On the Properties of Random Feature Methods

Zhu Li

Department of Statistics

Wolfson College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2020

The Book of Nature is written in the language of mathematics.

— Galileo Galilei

Declaration

I, Zhu Li, hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Name: Zhu Li

Date: October 2020

Acknowledgements

I would like to express my deepest gratitude to those who have supported me during my DPhil studies. First and foremost, I would like to thank my supervisors Professor David Steinsaltz and Professor Dino Sejdinovic. Although we did not have the chance to work together due to my shift in research interest, I would like to thank David for giving me the chance to study DPhil at University of Oxford. I would also like to thank Dino who has not only been a wonderful DPhil supervisor, but also a valuable mentor and friend. My DPhil studies have been a long journey during which I have had to overcome many unexpected challenges. No matter what difficulties I encounter, Dino was there to offer help. I am truly grateful to have him as my supervisor. He also gave me intellectual freedom to pursue my research ideas, while guiding me with his profound knowledge. His perspectives and suggestions have not only been proven valuable for our research works, but have also influenced the way I think about research. It is my great privilege to work with both David and Dino.

My DPhil studies would have been impossible without the opportunities to collaborate with many incredible colleagues. I would like to thank Dino Oglic, who put a huge amount of effort into our work, and helped improve the quality of our work significantly through his attention to details. I would also like to thank Professor Gustau Camps-Valls and Adrian Perez for their fruitful discussions and valuable suggestions during our collaboration. Finally, I would like to thank Professor Weijie J. Su, who taught me how to find interesting research topics and how to properly present the research findings. I feel extremely grateful to have worked with all my collaborators and I look forward to our future collaborations.

My life in Oxford has been greatly enriched by many friendly and interesting

colleagues in our department. I would like to give a huge thanks to Xiaoyu Lu, Qinyi Zhang, Ho Chung Leon Law, Lucian Chan, Jin Xu, Anthony Caterini, Fan Wu, Robert Hu, Alan Chau, Chao Zhang, and many others who have been with me. A special thanks goes to Jean-François Ton. I still remember when my project got stuck and I was about to fail my degree, he offered tremendous support and taught me how to use Github at 3 a.m. in the morning in our department library. I would also especially thank Nadia Ma, who spends enormously amount of time in proofreading my thesis, even though she knows nothing about statistics. I feel grateful to have all these great friends.

During my studies, I have had the opportunity to visit the Insititute of Automation in Chinese Academy of Science. I would like to thank Professor Liang Wang and Yan Huang for their help and advice on how to link my research with applications.

Last but not least, I would like to thank my parents for their love, patience, support and understanding no matter what decisions I made. I feel incredibly lucky to have them and feel grateful for all the emotional and financial support during my journeys studying abroad.

Abstract

In this thesis, we study the theoretical properties of the random feature methods, which are initially proposed to address the scalability issues of the kernel methods. Although empirically efficient, existing theoretical analysis of random feature methods often provide pessimistic bounds. Hence, this thesis starts with theoretical investigation of the generalization properties of random feature methods. Our first goal is to understand the statistical consistency of random feature estimators in various kernel learning frameworks. By adopting the kernel matrix approximation perspective, we demonstrate that random feature estimators asymptotically converge to the same estimators as the original kernel methods. This consistency property is valid for both kernel ridge regression and distribution regression settings.

Next, we explore the computation and statistical accuracy trade-off of random feature methods. We provide the first unified risk analysis of learning with random feature approximation for both regression and classification under various loss functions (such as Lipschitz continous loss and the 0-1 loss). We express the trade-off through the regularization parameter and the *number of effective degrees of freedom*. For regression problems, we theoretically demonstrate that random feature methods can provide us computational gain without incurring any prediction accuracy loss. Our study of standard random feature estimators improves the existing bounds on the number of features required to guarantee the minimax risk convergence rate. In addition, we show that a data-dependent sampling scheme of the random features based on *leverage scores* can further reduces the required number of features. For the classification problem with Lipschitz continuous loss and 0-1 loss, we first improve the existing bound on the required number of random features to achieve the minimax optimal learning rate. We then demonstrate that under appropriate assumptions on the data generating distribution,

it is possible to obtain a fast learning rate under both standard and leverage score sampling scheme. As ridge leverage scores are expensive to compute, we devise a simple approximation scheme which provably reduces the computational cost without loss of statistical efficiency. Our empirical results illustrate the effectiveness of the proposed scheme relative to the standard random features method.

