matrix $\mathbb{K}$ whose elements are $\mathbb{K}_{ij} = k(x_i, x_j)$ is always a PSD matrix.*

A continuous, bounded, PSD kernel function k is called a Mercer kernel.

The following theorem [22] states the existence and uniqueness of a reproducing Hilbert space with respect to a Mercer kernel. The domain $\mathbb{R}$ in the following theorem can replaced by a subset such as $[0, 1]$ or $[0, +\infty)$.

Theorem 7.2.2. *For a Mercer Kernel $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, there exists an unique Hilbert Space $(\mathcal{H}_k, \langle \cdot, \cdot \rangle_k)$ of functions on $\mathbb{R}$ satisfying the following conditions. Let $k_x : z \mapsto k(x, z)$:*

1. *For all $x \in \mathbb{R}$, $k_x \in \mathcal{H}_k$.*

2. *The linear span of $\{k_x \mid x \in \mathbb{R}\}$ is dense (w.r.t $\| \cdot \|_k$) in $\mathcal{H}_k$.*

3. *For all $f \in \mathcal{H}_k, x \in \mathbb{R}$, $f(x) = \langle f, k_x \rangle_k$ (reproducing property).*

We call this Hilbert space the Reproducing kernel Hilbert space (RKHS) *associated with kernel k.*

Example 7.2.3. *The space $W_1([0, 1])$ is a RKHS with kernel*

$$k(s, t) = \frac{\cosh(\min(s, t)) \cosh(1 - \max(s, t))}{\sinh(1)} \tag{7.9}$$

For the proof, see Appendix A of [33] or [1]. The RKHS inner product for this kernel is defined as

$$\langle f, g \rangle_{W_1([0,1])} = \int_0^1 f(\tau) g(\tau) d\tau + \int_0^1 f'(\tau) g'(\tau) d\tau \tag{7.10}$$

The reproducing property reads as: for any $x \in [0,1]$ *and any* $f \in W_1([0,1])$

$$\begin{aligned} f(x) &= \langle f, k_x \rangle_{W_1([0,1])} \\ &= \int_0^1 f(\tau) k(x,\tau) d\tau + \int_0^1 f'(\tau) \frac{\partial}{\partial \tau} k(x,\tau) d\tau. \end{aligned} \tag{7.11}$$

Under mild conditions [109], a Mercer kernel has the following Mercer expansion.

$$k(s,t) = \sum_{j \in \mathcal{J}} \lambda_j \phi_j(s) \phi_j(t), \tag{7.12}$$

where $\mathcal{J}$ is an at most countably infinite index set. The eigenvalues λ_j are real numbers. The eigenfunctions (basis functions) $\{\phi_j\}$ can also be a complete basis of some L_2 space or the RKHS.

Although the majority of estimation procedures under RKHS models leverage the reproducing property, the method considered in this paper uses the feature maps directly (which is of a sieve nature). There have been studies showing that considering the problem from this perspective can give substantial computational advantage over standard kernel methods [142, 143]. In this manuscript we will also show how sieve estimators can be more easily adapted to employ variable selection and can additionally be adaptive to dimension. Now, we present the important connection between a RKHS and a Sobolev ellipsoid established in the literature (e.g., p.37,Theorem 4 in [22]).

Theorem 7.2.4. *Under mild conditions, the Hilbert space* $\mathcal{H}_k$ *of the kernel* k *(defined in Theorem 7.2.2) is identical – same function class with the same inner product – to the following Hilbert space* $\mathbb{H}_k$*.*

$$\mathbb{H}_k = \left\{ f \mid f = \sum_{j=1}^{\infty} a_j \phi_j \quad \textit{with} \quad \sum_{j=1}^{\infty} a_j^2 \lambda_j^{-1} < \infty \right\} \tag{7.13}$$

The RKHS inner product can be explicitly written as:

$$\langle f, g \rangle_k = \sum_{j=1}^{\infty} \lambda_j^{-1} a_j b_j \tag{7.14}$$

for $f = \sum_j a_j \phi_j$*, and* $g = \sum_j b_j \phi_j$*. The functions* ϕ_j *and real numbers* λ_j *are the eigensystem in the Mercer expansion* (7.12) *(assuming* $\mathcal{J} = \mathbb{N}^+$*).*

Example 7.2.5. *The reproducing kernel for $W_1([0,1])$ has the following Mercer expansion:*

$$k(s,t) = \sum_{j=1}^{\infty} \phi_j(s)\phi_j(t), \tag{7.15}$$

with

$$\begin{aligned} \lambda_1 = 1, \quad \phi_1(x) = 1, \\ \lambda_j = \frac{1}{1 + ((n-1)\pi)^2}, \quad \phi_j(x) = \sqrt{2}\cos((n-1)\pi x) \text{ for } j \geq 2. \end{aligned} \tag{7.16}$$

Therefore, we also have the following characterization of a ball in $W_1([0,1])$:

$$\{f \in W_1([0,1]) \mid \|f\|_{W_1}^2 \leq Q^2\} = \left\{ f = \sum_{j=1}^{\infty} a_j \phi_j \quad \text{with} \quad \sum_{j=1}^{\infty} a_j^2 \lambda_j^{-1} \leq Q^2 \right\} \tag{7.17}$$

To summarize, a ball in a RKHS is a Sobolev ellipsoid.

7.2.2 Multivariate RKHS and Sobolev Ellipsoids

Given a univariate RKHS, one of the most naturally related multivariate RKHS is the one corresponding to the product kernel. This also happens to correspond to one of the most commonly used multivariate kernels in practice: The multivariate Gaussian kernel which is a product of univariate Gaussian kernels.

Definition 7.2.6. *Given a univariate Mercer kernel $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, we define its (natural, d-dimensional) product kernel $k^d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to be:*

$$k^d(\mathbf{s}, \mathbf{t}) = \prod_{j=1}^{d} k(\mathbf{s}^j, \mathbf{t}^j). \tag{7.18}$$

We can also define the RKHS of k^d using the fact that k^d is also a Mercer kernel (Proposition 12.31 of [128]). Typical elements in this multivariate RKHS take the following form:

$$f(\mathbf{x}) = \sum_{l=1}^{m} \prod_{k=1}^{d} f_{kl}(\mathbf{x}^k), \text{ with } f_{kl} \in \mathcal{H}_k. \tag{7.19}$$

There are multiple ways to engage with an element in $\mathcal{H}_{k^d}$ and its inner product. One way, as presented above, is using the property that $\mathcal{H}_{k^d}$ is a tensor product Hilbert space

of d univariate Hilbert spaces. This would lead to the following characterization of its inner product.

Proposition 7.2.7. *The RKHS for k^d, $\mathcal{H}_{k^d}$, is equipped with the inner product:*

$$\langle h, g\rangle_{k^d} = \sum_{i=1}^{n}\sum_{l=1}^{m}\prod_{j=1}^{d}\langle h_{ij}, g_{lj}\rangle_k \tag{7.20}$$

for $h(\mathbf{x}) = \sum_{i=1}^{n}\prod_{j=1}^{d} h_{ij}(\mathbf{x}^j)$, $g(\mathbf{x}) = \sum_{l=1}^{m}\prod_{j=1}^{d} g_{lj}(\mathbf{x}^j)$. The component functions h_{ij}, g_{lj} all belong to the univariate RKHS $\mathcal{H}_k$.

.

Alternatively, we can also consider the basis expansion form of the functions in $\mathcal{H}_{k^d}$ (similar to Theorem 7.2.4). The tensor product kernel k^d has the following Mercer expansion (which can be formally verified using its Mercer expansion):

$$k^d(\mathbf{s}, \mathbf{t}) = \sum_{\mathbf{j}\in(\mathbb{N}^+)^d}\prod_{k=1}^{d}\lambda_{\mathbf{j}^k}\psi_{\mathbf{j}}(\mathbf{s})\psi_{\mathbf{j}}(\mathbf{t}), \text{ with } \lambda_{\mathbf{j}^k} \in \mathbb{R}. \tag{7.21}$$

We have the following equivalent characterization:

Proposition 7.2.8. *The inner product presented in Proposition 7.2.7 is equivalent to the following one expressed in basis expansion form:*

$$\langle h, g\rangle_{k^d} = \sum_{\mathbf{j}\in(\mathbb{N}^+)^d}\left(\prod_{k=1}^{d}\lambda_{\mathbf{j}^k}\right)^{-1} h_{\mathbf{j}}g_{\mathbf{j}} \tag{7.22}$$

for h, g in the multivariate RKHS $\mathcal{H}_{k^d}$ with the basis expansion $h = \sum_{\mathbf{j}\in(\mathbb{N}^+)^d} h_{\mathbf{j}}\psi_{\mathbf{j}}$, $g = \sum_{\mathbf{j}\in(\mathbb{N}^+)^d} g_{\mathbf{j}}\psi_{\mathbf{j}}$. The multivariate basis $\psi_{\mathbf{j}}(\mathbf{x}) = \prod_{k=1}^{d}\phi_{\mathbf{j}^k}(\mathbf{x}^{\mathbf{j}^k})$ is the product of the eigenfunctions (as defined in (7.12)) of the univariate kernel k.

Example 7.2.9. *The natural d-dimensional tensor product extension of $W_1([0,1])$ space is the RKHS of the kernel:*

$$\begin{aligned} k^d(\mathbf{s}, \mathbf{t}) &= \prod_{m=1}^{d} k(\mathbf{s}^m, \mathbf{t}^m) = \prod_{m=1}^{d} k(\mathbf{s}^m, \mathbf{t}^m) \\ &= \{\sinh(1)\}^{-d}\prod_{m=1}^{d}\cosh(\min(\mathbf{s}^m, \mathbf{t}^m))\cosh(1 - \max(\mathbf{s}^m, \mathbf{t}^m)) \end{aligned} \tag{7.23}$$

The inner product, according to Proposition 7.2.7, can be explicitly written as:

$$\begin{aligned}\langle h, g\rangle_{k^d} &= \sum_{k=1}^{n}\sum_{l=1}^{m}\prod_{j=1}^{d}\langle h_{kj}, g_{lj}\rangle_{W_1([0,1])} \\ &= \sum_{k=1}^{n}\sum_{l=1}^{m}\prod_{j=1}^{d}\left(\int_0^1 h_{kj}(\tau)g_{lj}(\tau)d\tau + \int_0^1 h'_{kj}(\tau)g'_{lj}(\tau)d\tau\right)\end{aligned} \tag{7.24}$$

for $h(\mathbf{x}) = \sum_{k=1}^{n}\prod_{j=1}^{d} h_{kj}(\mathbf{x}^j)$, $g(\mathbf{x}) = \sum_{l=1}^{m}\prod_{j=1}^{d} g_{lj}(\mathbf{x}^j)$. *The component functions* h_{kj}, g_{lj} *all belong to* $W_1([0,1])$. *Then the RKHS-norm (induced by the inner product) for a function* $h \in \mathcal{H}_{k^d}$ *is:*

$$\begin{aligned}\|h\|_{k^d} &= \sum_{k=1}^{n}\sum_{l=1}^{n}\prod_{j=1}^{d}\left(\int_0^1 h_{kj}(\tau)h_{lj}(\tau)d\tau + \int_0^1 h'_{kj}(\tau)h'_{lj}(\tau)d\tau\right) \\ &\overset{(1)}{=} \sum_{\|\mathbf{a}\|_\infty \le 1}\|D^{\mathbf{a}}h\|^2_{L_2([0,1]^d)}.\end{aligned} \tag{7.25}$$

The above step (1) can be checked directly (and using Fubini's theorem). We present the calculation for a simple case where $h(\mathbf{x}) = \prod_{j=1}^{d} h_j(\mathbf{x}^j)$ *and* $d = 2$*:*

$$\begin{aligned}\|h\|_{k^2} &= \left(\int_0^1 h_1^2(\tau_1)d\tau_1\right)\left(\int_0^1 h_2^2(\tau_2)d\tau_2\right) + \left(\int_0^1 (h_1(\tau_1))^2 d\tau_1\right)\left(\int_0^1 (h_2'(\tau_2))^2 d\tau_2\right) + \\ &\left(\int_0^1 (h_1'(\tau_1))^2 d\tau_1\right)\left(\int_0^1 h_2^2(\tau_2)d\tau_2\right) + \left(\int_0^1 (h_1'(\tau_1))^2 d\tau_1\right)\left(\int_0^1 (h_2'(\tau_2))^2 d\tau_2\right) \\ &= \int_{[0,1]^2} h^2(\tau_1,\tau_2)d\tau_1 d\tau_2 + \int_{[0,1]^2}\left(\frac{\partial}{\partial\tau_2}h(\tau_1,\tau_2)\right)^2 d\tau_1 d\tau_2 + \\ &\int_{[0,1]^2}\left(\frac{\partial}{\partial\tau_1}h(\tau_1,\tau_2)\right)^2 d\tau_1 d\tau_2 + \int_{[0,1]^2}\left(\frac{\partial^2}{\partial\tau_1\partial\tau_2}h(\tau_1,\tau_2)\right)^2 d\tau_1 d\tau_2\end{aligned} \tag{7.26}$$

Briefly, the $W_1([0,1])$ *space is an example of a univariate RKHS; the* $S_1([0,1])$ *space, when equipped with a proper inner product, is the tensor product extension of* $W_1([0,1])$. *Moreover, Proposition 7.2.8 implies an equivalent way to express the RKHS inner product and its induced norm. Specifically, we know that*

$$\sum_{\|\mathbf{a}\|_\infty \le 1}\|D^{\mathbf{a}}h\|^2_{L_2([0,1]^d)} = \|h\|_{k^d} = \sum_{\mathbf{j}\in(\mathbb{N}^+)^d}\left(\prod_{k=1}^{d}\mathbf{j}^k\right)^2\beta_{\mathbf{j}}^2 \tag{7.27}$$

for $h = \sum_{\mathbf{j}\in(\mathbb{N}^+)^d} \beta_{\mathbf{j}}\psi_{\mathbf{j}}$. *The multivariate basis* $\psi_{\mathbf{j}} = \prod_{k=1}^{d} \phi_{\mathbf{j}^k}$ *is the product of the cosine functions (defined in* (7.16)*).*

In the rest of the paper, we will switch from concrete example spaces to more abstract Sobolev ellipsoid-type spaces. The (univariate) Sobolev ellipsoid has been a benchmark model in the literature of sieve estimators: We just showed how it relates to multivariate spaces. In the multivariate case, we will be engaging with a true function f^0 that belongs to the multivariate Sobolev "ellipsoid":

$$f^0 \in \left\{ f = \sum_{\mathbf{j}\in(\mathbb{N}^+)^d} \beta_{\mathbf{j}}\psi_{\mathbf{j}} \;\middle|\; \sum_{\mathbf{j}\in(\mathbb{N}^+)^d} \left(\prod_{k=1}^{d} \mathbf{j}^k\right)^{2s} \beta_{\mathbf{j}}^2 \leq Q^2 \right\}. \tag{7.28}$$

for some product basis $\psi_{\mathbf{j}}$. In particular, we assume the regression function can be expanded as an infinite linear combination of a set of basis functions $\psi_{\mathbf{j}}$ indexed by d-tuples. At the same time, we require $\beta_{\mathbf{j}}$ to converge to zero at a fast enough rate as the product of index $\mathbf{j}$ goes to infinity. The function space in (7.28) is the same as a ball in some multivariate RKHS (as illustrated in Example 7.2.9). We also introduced another parameter s that determines the decay rate of $\beta_{\mathbf{j}}$, which is often interpreted as a smoothness parameter ([126], Chapter 2).

7.3 Unravelling and Approximation Results

7.3.1 Magnitude of Unravelled Series

In this section we will first quantify the asymptotic behavior of unravelled series c_j, which is depicted in the right panel of Figure 4.1. We will use these results to reduce Sobolev ellipsoids indexed by D-tuples (7.28) to those indexed by natural numbers. This will directly lead to some useful approximation results in multivariate tensor product spaces.

In general, we cannot give a closed form for the unravelled sequence C_j as function of j (in Algorithm 7.1we gave an algorithm to generate finitely many elements). However, it is still possible to derive some results on the magnitude of c_j as a function of j. To this end, we first introduce the concept of a divisor function.

Definition 7.3.1. *We use $\tau_D(\cdot) : \mathbb{N}^+ \to \mathbb{N}^+$ to denote the D-th divisor function, which counts the number of unique ways to factor n as a product of D positive integers (where order matters). Formally,*

$$\tau_D(n) = \sum_{\substack{(\mathbf{j}^1,...,\mathbf{j}^D)\in(\mathbb{N}^+)^D \\ \prod_{k=1}^{D}\mathbf{j}^k=n}} 1 \tag{7.29}$$

The divisor function τ_D distinguishes the order of factorization: For example $\tau_2(4) = 3$ because there are 3 ways to write 4 as a product of 2 numbers: $4 = 1 \times 4 = 4 \times 1 = 2 \times 2$. In the exposition of this section, we also need to engage with the following partial sum of divisor functions.

Definition 7.3.2. *We define the sequence $T_D(x)$ to be the sum of the D-divisor function evaluated at the first $\lfloor x \rfloor$ positive natural numbers, that is*

$$T_D(x) = \sum_{n\leq x} \tau_D(n). \tag{7.30}$$

Clearly, $T_D(x)$ is the number of D-tuples $\mathbf{j} = (\mathbf{j}^1, .., \mathbf{j}^D) \in (\mathbb{N}^+)^D$ with $\prod_{k=1}^{D}\mathbf{j}^k \leq x$. The number x is not necessarily an integer: the summation index $n \leq x$ should be interpreted as $\{1, 2, ..., \lfloor x \rfloor\}$.

The first several elements in c_j (depicted in Figure 4.1) are $1, 2, 2, 3, 3, 4, 4, 4, ...$ As our readers may notice, each natural number n shows up exactly $\tau_2(n)$ times: if we know (on average) how many ways there are to factor a positive integer, we can sketch the general magnitude of the unravelled sequence as well. The following lemma formalizes such an idea.

Lemma 7.3.3. *Define $c_{\mathbf{j}} = \prod_{k=1}^{D}\mathbf{j}^k$ as a function on the D-tuple $\mathbf{j} = (\mathbf{j}^1, ..., \mathbf{j}^D) \in (\mathbb{N}^+)^D$. Let c_j be the $c_{\mathbf{j}}$-unravelling sequence of $c_{\mathbf{j}}$ (see definition in Section 4.6). Then, for D fixed, we know its asymptotic magnitude is:*

$$c_j = \Theta\left(j \log^{-(D-1)} j\right) \tag{7.31}$$

Proof. All the elements of c_j are positive integers since they are products of positive integers. And every positive integer shows up in c_j at least once. We also observe that there are

repeated elements in c_j: For any positive integer m, it shows up exactly $\tau_D(m)$ times in the sequence c_j.

To determine the increase rate of c_j, it is enough to determine the largest b_j such that

$$T_D(b_j) = \sum_{m=1}^{b_j} \tau_D(m) \leq j. \tag{7.32}$$

The unravelling sequence c_j increases at the same rate as b_j. To quantify the summation on the LHS, we need to use the following result from number theory:

$$\sum_{m=1}^{x} \tau_D(m) = \frac{\log^{D-1} x}{(D-1)!} x + O\left(x \log^{D-2} x\right), \tag{7.33}$$

where the big O notation indicates $x \to \infty$ (but D is fixed). If we divide both sides by x, then we know: on average, there are $(\log x)^{D-1}$ ways to factorize a natural number into a product of D natural numbers. This result has been established in the literature of number theory, we give more discussion and references in Appendix 7.5. For the special case when $D = 2$, there are sharper results available, e.g. Theorem 3.2 in [117].

Let $b_j = \lfloor (D-1)! j \log^{-(D-1)} j \rfloor$. Plug b_j into (7.33):

$$\begin{aligned} \sum_{m=1}^{b_j} \tau_D(m) &= b_j \log^{D-1} b_j + O\left(b_j \log^{D-2} b_j\right) \\ &= \Theta\left(j(\log j)^{-(D-1)} \log^{D-1}\left\{j(\log j)^{-(D-1)}\right\}\right) = \Theta(j) \end{aligned} \tag{7.34}$$

It is direct to check that if $b_j = q_j j \log^{-(D-1)} j$ for any positive $q_j \to \infty$, $b_j \log^{D-1} b_j$ would diverge at a rate faster than j. So we know the largest b_j we can take is of order $j \log^{-(D-1)} j$, which concludes our proof. □

Corollary 7.3.4. *Let $c_{\mathbf{j}} = \prod_{k=1}^{D} (\mathbf{j}^k)^s$ be a function defined on the D-tuple $\mathbf{j} = (\mathbf{j}^1, ..., \mathbf{j}^D) \in (\mathbb{N}^+)^D$ for some $s > 0$. Let c_j be the $c_{\mathbf{j}}$-unravelling sequence of $c_{\mathbf{j}}$. Then we know*

$$c_j = \Theta\left(\left(j \log^{-(D-1)} j\right)^s\right) \tag{7.35}$$

The next theorem is the main result in this section, which uses Lemma 7.3.3 or Corollary 7.3.4.

Theorem 7.3.5. *Let $W(s, Q, \{\psi_{\mathbf{j}}\})$ be the multivariate product Sobolev space:*

$$W\left(s, Q, \{\psi_{\mathbf{j}}\}\right) = \left\{ f = \sum_{\mathbf{j} \in (\mathbb{N}^+)^D} \beta_{\mathbf{j}} \psi_{\mathbf{j}}, \text{ for some } \beta_{\mathbf{j}} \in \mathbb{R} \;\middle|\; \sum_{\mathbf{j} \in (\mathbb{N}^+)^D} c_{\mathbf{j}}^{2s} \beta_{\mathbf{j}}^2 \leq Q^2 \right\}, \tag{7.36}$$

where $c_{\mathbf{j}} = \prod_{k=1}^{D} \mathbf{j}^k$ for $\mathbf{j} = (\mathbf{j}^1, ..., \mathbf{j}^D) \in (\mathbb{N}^+)^D$. Denote (ψ_j) as the $c_{\mathbf{j}}$-unravelling sequence of $\{\psi_{\mathbf{j}}\}$.

Then there exists two constants $C_i(s, D)$, $i \in \{1, 2\}$ such that

$$\begin{aligned} & \left\{ f = \sum_{j=1}^{\infty} \beta_j \psi_j, \text{ for some } \beta_j \in \mathbb{R} \;\middle|\; \sum_{j=1}^{\infty} \left(\frac{j}{\log^{D-1} j \vee 1} \right)^{2s} \beta_j^2 \leq C_1(s, D) Q^2 \right\} \\ \subset \quad & W(s, Q, \{\psi_{\mathbf{j}}\}) \\ \subset & \left\{ f = \sum_{j=1}^{\infty} \beta_j \psi_j, \text{ for some } \beta_j \in \mathbb{R} \;\middle|\; \sum_{j=1}^{\infty} \left(\frac{j}{\log^{D-1} j \vee 1} \right)^{2s} \beta_j^2 \leq C_2(s, D) Q^2 \right\}. \end{aligned} \tag{7.37}$$

In plain[er] language, Theorem 7.3.5 states that: The multivariate function space $W(s, Q, \{\psi_{\mathbf{j}}\})$ can be sandwiched between two formally simpler function spaces. These "bread" function spaces in (7.37) are still multivariate function spaces, but the basis functions (ψ_j) are listed in a sequence. In contrast, $W(s, Q, \{\psi_{\mathbf{j}}\})$ has basis functions indexed by D-tuples.

Proof. The multivariate ellipsoid $W(s, Q, \{\psi_{\mathbf{j}}\})$ is exactly the same space as:

$$\left\{ f = \sum_{j=1}^{\infty} \beta_j \psi_j \;\middle|\; \sum_{j=1}^{\infty} c_j^{2s} \beta_j^2 \leq Q^2 \right\}, \tag{7.38}$$

where c_j, β_j, ψ_j are the $c_{\mathbf{j}}$-unravelling sequences of $c_{\mathbf{j}}, \beta_{\mathbf{j}}, \psi_{\mathbf{j}}$, respectively. In this step we performed nothing but a change of notation.

According to Corollary 7.3.4, c_j is asymptotically of the same order as $\left(j \log^{-(D-1)} j \right)^{2s}$ as $j \to \infty$. Define $b_j = \left(\frac{j}{\log^{D-1} j \vee 1} \right)^{2s}$, then we know that there exist constants C_1, C_2 (that only depends on s, and D) such that $C_1 b_j \leq c_j \leq C_2 b_j$ for all $j \in \mathbb{N}^+$. Plugging this in to (7.38) concludes our proof. □

7.3.2 Approximation in Dense Tensor Product Models

In this section, we will use the results in Theorem 7.3.5 to derive some approximation results that are crucial to understand the performance of our sieve estimators. Before we go into more detail, we provide some intuitive discussion of why Theorem 7.3.5 can simplify our analysis. Let's denote the three function spaces in (7.37) as W_1, W_2 and W_3 ($W_1 \subset W_2 \subset W_3$). To study the problem of approximation/estimation of functions in W_2, it is equivalent – up to a constant – to study the corresponding problems in W_1 or W_3. The regression problem under the assumption $f^0 \in W_2$ is easier than assuming $f^0 \in W_3$ but harder than $f^0 \in W_1$. Therefore the generalization error of any estimators for truth $f^0 \in W_2$ should be of the same order as $f^0 \in W_1$ or W_3. Similar statements also hold for minimax rates analysis.

Ellipsoids related to a (univariate) series c_j can be treated much more directly than those related to the D-tuple $c_{\mathbf{j}}$. For readers who are familiar with classical projection estimators (e.g. [119]), the following approximation results may be familiar.

Before we engage with those results, we give a bit more notation: In the remainder of our discussion in the appendix, we will use $\mathcal{X} \subset \mathbb{R}$ to denote a subset of real line and use ν to denote a (finite) Borel measure on $\mathcal{X}$. We do not need to specify either $\mathcal{X}$ or ν accurately: Often we just need $\mathcal{X}^d$ to be large enough to cover the support of feature distribution ρ_X, and in many important cases $\nu =$ uniform measure is enough for our purposes.

Lemma 7.3.6. *Suppose function f^* has the expansion $f^* = \sum_{j=1}^{\infty} \beta_j^* \psi_j$ with respect to a set of ν-orthonormal system, i.e. $\langle \psi_j, \psi_i \rangle_{L_2(\nu)} = \delta_{ij}$. Assume $\|\psi_j\|_\infty \leq M$ for all j. If the expansion coefficients satisfy the following ellipsoid-type condition:*

$$f^* = \sum_{j=1}^{\infty} \beta_j^* \psi_j \in \left\{ f = \sum_{j=1}^{\infty} \beta_j \psi_j \in L_2(\nu) \;\; | \;\; \sum_{j=1}^{\infty} \left(\frac{j}{\log^{D-1} j \vee 1} \right)^{2s} \beta_j^2 \leq Q^2 \right\}, \tag{7.39}$$

with some $s > 1/2$. Then the sequence of functions

$$f_n^* = \sum_{j=1}^{J_n} \beta_{nj}^* \psi_j \text{ with } J_n = \lfloor (\log^{D-1} n)^{2s/(2s+1)} n^{1/(2s+1)} \rfloor, \quad n = 2, 3, ... \tag{7.40}$$

satisfy the following:

- *There is a constant $C(M, s, D, Q)$, such that for any n:*

$$\|f_n^*\|_\infty \leq C(M, s, D, Q) \tag{7.41}$$

- *For any measure ρ_X that is absolute continuous to ν with a bounded density:*

$$\begin{aligned} \|f_n^* - f^*\|_{2,\rho_X}^2 &= \int \{f_n^*(\tau) - f^*(\tau)\}^2 \, d\rho_X \\ &\leq C(s, D, \rho_X, Q) \left(\frac{\log^{D-1} n}{n}\right)^{\frac{2s}{2s+1}} \end{aligned} \tag{7.42}$$

Proof. • We first prove the uniform bound in the $\|\cdot\|_\infty$-norm. According to our discussion Appendix 7.2.2, a Sobolev-ellipsoid can be seen as a ball in an RKHS. That is, the functions f^*, f_n^* all belong to an RKHS with reproducing kernel

$$k(s,t) = \sum_{j=1}^{\infty} \lambda_j \psi_j(s) \psi_j(t), \tag{7.43}$$

where $\lambda_j = \left(\frac{\log^{D-1} j \vee 1}{j}\right)^{2s}$. Denote the RKHS inner product as $\langle \cdot, \cdot \rangle_k$:

$$\begin{aligned} \|f_n^*\|_\infty &= \sup_{x \in \mathcal{X}} f_n^*(x) = \sup_{x \in \mathcal{X}} \langle f_n^*, k(x, \cdot) \rangle_k \\ &\leq \|f_n^*\|_k \sup_{x \in \mathcal{X}} \|k(x, \cdot)\|_k \\ &\overset{(1)}{\leq} QC(M, s, D). \end{aligned} \tag{7.44}$$

In step (1), we need the explicit representation of the RKHS norm (Theorem 7.2.4). The RKHS norm of kernel k (centered at x) is

$$\|k(x, \cdot)\|_k = \sum_{j=1}^{\infty} (\lambda_j \psi_j(x))^2 / \lambda_j \leq M^2 \sum_{j=1}^{\infty} \lambda_j = C(M, s, D)$$

- Next we prove the bound in ρ_X-2-norm. Let U denote a bound on the density of ρ_X (with respect to ν):

$$\begin{aligned} \|f_n^* - f^*\|_{2,\rho_X}^2 &\leq U \|f_n^* - f^*\|_{2,\nu}^2 = \sum_{J_n+1}^{\infty} (\beta_j^*)^2 \\ &\leq \lambda_{J_n} \sum_{J_n+1}^{\infty} (\beta_j^*)^2 / \lambda_j \leq \lambda_{J_n} Q^2 \end{aligned} \tag{7.45}$$

We just need to determine the magnitude of λ_{J_n}:

$$\begin{aligned}\lambda_{J_n} &\leq cJ_n^{-2s}(\log^{D-1} J_n)^{2s} \\ &= c\left\{(\log^{D-1} n)^{\frac{2s}{2s+1}} n^{1/(2s+1)}\right\}^{-2s}\left[\log^{D-1}\left\{(\log^{D-1} n)^{\frac{2s}{2s+1}} n^{1/(2s+1)}\right\}\right]^{2s} \\ &\leq C(s,D)n^{-\frac{2s}{2s+1}}(\log^{D-1} n)^{-\frac{4s^2}{2s+1}+2s} = C(s,D)\left(\frac{\log^{D-1} n}{n}\right)^{\frac{2s}{2s+1}},\end{aligned} \tag{7.46}$$

which concludes our proof.

□

7.3.3 Approximation in Sparse Tensor Product Models

In the last section we investigated the approximation error under dense tensor product models. In this section we will switch to the sparse/ high dimensional setting.

Now we present some more general conditions on the product basis and sparse nonparametric models. They can be seen as generalization of Condition 4.7.1 and Condition 4.7.2 in the main text.

Notation: recall that $\mathcal{X} \subset \mathbb{R}$ is a subset of real line and ν is a Borel measure on $\mathcal{X}$.

Condition 7.3.7. *Let ϕ_j be an orthonormal system of univariate functions, that is, $\langle \phi_i, \phi_j \rangle_{L_2(\nu)} = \delta_{ij}$. Assume $\phi_1 = 1$, $\|\phi_j\|_\infty \leq M$ for all $j = 1, 2, ...$ Consider their natural d-dimensional product extension $\psi_{\mathbf{j}}(\mathbf{x}) = \prod_{k=1}^d \phi_{\mathbf{j}^k}(\mathbf{x}^k)$, denote (ψ_j) to be the $c_{\mathbf{j}}$-unravelling sequence of $\{\psi_{\mathbf{j}}\}$. The unravelling rule $c_{\mathbf{j}}$ is defined as*

$$c_{\mathbf{j}} = \begin{cases} \prod_{k=1}^d \mathbf{j}^k & \text{, if at most } D' \text{ entries of } \mathbf{j} \text{ are greater than 1} \\ \infty & \text{otherwise} \end{cases} \tag{7.47}$$

Condition 7.3.8. *There exists a D-variate function $f^* : \mathcal{X}^D \to \mathbb{R}$ such that:*

1. *There is set of indices $\{k_1, .., k_D\} \subset \{1, 2, ..., d\}$ such that for any $\mathbf{u} \in \mathcal{X}^d$,*

$$f^0(\mathbf{u}) = f^*(\mathbf{u}^{k_1}, \mathbf{u}^{k_2}, ..., \mathbf{u}^{k_D}). \tag{7.48}$$

2. The function f^* satisfies the following ellipsoid condition:

$$f^* \in \left\{ f = \sum_{j=1}^{\infty} \beta_j \diamondsuit_j \;\; | \;\; \sum_{j=1}^{\infty} \left(\frac{j}{\log^{D-1} j \vee 1} \right)^{2s} \beta_j^2 \leq Q^2 \right\}. \tag{7.49}$$

The function sequence $(\diamondsuit_j)$ is the $\triangle_{\mathbf{j}}$-unravelling of $\diamondsuit_{\mathbf{j}} = \prod_{l=1}^{D} \phi_{\mathbf{j}^l}(\mathbf{u}^{k_l})$, $\mathbf{j} \in (\mathbb{N}^+)^D$. And the unravelling rule is defined by $\triangle_{\mathbf{j}} = \prod_{l=1}^{D} \mathbf{j}^l$.

The first part in Condition 7.3.8 is a feature sparsity assumption. Although f^0 formally is a function of d-dimensional vector $\mathbf{x}$ (d can be large), this assumption states that it can be completely described using a small subset of the dimensions of $\mathbf{x}$ (specifically, we assume it depends on D out of the d dimensions).

The second part in Condition 7.3.8 is in nature a smoothness assumption, but expressed in a basis expansion/Sobolev ellipsoid fashion. The basis functions $\diamondsuit_{\mathbf{j}}$ and unravelling rules $\triangle_{\mathbf{j}}$ only engage with the informative features $(\mathbf{u}^{k_1}, \mathbf{u}^{k_2}, ..., \mathbf{u}^{k_D})$. According to Lemma 7.3.6, if we use the first $J_n^{\text{oracle}} = \lfloor (\log^{D-1} n)^{2s/(2s+1)} n^{1/(2s+1)} \rfloor$ functions of $\diamondsuit_j$, we can construct a sequence of approximation functions $f_n^{\text{oracle}} = \sum_{j=1}^{J_n} \beta_{nj}^{\text{oracle}} \diamondsuit_j$ of f^* that satisfy

$$\| f_n^{\text{oracle}} - f^* \|_{2,\rho_X}^2 = O\left(\frac{\log^{D-1} n}{n} \right)^{\frac{2s}{2s+1}}. \tag{7.50}$$

However, in real-world problems, we unfortunately do not have *a priori* accessible information of which D dimensions of $\mathbf{x}$ are important. We thus cannot just use the oracle basis $\diamondsuit_j$ that only depend on the D relevant dimensions. The basis functions we use in (4.15) take the form of $\psi_{\mathbf{j}} = \prod_{k=1}^{d} \phi_{\mathbf{j}^k}(\mathbf{x}^k)$, involving d univariate functions as described in Condition 7.3.7. We are interested in how many functions we need to include in the sequence of ψ_j, such that we can achieve the same approximation error as f_n^{oracle}. The following Lemma tells us this number is exponential in the intrinsic dimension D (which we treat as a fixed number) but only polynomial in the ambient dimension d (which may formally increase with the sample size n). The polynomial dependence in d is important both theoretically and in practice.

Lemma 7.3.9. *Assume f^0 satisfies Condition 7.3.8. Denote (ψ_j) as the sequence of product basis functions in Condition 7.3.7. If the working dimension D' in Condition 7.3.7 is greater than or equal to the intrinsic dimension D in Condition 7.3.8, then:*

- *The true regression function f^0 can be expanded with respect to ψ_j as well, that is,*

$$f^0 = \sum_{j=1}^{\infty} \beta_j^0 \psi_j, \text{ for } \beta_j^0 \in \mathbb{R}. \tag{7.51}$$

- *There exists a sequence of functions $f_{\beta_n^0} = \sum_{j=1}^{J_n} \beta_{nj}^0 \psi_j$ with $J_n \leq C(s, D) d^{D'} n^{1/(2s+1)} \log^{D'-1} n$ such that*

$$\|f_{\beta_n^0}\|_\infty \leq C(M, s, D, Q) \tag{7.52}$$

and

$$\|f_{\beta_n^0} - f^0\|_{2,\rho_X}^2 \leq C(s, D, \rho_X, Q) \left(\frac{\log^{D-1} n}{n} \right)^{\frac{2s}{2s+1}}. \tag{7.53}$$

Proof. We introduce the mapping $1_{d \to D} : \mathbb{R}^d \to \mathbb{R}^D$ that only keeps the relevant dimensions of a feature $\mathbf{x}$:

$$1_{d \to D}(\mathbf{x}) = (\mathbf{x}^{k_1}, ..., \mathbf{x}^{k_D}), \tag{7.54}$$

where $k_1, ..., k_D$ are the informative dimension indices defined in Condition 7.3.8. By assumption, the true regression function can be written as:

$$f^0(\mathbf{x}) = f^*(1_{d \to D}(\mathbf{x})) = \sum_{j=1}^{\infty} \beta_j^* \diamondsuit_j (1_{d \to D}(\mathbf{x})) = \sum_{j=1}^{\infty} \beta_j^* \diamondsuit_j \circ 1_{d \to D}(\mathbf{x}) \tag{7.55}$$

Each of the basis functions above, $\diamondsuit_j \circ 1_{d \to D}$, varies at most in D dimensions. The function set $\{\psi_j\}$ in Condition 7.3.7 includes all the function product functions varying in at most D' dimensions. Since $\diamondsuit_j \circ 1_{d \to D}$ are also product functions, we conclude $\{\diamondsuit_j \circ 1_{d \to D}, j \in \mathbb{N}\} \subset \{\psi_j, j \in \mathbb{N}\}$. Therefore f^0 also has an expansion with respect to ψ_j as in (7.51).

Approximating f^* satisfying Condition 7.3.8 (equivalently, f^0), using the oracle basis $\diamondsuit_j$ in the ellipsoid assumption (7.49), is already studied in Lemma 7.3.6. We know that we need the first $J_n^{\text{oracle}} = (\log^{D-1} n)^{2s/(2s+1)} n^{1/(2s+1)}$ basis elements from $\{\diamondsuit_j\}$ in order to achieve the desired approximation error. We claim that

$$\left\{ \diamondsuit_1, ..., \diamondsuit_{J_n^{\text{oracle}}} \right\} \subset \left\{ \diamondsuit_{\mathbf{j}}, \mathbf{j} \in (\mathbb{N}^+)^D \;\middle|\; c_{\mathbf{j}} = \prod_{k=1}^{D} \mathbf{j}^k \leq C(s, D) R_n \right\}. \tag{7.56}$$

where $R_n = n^{1/(2s+1)}(\log n)^{-(D-1)/(2s+1)}$. To see this, we need to apply the number theory results we used to establish the equivalence between ellipsoids. According to Lemma 7.3.3 we know that

$$T_D\left(C(s,D)R_n\right) = C(s,D)n^{1/(2s+1)}\log^{\frac{(D-1)2s}{2s+1}} n + \text{ lower order terms} \geq J_n^{\text{oracle}}. \tag{7.57}$$

(recall that T_D is defined in Definition 7.3.2). However, in practice we do not know the oracle features, so we can only work with ψ_j or $\psi_{\mathbf{j}}$ (not $\diamondsuit_j$ or $\diamondsuit_{\mathbf{j}}$). To approximate f^0 well, we need to choose J_n large enough so that all the functions below are included:

$$\left\{\psi_{\mathbf{j}}, \mathbf{j} \in (\mathbb{N}^+)^d \;\middle|\; c_{\mathbf{j}} = \prod_{k=1}^{d} \mathbf{j}^k \leq C(s,D)R_n \text{ and at most } D \text{ of } \mathbf{j}^k \text{ are greater than } 1\right\}. \tag{7.58}$$

This ensures all the functions in the RHS of (7.56) are included (strictly speaking, their d-dimensional extensions are included). By our assumption that $D' > D$, we only need to select J_n large enough so that the following basis functions are all included:

$$\begin{aligned}&\left\{\psi_{\mathbf{j}}, \mathbf{j} \in (\mathbb{N}^+)^d \;\middle|\; c_{\mathbf{j}} = \prod_{k=1}^{d} \mathbf{j}^k \leq C(s,D)R_n \text{ and at most } D' \text{ of } \mathbf{j}^k \text{ are greater than } 1\right\} \\ &= \bigcup_{m=1}^{\lfloor C(s,D)R_n \rfloor} \left\{\psi_{\mathbf{j}} \;\middle|\; c_{\mathbf{j}} = \prod_{k=1}^{d} \mathbf{j}^k = m \text{ and at most } D' \text{ of } \mathbf{j}^k \text{ are greater than } 1\right\}\end{aligned} \tag{7.59}$$

How many elements are there in (7.59)? We give the following bound:

$$\begin{aligned}&\#\text{ of elements in (7.59)} \leq \\ &\underbrace{\sum_{m=1}^{C(s,D)R_n}}_{\text{consider all the } \mathbf{j} \text{ whose product is } m} \quad \underbrace{C_{D'}^d}_{\text{choose } D' \text{ dimensions}} \cdot \underbrace{\tau_{D'}(m)}_{\text{factorize } m \text{ into a product of } D' \text{numbers}} \\ &\leq C_{D'}^d T_{D'}(C(s,D)R_n) \overset{(1)}{\leq} C(s,D,D')d^{D'}n^{1/(2s+1)}\log^{D'-1} n.\end{aligned} \tag{7.60}$$

In (1) we used Lemma 7.3.3 and the well-known bound on the binomial coefficients $C_{D'}^d \leq C(D')d^{D'}$. Unraveling the functions set in (7.59) will give us at most the first $C(s,D,D')d^{D'}n^{1/(2s+1)}\log^{D'-1} n$ elements in (ψ_j). To achieve the desired approximation error bound, we do not need to use any additional basis elements. □

7.4 Theoretical Guarantees of Penalized Sieve Estimators

To present the statistical guarantees of l_1-penalized sieve estimators, we are going to employ the following steps:

1. We will give nonparametric oracle inequalities to control the "training-design error" of the estimators and the deviation of the estimated regression coefficients (Corollary 7.4.5).

2. We will use the information of the regression coefficients to derive a metric entropy bound on the function space the estimator lies in (Lemma 7.4.8).

3. We will control the difference between the training and testing errors of the estimate using results from empirical process theory (Theorem 7.4.10).

7.4.1 Nonparametric Oracle Inequalities

We first define the concept of the compatibility constant, which is an important component in the oracle inequalities and widely used in the analysis of penalized methods. In the rest of the section, for a $\beta = (\beta_1, ..., \beta_J)^\top \in \mathbb{R}^J$, we define its related function f_β as

$$f_\beta = \sum_{j=1}^{J} \beta_j \psi_j, \tag{7.61}$$

where (ψ_j) is the sequence of functions in Condition 7.3.7.

Definition 7.4.1. *For a given matrix Σ of size $J \times J$, constant L, and an index set $S \subset \{1, 2, ..., J\}$, we define the (Σ, L, S)-compatibility constant $\phi_\Sigma(L, S)$ to be*

$$\phi_\Sigma^2(L, S) = \min_\beta \left\{ \frac{|S|\beta^\top \Sigma \beta}{\|\beta_S\|_1^2} : \|\beta_{-S}\|_1 \leq L\|\beta_S\|_1 \right\}, \tag{7.62}$$

where $-S$ is the complementary set of S in $\{1, 2, .., J\}$. The notation $\beta_S \in \mathbb{R}^J$ is a shorthand for the "restriction" of a vector $\beta \in \mathbb{R}^J$ on the index set S: $(\beta_S)_j = \beta_j$ if $j \in S$, otherwise $(\beta_S)_j = 0$.

The following oracle inequality is a generalization of Theorem 2.2 in [121]. In our case, the true regression function does not have to be linear.

Theorem 7.4.2. *Let $(\mathbf{X}_i, Y_i)$, $i = 1, 2, ..., n$ denote the n IID samples. We use f^0 to denote the true conditional mean function and define $\epsilon_i = Y_i - f^0(\mathbf{X}_i)$. Let $J_n \geq 1$ be the number of basis function used in estimation. Let (ψ_j) be the unravelled sequence described in Condition 7.3.7. Let λ_ϵ be a number satisfying:*

$$\sup_{1 \leq j \leq J_n} \left| \frac{1}{n} \sum_{i=1}^{n} \psi_j(\mathbf{X}_i) \epsilon_i \right| \leq \lambda_\epsilon \tag{7.63}$$

Let $0 \leq \delta < 1$ and define for $\lambda > \lambda_\epsilon > 0$:

$$\underline{\lambda} = \lambda - \lambda_\epsilon, \bar{\lambda} = \lambda + \lambda_\epsilon + \delta \underline{\lambda}, \text{ and } L = \frac{\bar{\lambda}}{(1-\delta)\underline{\lambda}}. \tag{7.64}$$

We use $\hat{\beta}_n = (\beta_1^{PLS}, ..., \beta_{J_n}^{PLS})^\top$ to denote the minimizer of the penalized problem (4.15).

Then for any $\beta \in \mathbb{R}^{J_n}$ and any set $S \subset \{1, 2, ..., J_n\}$:

$$2\delta\underline{\lambda} \|\hat{\beta}_n - \beta\|_1 + \left\| f_{\hat{\beta}_n} - f^0 \right\|_n^2 \leq \left\| f_\beta - f^0 \right\|_n^2 + \frac{\bar{\lambda}^2 |S|}{\phi_{\hat{\Sigma}}^2(L, S)} + 4\lambda \left\| \beta_{-S} \right\|_1 \tag{7.65}$$

where $\phi_{\hat{\Sigma}}^2(L, S)$ is the $(\hat{\Sigma}, L, S)$-compatibility constant and the $\hat{\Sigma}$ is the empirical covariance matrix: $\hat{\Sigma}_{ij} = \frac{1}{n} \sum_{k=1}^{n} \psi_i(\mathbf{X}_k) \psi_j(\mathbf{X}_k)$.

Proof. We define $2\diamondsuit = \|f_{\hat{\beta}_n} - f^0\|_n^2 - \|f_\beta - f^0\|_n^2 + \|f_{\hat{\beta}_n} - f_\beta\|_n^2$. The empirical norm $\|\cdot\|_n$ can also be written in matrix form:

$$\begin{aligned} \|f_{\hat{\beta}_n} - f^0\|_n^2 =& \frac{1}{n} \sum_{i=1}^{n} \left(f_{\hat{\beta}_n}(\mathbf{X}_i) - f^0(\mathbf{X}_i) \right)^2 \\ &= \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{J_n} \beta_j^{PLS} \psi_j(\mathbf{X}_i) - f^0(\mathbf{X}_i) \right)^2 \\ &= \langle \hat{\Psi} \hat{\beta}_n - f^0(X), \hat{\Psi} \hat{\beta}_n - f^0(X) \rangle / n. \end{aligned} \tag{7.66}$$

The design matrix, $\hat{\Psi}$, has entries $\hat{\Psi}_{i,j} = \psi_j(\mathbf{X}_i)$. And $f^0(X) = (f^0(\mathbf{X}_1), ..., f^0(\mathbf{X}_n))^\top \in \mathbb{R}^n$ is the evaluation vector of f^0 at n features vectors $\{\mathbf{X}_i\}_{i=1}^n$. We will use the above equivalence later.

Similar to the proof in the literature [121], we consider two cases for $\diamondsuit$:

- If $\diamondsuit \leq -\delta\underline{\lambda}\|\hat{\beta}_n - \beta\|_1 + 2\lambda\|\beta_{-S}\|_1$. Then we have

$$\begin{aligned} & 2\delta\underline{\lambda}\|\hat{\beta}_n - \beta\|_1 + \|f_{\hat{\beta}_n} - f^0\|_n^2 \\ & \quad = 2\delta\underline{\lambda}\|\hat{\beta}_n - \beta\|_1 + 2\diamondsuit + \|f_\beta - f^0\|_n^2 - \|f_{\hat{\beta}_n} - f_\beta\|_n^2 \\ & \quad \leq \|f_\beta - f^0\|_n^2 - \|f_{\hat{\beta}_n} - f_\beta\|_n^2 + 4\lambda\|\beta_{-S}\|_1 \\ & \quad \leq \|f_\beta - f^0\|_n^2 + 4\lambda\|\beta_{-S}\|_1 \end{aligned} \tag{7.67}$$

- In the case when $\diamondsuit > -\delta\underline{\lambda}\|\hat{\beta}_n - \beta\|_1 + 2\lambda\|\beta_{-S}\|_1$, we start with the following *two point inequality* (Lemma 6.1 in [121]):

$$\langle\hat{\Psi}(\beta - \hat{\beta}_n), Y - \hat{\Psi}\hat{\beta}_n\rangle/n \leq \lambda\|\beta\|_1 - \lambda\|\hat{\beta}_n\|_1 \tag{7.68}$$

  Using the results in the beginning of this proof, we know $\diamondsuit$ can be expanded as:

$$\diamondsuit = \langle\hat{\Psi}\hat{\beta}_n, \hat{\Psi}\hat{\beta}_n\rangle/n - \langle\hat{\Psi}\hat{\beta}_n, f^0(X)\rangle/n + \langle\hat{\Psi}\beta, f^0(X)\rangle/n - \langle\hat{\Psi}\hat{\beta}_n, \hat{\Psi}\beta\rangle/n \tag{7.69}$$

  Then eq (7.68) implies that:

$$\diamondsuit \leq \langle\hat{\Psi}\hat{\beta}_n, \epsilon\rangle/n - \langle\hat{\Psi}\beta, \epsilon\rangle/n + \lambda\|\beta\|_1 - \lambda\|\hat{\beta}_n\|_1, \tag{7.70}$$

  The ϵ vector stores the noise variables: $\epsilon_i = Y_i - f^0(\mathbf{X}_i)$. The rest of the proof follows identically to that of Theorem 2.2 in [121] (page 21), replacing the $\left(\hat{\beta} - \beta\right)^\top \hat{\Sigma} \left(\hat{\beta} - \beta^0\right)$ term there by $\diamondsuit$.

□

The following lemmas tell us that the random compatibility constant $\phi_{\hat{\Sigma}}(L, S)$ is bounded away from zero with high probability.

Lemma 7.4.3. *Let Σ be the population $J_n \times J_n$ covariance matrix $\Sigma_{ij} = E[\psi_i(\mathbf{X})\psi_j(\mathbf{X})]$, where (ψ_j) is the unravelled function sequence defined in Condition 7.3.7. Assume the feature density function $p_X(\mathbf{x}) = d\rho_X/d\nu^d \geq u > 0$ is bounded away from 0. Here ν^d is the d-dimension product measure of ν in Condition 7.3.7.*

Then we know Σ has a compatibility constant that does not depend on L, S: $\phi_\Sigma^2 \geq u$.

Proof. For any $\beta \in \mathbb{R}^{J_n}$:

$$\begin{aligned}\beta^\top \Sigma \beta &= \sum_{1\le i,j\le J_n} \beta_i \beta_j E[\psi_i(\mathbf{X})\psi_j(\mathbf{X})] = E\left[\left(\sum_{j=1}^{J_n} \psi_j(\mathbf{X})\beta_j\right)^2\right] \\ &\ge u\int \left(\sum_{j=1}^{J_n} \psi_j(\mathbf{x})\beta_j\right)^2 d\mathbf{x} \overset{(1)}{=} u\|\beta\|_2^2.\end{aligned} \tag{7.71}$$

In step (1) we used the orthonomality of ψ_j stated in Condition 7.3.7. At the same time, we have $\|\beta_S\|_1^2 \le \|\beta\|_2^2|S|$. Checking the definition of compatibility (Definition 7.4.1), we conclude for any L, S, the matrix Σ has a uniform compatibility constant ϕ_Σ greater than $\sqrt{u}$ (meaning that this lower bound does not depend on either L or S). □

Lemma 7.4.4. *Under the same conditions as in Lemma 7.4.3, we know the empirical matrix $\hat{\Sigma}$ has a compatibility constant $\phi_{\hat{\Sigma}}^2(L,S) \ge u/2$, with probability at least $1 - J_n^2 \exp(-na^2/2M^{4D'})$, $a = u(L+1)^{-2}|S|^{-1}/2$.*

Proof. We first consider the difference between two quadratic forms related to the two covariance matrices:

$$|\beta^\top \hat{\Sigma}\beta - \beta^\top \Sigma \beta| = \left|\sum_{1\le i,j\le J_n} \beta_i\beta_j(\hat{\Sigma}_{ij} - \Sigma_{ij})\right| \le \|\beta\|_1^2 \|\hat{\Sigma} - \Sigma\|_\infty \tag{7.72}$$

By the definition of compatibility constant ϕ_Σ, for any β such that $\|\beta_{-S}\|_1 \le L\|\beta_S\|_1$, we have

$$\|\beta\|_1 \le (L+1)\|\beta_S\|_1 \le (L+1)\sqrt{|S|\beta^\top\Sigma\beta}/\phi_\Sigma \tag{7.73}$$

Plugging this into (7.72), we have:

$$\begin{aligned}&\mid \beta^\top\hat{\Sigma}\beta - \beta^\top\Sigma\beta \mid \le (L+1)^2\|\hat{\Sigma}-\Sigma\|_\infty |S|\beta^\top\Sigma\beta/\phi_\Sigma^2 \\ \iff &\left|\frac{\beta^\top\hat{\Sigma}\beta}{\beta^\top\Sigma\beta} - 1\right| \le (L+1)^2\|\hat{\Sigma}-\Sigma\|_\infty |S|/\phi_\Sigma^2\end{aligned} \tag{7.74}$$

By a typical application of Hoeffding's inequality (every entry in $\hat{\Sigma}$ is a bounded random variable), we know with probability at least $1 - J_n^2\exp(-na^2/2M^{4D'})$, where $a = u(L+1)^{-2}|S|^{-1}/2$, that

$$\|\hat{\Sigma} - \Sigma\|_\infty \le (2(L+1)^2|S|/u)^{-1} \tag{7.75}$$

This means, with the same probability we have

$$\left|\frac{\beta^\top \hat{\Sigma}\beta}{\beta^\top \Sigma\beta} - 1\right| \leq \frac{1}{2} \tag{7.76}$$

Therefore, for all any β such that $\|\beta_{-S}\|_1 \leq L\|\beta_S\|_1$, we have that:

$$\frac{|S|\beta^\top \hat{\Sigma}\beta}{\|\beta_S\|_1^2} \geq \frac{|S|\beta^\top \Sigma\beta}{2\|\beta_S\|_1^2}, \tag{7.77}$$

with high probability. By the definition of the compatibility constant, we can read out

$$\phi^2_{\hat{\Sigma}}(L, S) \geq \phi^2_{\Sigma}/2 \tag{7.78}$$

which concludes our proof. □

Corollary 7.4.5. *Let $\lambda_\epsilon = [2\log(2J_n)/\{C(C_{sub}, M, D')n\}]^{1/2}$ and assume ϵ_i to be uniform sub-Gaussian noise. Then, under the same conditions as in Theorem 7.4.2, for any $\beta \in \mathbb{R}^{J_n}$ whose support is $S \subset \{1, 2, ..., J_n\}$, we have*

$$\lambda_\epsilon\|\hat{\beta}_n - \beta\|_1 + \left\|f_{\hat{\beta}_n} - f^0\right\|_n^2 \leq \frac{3}{2}\left\|f_\beta - f^0\right\|_2^2 + \frac{49\lambda_\epsilon^2|S|}{2u} \tag{7.79}$$

with probability larger than $1 - 1/(2J_n) - J_n^2\exp(-na^2/2M^{4D'}) - \exp(-cn\|f_\beta - f^0\|_2^2/M_0^2)$, where $a = u(L+1)^{-2}|S|^{-1}/2$. The definition of f_β is stated in (7.61).

Proof. First we show that for the chosen λ_ϵ, the following holds with high probability:

$$\sup_{1\leq j\leq J_n}\left|\frac{1}{n}\sum_{i=1}^n \psi_j(\mathbf{X}_i)\epsilon_i\right| \leq \lambda_\epsilon. \tag{7.80}$$

Since (ϵ_i) are sub-Gaussian random variables (with a parameter not depending on $\mathbf{X}_i$), we know there exists a constant C_{sub} such that

$$\|\epsilon_i\|_{L^p} = \left\{E\left(|\epsilon_i|^p\right)\right\}^{1/p} \leq C_{sub}\sqrt{p} \quad \text{for all } p \geq 1 \tag{7.81}$$

For reference, see e.g. Proposition 2.5.2 in [125]. Since the basis functions ψ_j are also uniformly bounded (by $M^{D'}$), we have

$$\|\psi_j(\mathbf{X}_i)\epsilon_i\|_{L^p} \leq C_{sub}M^{D'}\sqrt{p} \quad \text{for all } p \geq 1. \tag{7.82}$$

This means $\psi_j(\mathbf{X}_i)\epsilon_i$ is also sub-Gaussian. Applying a union bound and Hoeffding's inequality for sub-Gaussian variables (e.g. Theorem 2.6.3 in [125]), we get:

$$\begin{aligned} \text{pr}\left\{\sup_{1\le j\le J_n}\left|\frac{1}{n}\sum_{i=1}^{n}\psi_j(\mathbf{X}_i)\epsilon_i\right|\ge t\right\} &\le \sum_{j=1}^{J_n}\text{pr}\left\{\left|\frac{1}{n}\sum_{i=1}^{n}\psi_j(\mathbf{X}_i)\epsilon_i\right|\ge t\right\} \\ &\le 2J_n\exp\left\{-C(C_{sub},M,D')nt^2\right\} \end{aligned} \tag{7.83}$$

Taking $t=\lambda_\epsilon=[2\log(2J_n)/\{C(C_{sub},M,D')n\}]^{1/2}$, we see that

$$\text{pr}\left\{\sup_{1\le j\le J_n}\left|\frac{1}{n}\sum_{i=1}^{n}\psi_j(\mathbf{X}_i)\epsilon_i\right|\le\lambda_\epsilon\right\}\ge 1-1/(2J_n) \tag{7.84}$$

This is what we claimed in the beginning of the proof.

Next, we bound the difference between $\|f_\beta-f^0\|_n^2$ and $\|f_\beta-f^0\|_2^2$ for any fixed f_β satisfying $\|f_\beta\|_\infty<2\|f^0\|_\infty$. First, the difference $(f_\beta(\mathbf{X}_i)-f^0(\mathbf{X}_i))^2$ is a bounded variable, therefore it is sub-Gaussian with parameter $9M_0^2$, where M_0 bounds $\|f^0\|_\infty$. The centered version, $\{f_\beta(\mathbf{X}_i)-f^0(\mathbf{X}_i)\}^2-\|f_\beta-f^0\|_2^2$ is also sub-Gaussian with parameter CM_0^2 (see e.g. Lemma 2.6.8 in [125]). Again applying Hoeffding's inequality we see that

$$\begin{aligned} &\text{pr}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\{f_\beta(\mathbf{X}_i)-f^0(\mathbf{X}_i)\}^2-\|f_\beta-f^0\|_2^2\right|\ge t\right]\le\exp(-cnt^2/M_0^2) \\ &\Rightarrow \text{pr}\left(\left|\frac{\|f_\beta-f^0\|_n^2}{\|f_\beta-f^0\|_2^2}-1\right|\ge\frac{1}{2}\right)\le\exp(-cn\|f_\beta-f^0\|_2^2/M_0^2) \end{aligned} \tag{7.85}$$

We know, with probability larger than $1-\exp(-cn\|f_\beta-f^0\|_2^2/M_0^2)$

$$\frac{\|f_\beta-f^0\|_n^2}{\|f_\beta-f^0\|_2^2}\le\frac{3}{2} \tag{7.86}$$

By combining (7.84), (7.86), Lemma 7.4.4 and Theorem 7.4.2, we can conclude our proof. □

7.4.2 Theoretical Guarantees under Sparse Tensor Product Models

In this section, we will combine the oracle inequalities developed in the last section with approximation results to derive performance guarantees of the l_1-penalized sieve estimator.

Recall the following notation: d is the overall ambient dimension of our features $\mathbf{X}_i$, D is the number of explanatory features related to the outcome Y (the active dimension), s is

the smoothness parameter of f^0 (Condition 7.3.8), and J_n is the number of basis functions in the lasso problem (4.15). The constant C_{sub} is the sub-Gaussian parameter for the noise variables, u is the lower bound of the feature density function and M_0 is a bound on the $\|\cdot\|_\infty$-norm of f^0.

Corollary 7.4.6. *Let $f_{\hat{\beta}_n}$ be the penalized sieve estimate of f^0, and $f_{\beta_n^0}$ be the approximation of f^0 as in Lemma 7.3.9. Choose the penalization hyperparameter as $\lambda_\epsilon = [2\log(2J_n)/\{C(C_{sub}, M, D')n\}]^{1/2}$. Under the same conditions as in Theorem 4.7.3, we have the following two bounds*

$$\begin{aligned} \left\| f_{\hat{\beta}_n} - f^0 \right\|_n^2 &\leq C(C_{sub}, M, D', \rho_X, f^0) \log(J_n) \left(\frac{\log^{D-1}(n)}{n} \right)^{\frac{2s}{2s+1}} \\ \|\hat{\beta}_n - \beta_n^0\|_1 &\leq C(C_{sub}, M, D', \rho_X, f^0)(\log J_n/n)^{1/2} n^{1/(2s+1)} (\log n)^{2s(D-1)/(2s+1)} \end{aligned} \tag{7.87}$$

with probability at least

$$1 - 1/(2J_n) - J_n^2 \exp(-na^2/2M^{4D'}) - \exp\left(-C(s, D, \rho_X, f^0)(\log n)^{(D-1)2s/(2s+1)} n^{1/(2s+1)}\right),$$

where $a = u(L+1)^{-2}|S|^{-1}/2$.

Proof. To get the bounds above, we only need to combine the oracle inequality in Corollary 7.4.5 with the approximation results in Lemma 7.3.9.

In Lemma 7.3.9, we discussed that so long as J_n is large enough, we can find a function $f_{\beta_n^0}$ that approximates f^0 well enough. Plugging the results of Lemma 7.3.9 into the oracle inequality (7.79), we have:

$$\lambda_\epsilon \|\hat{\beta}_n - \beta\|_1 + \left\| f_{\hat{\beta}_n} - f^0 \right\|_n^2 \leq C(s, D, \rho_X, f^0) \left(\frac{\log^{D-1} n}{n} \right)^{\frac{2s}{2s+1}} + \frac{49\lambda_\epsilon^2 |S_n|}{2u}, \tag{7.88}$$

here $|S_n|$ is the cardinality of non-zero elements in β_n^0. Although formally $f_{\beta_n^0}$ is a linear combination of $J_n = C(s, D) d^{D'} n^{1/(2s+1)} \log^{D'-1} n$ basis functions, the size of its support is much smaller (thanks to the feature sparsity conditions). In fact, $f_{\beta_n^0}$ only needs to engage with the informative dimensions of the features. In Lemma 7.3.6, we showed that $|S_n|$ can be bounded by $(\log^{D-1} n)^{2s/(2s+1)} n^{1/(2s+1)}$. Plugging this in the above inequality gives:

$$\lambda_\epsilon \|\hat{\beta}_n - \beta\|_1 + \left\| f_{\hat{\beta}_n} - f^0 \right\|_n^2 \leq C(s, D, \rho_X, f^0) \left(\frac{\log^{D-1} n}{n} \right)^{\frac{2s}{2s+1}} + C(C_{sub}, M, D', \rho_X) \log(J_n) \left(\frac{\log^{D-1} n}{n} \right)^{\frac{2s}{2s+1}}. \tag{7.89}$$

This gives us the results regarding the $\|\cdot\|_n$-norm distance and l_1-distance stated in Corollary 7.4.6 (the second term will dominate for large n). □

At this point, we already established bounds on the training-design error (expressed as the $\|\cdot\|_n$-norm). However, for most prediction problems we are interested in the testing error (quantified in the $\|\cdot\|_{2,\rho_X}$-norm). For arbitrarily flexible estimators, a low training-design error does not imply a strong generalization ability. However, according to Corollary 7.4.6, the coefficient $\hat{\beta}_n$ lives in a small $\|\cdot\|_1$-ball centered around the oracle β_n^0 with high probability. From this we can also develop some bounds on metric entropy of the space in which $f_{\hat{\beta}_n}$ takes value. These will in turn link the expected distance to the empirical distance.

In the following discussion we will use the concept of metric entropy of a function space. For more comprehensive discussion, see Chapter 2 of [120].

Definition 7.4.7. *Let Q be a measure on $\mathcal{X}$ and let $\mathcal{G}$ be a function space $\mathcal{G} \subset L_2(\mathcal{X}; Q)$. Consider for each $\delta > 0$, a collection of functions $g_1, ..., g_N$, such that for each $g \in \mathcal{G}$, there is a $j = j(g) \in \{1, ..., N\}$, such that*

$$\left(\int_{\mathcal{X}} (g(x) - g_j(x))^2 dQ(x) \right)^{1/2} \leq \delta. \tag{7.90}$$

Let $N(\delta, \mathcal{G}, Q)$ be the smallest value of N for which such a covering by balls with radius δ and centers $g_1, ..., g_N$ exists. Then $N(\delta, \mathcal{G}, Q)$ is called the covering number (under measure Q) and $H(\delta, \mathcal{G}, Q) = \log N(\delta, \mathcal{G}, Q)$ is called the metric entropy of $\mathcal{G}$ (under measure Q).

One of the function spaces $\mathcal{G}_n$ we are going to consider is

$$\mathcal{G}_n = \mathcal{G}_n(\psi_j, \beta_n^0, r_n) = \left\{ f = \sum_{j=1}^{J_n} \beta_j \psi_j \mid \beta = (\beta_1, ..., \beta_{J_n})^\top \in B_1(\beta_n^0, r_n) \right\}, \tag{7.91}$$

with $r_n = r_n(s, D) = (\log J_n/n)^{1/2} n^{1/(2s+1)} \log^{2s(D-1)/(2s+1)}(n)$. This radius is of the same order as the RHS in (7.87). The set $B_1(\beta, r) \subset \mathbb{R}^{J_n}$ is the $\|\cdot\|_1$-ball of radius r centered at β, formally

$$B_1(\beta, r) = \{\gamma \in \mathbb{R}^{J_n} \mid \|\gamma - \beta\|_1 \leq r\} \tag{7.92}$$

For a specified sequence of r_n and J_n, $\mathcal{G}_n$ is a deterministic sequence of function spaces.

In the rest of this section, we will derive some bounds on the metric entropy of $\mathcal{G}_n$ and apply some maximal inequalities to relate the testing-design errors to the training-design errors. We will show that the metric entropy of the function space $\mathcal{G}_n$ (equipped with $\|\cdot\|_n$-norm) is of the same order as the metric entropy of $B_1(\beta_n, r_n)$ (equipped with Euclidean $\|\cdot\|_2$-norm). Since the latter is known in the literature (e.g, Lemma 3 in [91]), we have the following results:

Lemma 7.4.8. *Let $r_n = r_n(s, D) = (\log J_n/n)^{1/2} n^{1/(2s+1)} (\log n)^{2s(D-1)/(2s+1)}$. Then for the $\mathcal{G}_n$ defined in* (7.91)*, we have*

$$H(\delta, \mathcal{G}_n, P_n) \leq C(M) r_n^2 \delta^{-2} \log J_n \tag{7.93}$$

Proof. We first rewrite the empirical norm in matrix notation, for any $\beta = (\beta_1, ..., \beta_{J_n})^\top \in \mathbb{R}^{J_n}$:

$$\|f_\beta\|_n = \left\{ \frac{1}{n} \sum_{i=1}^n f_\beta^2(\mathbf{X}_i) \right\}^{1/2} = \frac{1}{\sqrt{n}} \left\{ \sum_{i=1}^n \left(\sum_{j=1}^{J_n} \beta_j \psi_j(\mathbf{X}_i) \right)^2 \right\}^{1/2} = \left\| \frac{1}{\sqrt{n}} \hat{\Psi} \beta \right\|_2, \tag{7.94}$$

where $\hat{\Psi}$ is the design matrix: $\hat{\Psi}_{ij} = \psi_j(\mathbf{X}_i)$.

Therefore, if there is a δ-cover of the set $\left\{ n^{-1/2} \hat{\Psi} \beta, \beta \in B_1(\beta_n^0, r_n) \right\} \subset \mathbb{R}^n$ under the Euclidean $\|\cdot\|_2$-norm, we can directly construct one for $\mathcal{G}_n$ under $\|\cdot\|_n$-norm. There are available bounds on the covering number of the $n^{-1/2}\hat{\Psi}\beta$ when β belongs to a l_1-ball. Specifically, we can apply Lemma 4 of [91]:

$$H\left(\delta, \left\{ n^{-1/2} \hat{\Psi} \beta, \beta \in B_1(\beta_n^0, r_n) \right\}, \|\cdot\|\right) \leq C(M) r_n^2 \delta^{-2} \log J_n. \tag{7.95}$$

This concludes our proof. □

To relate the training and testing errors, we need to consider a function space closely related to $\mathcal{G}_n$:

$$\tilde{\mathcal{G}}_n^2 = (\mathcal{G}_n - f^0)^2 \tag{7.96}$$

We summarize several properties of it in the following lemma.

Lemma 7.4.9. *Let* $r_n = (\log J_n/n)^{1/2} n^{1/(2s+1)} (\log n)^{2s(D-1)/(2s+1)}$ *and* $\delta_n < r_n$. *Then for the function space* $\tilde{\mathcal{G}}_n^2$ *we know:*

$$\begin{aligned} &\sup_{g\in\tilde{\mathcal{G}}_n^2} \|g\|_\infty \le C(M, D', s, D, Q), \\ &pr\left\{ \sup_{g\in\tilde{\mathcal{G}}_n^2} \|g\|_n \le C(M, D', s, D, Q) r_n \right\} \overset{n\to\infty}{\longrightarrow} 1 \text{ and} \\ &\int_{\delta_n}^{r_n} H^{1/2}(u, \tilde{\mathcal{G}}_n^2, P_n) du \le C(M, D', s, D, Q) r_n (\log J_n)^{1/2} \log(1/\delta_n). \end{aligned} \tag{7.97}$$

Proof.

- We first derive the bound on the $\|\cdot\|_\infty$-norm. By definition, every element g in $\tilde{\mathcal{G}}_n^2$ can be expressed as $g = (f - f^0)^2$ for some $f \in \mathcal{G}_n$. To bound $\|g\|_\infty$, it is enough to bound $\|f - f^0\|_\infty$.

$$\begin{aligned} \|f - f^0\|_\infty &\le \|f - f_{\beta_n^0}\|_\infty + \|f_{\beta_n^0} - f^0\|_\infty \\ &\le C(M, D') r_n + \|f_{\beta_n^0}\|_\infty + \|f^0\|_\infty \le C(M, D', s, D, Q). \end{aligned} \tag{7.98}$$

- We now bound the empirical norm $\|\cdot\|_n$. For any $f \in \mathcal{G}_n$, we can define a function $g = f - f^0$. And we know:

$$\begin{aligned} \|g\|_n &\le \|f - f_{\beta_n^0}\|_n + \|f_{\beta_n^0} - f^0\|_n \\ &\le C(M, D') r_n + C(s, D, \rho_X, Q)(\log^{D-1} n/n)^{s/2s+1} \text{ with high probability.} \end{aligned} \tag{7.99}$$

  The first term is using the explicit form of f and $f_{\beta_n^0}$. The second bound is based on the approximation results in Lemma 7.3.9 and the probability bound in (7.86). Since $r_n = (\log J_n/n)^{1/2} n^{1/(2s+1)} (\log n)^{2s(D-1)/(2s+1)}$, the order of the first term in (7.99) is larger than the second one's. For $g^2 \in \tilde{\mathcal{G}}_n^2$,

$$\|g^2\|_n^2 = \frac{1}{n}\sum_{i=1}^n \left\{ f(\mathbf{X}_i) - f^0(\mathbf{X}_i) \right\}^4 \le C(M, D', s, D, Q) \|g\|_n^2 \le C(M, D', s, D, Q) r_n^2. \tag{7.100}$$

So we conclude that for any $g^2 \in \tilde{\mathcal{G}}_n^2$, $\|g^2\|_n \leq C(M, D', s, D, Q) r_n$ with probability going to 1.

- Now we derive the bound on the integrated metric entropy. For any $h_1, h_2 \in \tilde{\mathcal{G}}_n^2$, there exist $f_1, f_2 \in \mathcal{G}_n$ such that $h_i = (f_i - f^0)^2$, $i \in \{1, 2\}$. So we know that

$$\begin{aligned}\|h_1 - h_2\|_n^2 &= \|(f_1 - f^0)^2 - (f_2 - f^0)^2\|_n^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} \left[\left\{f_1(\mathbf{X}_i) + f_2(\mathbf{X}_i) - 2f^0(\mathbf{X}_i)\right\}\left\{f_1(\mathbf{X}_i) - f_2(\mathbf{X}_i)\right\}\right]^2 \\ &\leq C(M, D', s, D, Q)\frac{1}{n}\sum_{i=1}^{n}\left\{f_1(\mathbf{X}_i) - f_2(\mathbf{X}_i)\right\}^2 = C(M, D', s, D, Q)\|f_1 - f_2\|_n^2\end{aligned} \tag{7.101}$$

Now we know that if we have a δ-covering of $\mathcal{G}_n$ with center points $\{f_k\}$, then the functions $\{(f_k - f^0)^2\}$ form a $C(M, D', s, D, Q)\delta$-covering of $\tilde{\mathcal{G}}_n^2$. Since we already have an entropy bound on $\mathcal{G}_n$ stated in Lemma 7.4.8, we have one for $\tilde{\mathcal{G}}_n^2$ of the same order as well. The integrated entropy can be bounded as follows:

$$\begin{aligned}\int_{\delta_n}^{r_n} H^{1/2}(\tau, \tilde{\mathcal{G}}_n^2, P_n)d\tau &\leq C(M, D', s, D, Q)\int_{\delta_n}^{r_n}(\log J_n)^{1/2} r_n \tau^{-1} d\tau \\ &\leq C(M, D', s, D, Q)(\log J_n)^{1/2} r_n \log(1/\delta_n),\end{aligned} \tag{7.102}$$

when $r_n \leq 1$.

□

Theorem 7.4.10. *Let*

$$\begin{aligned}\delta_n &= C(s, D)\log(J_n) n^{-\frac{2s}{2s+1}} \log^{\frac{2s(D-1)}{2s+1}+1}(n), \\ r_n &= C(s, D)(\log J_n/n)^{1/2} n^{1/(2s+1)} (\log n)^{2s(D-1)/(2s+1)}.\end{aligned} \tag{7.103}$$

And let $\hat{\beta}_n$ denote the minimizer of the penalized problem (4.15)*. Then, under the same conditions as in Theorem 4.7.3, we have*

$$\limsup_{n\to\infty} pr\left(\left|\|f_{\hat{\beta}_n} - f^0\|_n^2 - \|f_{\hat{\beta}_n} - f^0\|_2^2\right| \geq \delta_n\right) = 0 \tag{7.104}$$

Proof. We first need to apply the symmetrization trick (e.g. Corollary 3.4 in [120])

$$\begin{aligned}\text{pr}\left(\left|\|f_{\hat{\beta}_n}-f^0\|_n^2-\|f_{\hat{\beta}_n}-f^0\|_2^2\right|\geq\delta_n\right)&=\text{pr}\left\{\left|(P_n-P)\left(f_{\hat{\beta}_n}-f^0\right)^2\right|\geq\delta_n\right\}\\&\leq\text{pr}\left\{\sup_{g\in\tilde{\mathcal{G}}_n^2}|(P_n-P)g|\geq\delta_n\right\}+\text{pr}(f_{\hat{\beta}_n}\notin\mathcal{G}_n)\\&\leq 4\text{pr}\left\{\sup_{g\in\tilde{\mathcal{G}}_n^2}\left|\frac{1}{n}\sum_{i=1}^n W_i g(\mathbf{X}_i)\right|\geq\delta_n/4\right\}+\text{pr}(f_{\hat{\beta}_n}\notin\mathcal{G}_n)\end{aligned}\tag{7.105}$$

The W_i variables above are independent and identically distributed Rademacher variables ($\text{pr}(W_i=1)=\text{pr}(W_i=-1)=0.5$). They are bounded (therefore sub-Gaussian) random variables. The probability $\text{pr}(f_{\hat{\beta}_n}\notin\mathcal{G}_n)$ has been investigated in Corollary 7.4.6. To bound the first term in (7.105), we need to apply some maximal inequalities (e.g., Corollary 8.3 or Lemma 3.2 in [120]). These results require that $r_n>\delta_n$ and

$$\sqrt{n}\delta_n\geq C\left(\int_{\delta_n/8}^{r_n}H^{1/2}\left(\tau,\tilde{\mathcal{G}}_n^2,P_n\right)d\tau\vee r_n\right).\tag{7.106}$$

We already checked these properties in Lemma 7.4.9. So we conclude that with probability going to 1, the difference between the training and testing error is no larger than δ_n. □

Proof of Theorem 4.7.3. To show the testing error stated in Theorem 4.7.3, we just need to combine the results in Theorem 7.4.10 and Corollary 7.4.6. □

7.5 The Average Order of Divisor Functions

In this section we will present a derivation of the average order of D-divisor functions that was used in the proof of Lemma 7.3.3. This result is known to mathematicians working on number theory, and is usually considered as a direct generalization of the $D=2$ case. However, most standard references only include the special (but important) case when $D=2$. For the purpose of completeness, we replicate a proof based on an unpublished online note by Graham Jameson, from Lancaster University. For other references of similar results, see [56] (Proposition 6) and [26].

Lemma 7.5.1. *We have the following recurrence relation for T_D (over D):*

$$T_D(x) = \sum_{n \le x} T_{D-1}\left(\frac{x}{n}\right) \tag{7.107}$$

Proof. Fix $n \le x$. The number of D-tuples $\left(\mathbf{j}^1, \mathbf{j}^2, \ldots, \mathbf{j}^{D-1}, n\right)$ with $n \prod_{k=1}^{D-1} \mathbf{j}^k \le x$ is the number of $(D-1)$-tuples with $\prod_{k=1}^{D-1} \mathbf{j}^k \le x/n$, that is, $T_{D-1}(x/n)$. Hence $T_D(x) = \sum_{n \le x} T_{D-1}(x/n)$.

□

Lemma 7.5.2. *Define*

$$A_D(x) = \sum_{n \le x} \frac{x}{n} \log^D(x/n) \tag{7.108}$$

then we have

$$\frac{1}{D+1} x \log^{D+1} x \le A_D(x) \le \frac{1}{D+1} x \log^{D+1} x + x \log^D x \tag{7.109}$$

Proof. Let $f(t) = (x/t) \log^D(x/t)$ for $1 \le t \le x$ (also $f(t) = 0$ for $t > x$). Then $f(t)$ is decreasing and non-negative, and

$$\int_1^x f(t)dt = \left[\frac{x}{D+1} \log^{D+1}(u)\right]_1^x = \frac{x \log^{D+1}(x)}{D+1} \tag{7.110}$$

The statement follows, by using the following basic integral estimate (Proposition 1.4.2 of [57]): Let $f(t)$ be a decreasing, non-negative function for $t \ge 1$. Write $S(x) = \sum_{n \le x} f(n)$ and $I(x) = \int_1^x f(t)dt$. Then for all $x \ge 1$,

$$I(x) \le S(x) \le I(x) + f(1) \tag{7.111}$$

□

Lemma 7.5.3. *For any fixed $D \ge 2$,*

$$T_D(x) = \frac{1}{(D-1)!} x \log^{D-1} x + O\left(x \log^{D-2} x\right). \tag{7.112}$$

The $O(\cdot)$ in (7.112) is for $x \to \infty$.

Proof. Induction on D. The case $D = 2$ is known to be true (Theorem 3.2 [117]). Assume (7.112) for D, with the error term denoted by $q_D(x)$. Then by (7.107),

$$T_{D+1}(x) = \sum_{n \le x} T_D\left(\frac{x}{n}\right) = I(x) + Q(x) \tag{7.113}$$

where

$$\begin{aligned} I(x) &= \frac{1}{(D-1)!} \sum_{n \le x} \frac{x}{n} \log^{D-1} \frac{x}{n} = \frac{1}{(D-1)!} A_{D-1}(x) \\ Q(x) &= \sum_{n \le x} q_D\left(\frac{x}{n}\right) \sim \sum_{n \le x} \frac{x}{n} \log^{D-2} \frac{x}{n} = A_{D-2}(x) \end{aligned} \tag{7.114}$$

By (7.109),

$$I(x) = \frac{1}{D!} x \log^D x + O\left(x \log^{D-1} x\right) \tag{7.115}$$

and $Q(x) = O(x \log^{D-1} x)$. Hence (7.112) holds for $D + 1$. □

BIBLIOGRAPHY

[1] Ali Akgül, Esra Karatas Akgül, and Sahin Korhan. New reproducing kernel functions in the reproducing kernel sobolev spaces. *AIMS Mathematics*, 5(1):482–496, 2020.

[2] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, pages 775–783, 2015.

[3] Antonios K Alexandridis and Achilleas D Zapranis. Wavelet neural networks: A practical guide. *Neural Networks*, 42:1–27, 2013.

[4] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[5] Dmitry Babichev and Francis Bach. Constant step size stochastic gradient descent for probabilistic modeling. *stat*, 1050:21, 2018.

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

[7] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning and the bias-variance trade-off. *arXiv preprint arXiv:1812.11118*, 2018.

[8] Alexandre Belloni, Victor Chernozhukov, and Lie Wang. Pivotal estimation via square-root lasso in nonparametric regression. *The Annals of Statistics*, 42(2):757–788, 2014.

[9] David Benkeser and Mark Van Der Laan. The highly adaptive lasso estimator. In *2016 IEEE international conference on data science and advanced analytics (DSAA)*, pages 689–696. IEEE, 2016.

[10] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[11] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[12] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[13] Difeng Cai and Panayot S Vassilevski. Eigenvalue problems for exponential-type kernels. *Computational Methods in Applied Mathematics*, 20(1):61–78, 2020.

[14] T Tony Cai, Tengyuan Liang, Alexander Rakhlin, et al. Computational and statistical boundaries for submatrix localization in a large noisy matrix. *The Annals of Statistics*, 45(4):1403–1430, 2017.

[15] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Efficient second-order online kernel learning with adaptive embedding. In *Advances in Neural Information Processing Systems*, pages 6140–6150, 2017.

[16] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[17] Bernd Carl. Entropy numbers, s-numbers, and eigenvalue problems. *Journal of Functional Analysis*, 41(3):290–306, 1981.

[18] Bernd Carl and Irmtraud Stephani. *Entropy, Compactness and the Approximation of Operators*. Cambridge Tracts in Mathematics. Cambridge University Press, 1990.

[19] Xiaohong Chen. Large sample sieve estimation of semi-nonparametric models. *Handbook of econometrics*, 6:5549–5632, 2007.

[20] Andreas Christmann and Ingo Steinwart. Support vector machines. 2008.

[21] Donald L Cohn. *Measure theory*. Springer, 2013.

[22] Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bulletin of the American mathematical society*, 39(1):1–49, 2002.

[23] Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj, Maria-Florina F Balcan, and Le Song. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems*, pages 3041–3049, 2014.

[24] Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013.

[25] Aymeric Dieuleveut and Francis Bach. Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics*, 44(4):1363–1399, 2016.

[26] Nikolai Mikhailovich Dobrovol'skii and Alla Lenstovna Roshchenya. Number of lattice points in the hyperbolic cross. *Matematicheskie Zametki*, 63(3):363–369, 1998.

[27] John C Duchi. *Multiple Optimality Guarantees in Statistical Learning*. PhD thesis, UC Berkeley, 2014.

[28] Dinh Dũng, Vladimir Temlyakov, and Tino Ullrich. *Hyperbolic cross approximation*. Springer, 2018.

[29] Sam Efromovich. *Nonparametric curve estimation: methods, theory, and applications*. Springer Science & Business Media, 2008.

[30] Sam Efromovich. Orthogonal series density estimation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):467–476, 2010.

[31] R. Eubank and P. Speckman. Curve fitting by polynomial-trigonometric regression. *Biometrika*, 77:1–9, 1990.

[32] Gregory E Fasshauer. Green's functions: Taking another look at kernel approximation, radial basis functions, and splines. In *Approximation Theory XIII: San Antonio 2010*, pages 37–63. Springer, 2012.

[33] Gregory E Fasshauer and Michael J McCourt. *Kernel-based approximation methods using Matlab*, volume 19. World Scientific Publishing Company, 2015.

[34] Bengt Fornberg and Cécile Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM Journal on Scientific Computing*, 30(1):60–80, 2008.

[35] Jerome H Friedman. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–67, 1991.

[36] Jerome H Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.

[37] Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory*, pages 728–763, 2015.

[38] Pierre Gaillard and Sébastien Gerchinovitz. A chaining algorithm for online nonparametric regression. In *Conference on Learning Theory*, pages 764–796, 2015.

[39] Chao Gao, Zongming Ma, Zhao Ren, Harrison H Zhou, et al. Minimax estimation in sparse canonical correlation analysis. *The Annals of Statistics*, 43(5):2168–2197, 2015.

[40] Chao Gao, Zongming Ma, Harrison H Zhou, et al. Sparse cca: Adaptive estimation and computational barriers. *The Annals of Statistics*, 45(5):2074–2101, 2017.

[41] Fangyu Gao, Grace Wahba, Ronald Klein, and Barbara Klein. Smoothing spline anova for multivariate bernoulli observations with application to ophthalmology data. *Journal of the American Statistical Association*, 96(453):127–160, 2001.

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

[43] Francesca Grisoni, Viviana Consonni, Marco Vighi, Sara Villa, and Roberto Todeschini. Investigating the mechanisms of bioconcentration through qsar classification trees. *Environment international*, 88:198–205, 2016.

[44] Chong Gu. *Smoothing spline ANOVA models*, volume 297. Springer, 2013.

[45] László Györfi, Michael Kohler, Adam Krzyzak, and Harro Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

[46] Peter Hall and Jean D Opsomer. Theory for penalised spline regression. *Biometrika*, 92(1):105–118, 2005.

[47] Kam Hamidieh. A data-driven statistical model for predicting the critical temperature of a superconductor. *Computational Materials Science*, 154:346–354, 2018.

[48] Qiyang Han, Jon A Wellner, et al. Convergence rates of least squares regression estimators with heavy-tailed errors. *Annals of Statistics*, 47(4):2286–2319, 2019.

[49] Wolfgang Härdle, Gerard Kerkyacharian, Dominique Picard, and Alexander Tsybakov. *Wavelets, approximation, and statistical applications*, volume 129. Springer Science & Business Media, 2012.

[50] Asad Haris, Daniela Witten, and Noah Simon. Convex modeling of interactions with strong heredity. *Journal of Computational and Graphical Statistics*, 25(4):981–1004, 2016.

[51] Trevor Hastie. *gam: Generalized Additive Models*, 2019. R package version 1.16.1.

[52] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[53] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical learning with sparsity. *Monographs on statistics and applied probability*, 143:143, 2015.

[54] Eugenio Hernández and Guido Weiss. *A first course on wavelets*. CRC press, 1996.

[55] Joel Horowitz, Jussi Klemelä, and Enno Mammen. Optimal estimation in additive regression models. *Bernoulli*, 12(2):271–298, 2006.

[56] Daan Huybrechs, Arieh Iserles, et al. From high oscillation to rapid approximation iv: Accelerating convergence. *IMA Journal of Numerical Analysis*, 31(2):442–468, 2011.

[57] Graham James Oscar Jameson. *The prime number theorem*. Number 53. Cambridge University Press, 2003.

[58] Edward H Kennedy. Semiparametric theory and empirical processes in causal inference. In *Statistical causal inferences and their applications in public health research*, pages 141–167. Springer, 2016.

[59] Edward H Kennedy. Semiparametric doubly robust targeted double machine learning: a review. *arXiv preprint arXiv:2203.06469*, 2022.

[60] Rodney A Kennedy, Parastoo Sadeghi, Zubair Khalid, and Jason D McEwen. Classification and construction of closed-form kernels for signal representation on the 2-sphere. In *Wavelets and Sparsity XV*, volume 8858, page 88580M. International Society for Optics and Photonics, 2013.

[61] Jyrki Kivinen, Alex Smola, and Robert C Williamson. Online learning with kernels. *Advances in neural information processing systems*, 14:785–792, 2001.

[62] Andrei Nikolaevich Kolmogorov and Vladimir Mikhailovich Tikhomirov. ε-entropy and ε-capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.

[63] Alec Koppel, Garrett Warnell, Ethan Stump, and Alejandro Ribeiro. Parsimonious online learning with kernels via sparse projections in function space. *The Journal of Machine Learning Research*, 20(1):83–126, 2019.

[64] Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

[65] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

[66] Giovanni Leoni. *A first course in Sobolev spaces*. American Mathematical Soc., 2017.

[67] Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel" ridgeless" regression can generalize. *arXiv preprint arXiv:1808.00387*, 2018.

[68] Zhiyu Liang. *Eigen-analysis of kernel operators for nonlinear dimension reduction and discrimination*. PhD thesis, The Ohio State University, 2014.

[69] Xiwu Lin, Grace Wahba, Dong Xiang, Fangyu Gao, Ronald Klein, and Barbara Klein. Smoothing spline anova models for large data sets with bernoulli observations and the randomized gacv. *The Annals of Statistics*, 28(6):1570–1600, 2000.

[70] Yi Lin et al. Tensor product space anova models. *The Annals of Statistics*, 28(3):734–755, 2000.

[71] Fanghui Liu, Xiaolin Huang, Yudong Chen, and Johan AK Suykens. Random features for kernel approximation: A survey in algorithms, theory, and beyond. *arXiv preprint arXiv:2004.11154*, 2020.

[72] Zhenjuan Liu and Thanasis Stengos. Non-linearities in cross-country growth regressions: a semiparametric approach. *Journal of applied econometrics*, 14(5):527–538, 1999.

[73] Lennart Ljung and Torsten Söderström. *Theory and practice of recursive identification*. MIT press, 1983.

[74] Jing Lu, Steven CH Hoi, Jialei Wang, Peilin Zhao, and Zhi-Yong Liu. Large scale online kernel learning. *The Journal of Machine Learning Research*, 17(1):1613–1655, 2016.

[75] Zongming Ma, Yihong Wu, et al. Computational barriers in minimax submatrix detection. *The Annals of Statistics*, 43(3):1089–1116, 2015.

[76] Albert W Marshall, Ingram Olkin, and Barry C Arnold. *Inequalities: theory of majorization and its applications*, volume 143. Springer, 1979.

[77] Ulysse Marteau-Ferey, Francis Bach, and Alessandro Rudi. Globally convergent newton methods for ill-conditioned generalized self-concordant losses. In *Advances in Neural Information Processing Systems*, pages 7634–7644, 2019.

[78] Ulysse Marteau-Ferey, Dmitrii Ostrovskii, Francis Bach, and Alessandro Rudi. Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. *arXiv preprint arXiv:1902.03046*, 2019.

[79] Volker Michel. *Lectures on Constructive Approximation: Fourier, Spline, and Wavelet Methods on the Real Line, the Sphere, and the Ball*. Springer Science & Business Media, 2012.

[80] Piotr Mikusinski and Evan Weiss. The bochner integral. *arXiv preprint arXiv:1403.5209*, 2014.

[81] Arkadi Nemirovski. Topics in non-parametric. *Ecole d' Eté de Probabilités de Saint-Flour*, 28:85, 2000.

[82] A.S. Nemirovski and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience series in Discrete Mathematics. John Wiley, 1983.

[83] Erich Novak and Henryk Wozniakowski. *Tractability of multivariate problems. Vol. 1: Linear information*. 01 2008.

[84] Roland Opfer. Multiscale kernels. *Advances in computational mathematics*, 25(4):357–380, 2006.

[85] Michael L Overton. *Numerical computing with IEEE floating point arithmetic*. SIAM, 2001.

[86] Kaare Brandt Petersen and Michael Syskind Petersen. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

[87] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20:1177–1184, 2007.

[88] Alexander Rakhlin and Karthik Sridharan. Online nonparametric regression with general loss functions. *arXiv preprint arXiv:1501.06598*, 2015.

[89] E Rakotch et al. Numerical solution for eigenvalues and eigenfunctions of a hermitian kernel and an error estimate. *Math. Comput.*, 29:794–805, 1975.

[90] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research*, 13(2), 2012.

[91] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over l_q-balls. *IEEE transactions on information theory*, 57(10):6976–6994, 2011.

[92] Garvesh Raskutti, Bin Yu, and Martin J Wainwright. Lower bounds on minimax rates for nonparametric regression with additive sparsity and smoothness. In *Advances in Neural Information Processing Systems*, pages 1563–1570, 2009.

[93] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[94] Bellman Richard. Adaptive control processes: A guided tour. *Princeton, New Jersey, USA*, 1961.

[95] Bernard Rosner. *Fundamentals of biostatistics*. Cengage learning, 2015.

[96] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

[97] Veeranjaneyulu Sadhanala and Ryan J Tibshirani. Additive models with trend filtering. *The Annals of Statistics*, 47(6):3032–3068, 2019.

[98] Gabriele Santin and Robert Schaback. Approximation of eigenfunctions in kernel-based spaces. *Advances in Computational Mathematics*, 42(4):973–993, 2016.

[99] Hans-Jürgen Schmeisser. Recent developments in the theory of function spaces with dominating mixed smoothness. *Nonlinear Analysis, Function Spaces and Applications*, pages 145–204, 2007.

[100] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

[101] Jie Shen and Li-Lian Wang. Sparse spectral approximations of high-dimensional problems based on hyperbolic cross. *SIAM Journal on Numerical Analysis*, 48(3):1087–1109, 2010.

[102] Xiaotong Shen. On methods of sieves and penalization. *The Annals of Statistics*, pages 2555–2591, 1997.

[103] Jack Sherman and Winifred J Morrison. 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, 1950.

[104] Tao Shi, Mikhail Belkin, Bin Yu, et al. Data spectroscopy: Eigenspaces of convolution operators and clustering. *The Annals of Statistics*, 37(6B):3960–3984, 2009.

[105] Si Si, Sanjiv Kumar, and Yang Li. Nonlinear online learning with adaptive nystr\"{o} m approximation. *arXiv preprint arXiv:1802.07887*, 2018.

[106] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011.

[107] Noah Simon and Ali Shojaie. Convergence rates of nonparametric penalized regression under misspecified smoothness. *Statistica Sinica Preprint, No: SS-2018-0144*, 2018.

[108] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[109] Ingo Steinwart and Clint Scovel. Mercer's theorem on general domains: On the interaction between measures, kernels, and rkhss. *Constructive Approximation*, 35(3):363–417, 2012.

[110] Charles J Stone. Optimal rates of convergence for nonparametric estimators. *The annals of Statistics*, pages 1348–1360, 1980.

[111] Charles J Stone. Optimal global rates of convergence for nonparametric regression. *The annals of statistics*, pages 1040–1053, 1982.

[112] Charles J Stone. Additive regression and other nonparametric models. *The annals of Statistics*, pages 689–705, 1985.

[113] Hongwei Sun. Mercer theorem for rkhs on noncompact sets. *Journal of Complexity*, 21(3):337–349, 2005.

[114] Kean Ming Tan. Layer-wise learning strategy for nonparametric tensor product smoothing spline regression and graphical models. *Journal of machine learning research*, 20(119), 2019.

[115] Aliasghar Tarkhan and Noah Simon. Bigsurvsgd: Big survival data analysis via stochastic gradient descent. *arXiv preprint arXiv:2003.00116*, 2020.

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

[117] Gérald Tenenbaum. *Introduction to analytic and probabilistic number theory*, volume 163. American Mathematical Soc., 2015.

[118] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[119] Alexandre Tsybakov. *Introduction to Nonparametric Estimation*. Springer Science & Business Media, 2008.

[120] Sara van de Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

[121] Sara A Van de Geer. *Estimation and testing under sparsity*. Springer, 2016.

[122] Aad Van Der Vaart and Jon A Wellner. A local maximal inequality under uniform entropy. *Electronic Journal of Statistics*, 5(2011):192, 2011.

[123] Aad W Van Der Vaart and Jon A Wellner. Weak convergence. In *Weak convergence and empirical processes*, pages 16–28. Springer, 1996.

[124] Santosh S Vempala. *The random projection method*, volume 65. American Mathematical Soc., 2005.

[125] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[126] Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

[127] Grace Wahba, Yuedong Wang, Chong Gu, Ronald Klein, and Barbara Klein. Smoothing spline anova for exponential families, with application to the wisconsin epidemiological study of diabetic retinopathy: the 1994 neyman memorial lecture. *The Annals of Statistics*, 23(6):1865–1895, 1995.

[128] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

[129] Tengyao Wang, Quentin Berthet, Richard J Samworth, et al. Statistical and computational trade-offs in estimation of sparse principal components. *The Annals of Statistics*, 44(5):1896–1930, 2016.

[130] Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[131] Samuel George Waugh. *Extending and benchmarking Cascade-Correlation: extensions to the Cascade-Correlation architecture and benchmarking of feed-forward supervised artificial neural networks*. PhD thesis, University of Tasmania, 1995.

[132] Christopher Williams and Matthias Seeger. The effect of the input density distribution on kernel-based classifiers. In *Proceedings of the 17th international conference on machine learning*. Citeseer, 2000.

[133] Simon N Wood. *Generalized additive models: an introduction with R*. CRC press, 2017.

[134] Yunhua Xiang and Noah Simon. A flexible framework for nonparametric graphical modeling that accommodates machine learning. In *International Conference on Machine Learning*, pages 10442–10451. PMLR, 2020.

[135] Kui Xiong and Shiyuan Wang. The online random fourier features conjugate gradient algorithm. *IEEE Signal Processing Letters*, 26(5):740–744, 2019.

[136] Dongbin Xiu. *Numerical methods for stochastic computations: a spectral method approach*. Princeton university press, 2010.

[137] Yiming Ying and Massimiliano Pontil. Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, 8(5):561–596, 2008.

[138] Yiming Ying and D-X Zhou. Online regularized classification algorithms. *IEEE Transactions on Information Theory*, 52(11):4775–4788, 2006.

[139] Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pages 1975–1983, 2016.

[140] Ming Yuan, T Tony Cai, et al. A reproducing kernel hilbert space approach to functional linear regression. *The Annals of Statistics*, 38(6):3412–3444, 2010.

[141] Ming Yuan and Ding-Xuan Zhou. Minimax optimal rates of estimation in high dimensional additive models. *The Annals of Statistics*, 44(6):2564–2593, 2016.

[142] Tianyu Zhang and Noah Simon. An online projection estimator for nonparametric regression in reproducing kernel hilbert spaces. *arXiv preprint arXiv:2104.00780*, 2021.

[143] Tianyu Zhang and Noah Simon. A sieve stochastic gradient descent estimator for online nonparametric regression in sobolev ellipsoids. *arXiv preprint arXiv:2104.00846*, 2021.

[144] Yuchen Zhang, Martin J Wainwright, and Michael I Jordan. Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. In *Conference on Learning Theory*, pages 921–948, 2014.