Finally, we investigate the recent unusual *benign overfitting* phenomenon: overparameterized models has zero training loss but near-optimal prediction accuracy. Recent studies on benign overfitting reveal that the learning curve of the overparameterized model displays the *double descent* behavior, contradicting classical learning theory. Regarding the random feature model as a simple two-layer neural network with fixed first layer, we examine the conditions under which benign overfitting occurs. We adopt a new view of random feature and show that benign overfitting arises due to the noise which resides in such features (the noise may already be present in the data and propagate to the features or it may be added by the user to the features directly). The feature noise plays an important implicit regularization role in the benign overfitting phenomenon. In addition, we show that the overparameterized models may achieve the optimal learning rate in the minimax sense depending on the feature noise decay rate. Our study also reveals that the excess learning risk curve undergoes a multiple descent, instead of a double descent behavior.

Keywords— Random Features, Kernel Methods, Supervised Learning, Benign Overfitting

Table of Contents

List of Figures

List of Tables

List of Symbols

β_λ The RFF regression estimator with regularization parameter λ

$\gamma_{\mathcal{H}}(P, Q)$ The distance between probability measure P and Q in RKHS $\mathcal{H}$

$\hat{\mathbf{f}}_x^\lambda$ The in-sample prediction of the KRR estimator $\hat{f}^\lambda$

$\hat{\Sigma}$ The empririical cvariance operator of kernel k

$\hat{\Sigma}^s$ The empririical cvariance operator of the low rank approximate kernel $\tilde{k}$ with s random feature

$\hat{f}^\lambda$ The kernel ridge regression estimator with regularization parameter λ

$\hat{R}_n(\mathcal{H})$ The empirical Rademacher complexity of the function class $\mathcal{H}$

λ The regularization parameter

$\mathbb{E}(l_f)$ The expected risk of an estimator f

$\mathbb{E}_n(l_f)$ The empirical risk of an estimator f

$\mathbb{R}$ The set of real numbers

$\mathbf{B}_R$ The bias of the minimum norm least square estimator

$\mathbf{f}_x$ The in-sample prediction of an estimator f: $[f(x_1), \ldots, f(x_n))]$

$\mathbf{K}$ The Gram-matrix of kernel k

$\mathbf{M}_R$ The misspecification error of the minimum norm least square estimator

$\mathbf{V}_R$ The variance of the minimum norm least square estimator

$\mathbf{W}$ The Gaussian random matrix with each entry being i.i.d standard Gaussian ranom variable

$\mathbf{Z}$	The RFF feature matrix $\mathbf{Z} = [\mathbf{z}_{x_1}(\mathbf{v}), \ldots, \mathbf{z}_{x_n}(\mathbf{v})]^T$
$\mathbf{Z}_\xi$	The noisy feature matrix $\mathbf{Z} + \Xi$
$\mathbf{z}_x(\mathbf{v})$	The random feature vector $[z(v_1, x), \ldots, z(v_s, x)]$
$\mathcal{H}$	A Reproducing kernel Hilbert Space (RKHS)
$\mathcal{H}_0$	The unit ball of the RKHS $\mathcal{H}$
$\mathcal{X}$	A non-empty set
$\mathcal{Y}$	The label space
$\mathsf{M}_1^+(\mathcal{X})$	The set of all Borel probability measures defined on $\mathcal{X}$
μ_P	The kernel mean embedding of the probability distribution P
$\mu_{\hat{P}}$	The kernel mean embedding of the empirical distribution $\hat{P}$
$\boldsymbol{\Xi}$	The feature noise matrix $[\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_s]$
$\boldsymbol{\xi}$	The feature noise vector $[\xi_1, \ldots, \xi_s]$
Σ	The population cvariance operator of kernel k
$\tilde{\beta}^\lambda$	The regression coefficient of the RFFs approximation of $\hat{\mathbf{f}}_x^\lambda$
$\tilde{\mathbf{f}}_x^\lambda$	The RFFs in-sample prediction of $\hat{\mathbf{f}}_x^\lambda$
$\tilde{\mathbf{f}}_\beta^\lambda$	The in-sample prediction of label Y using RFFs approximation
$\tilde{\mathcal{H}}$	The RKHS spanned by the RFF
$\tilde{f}_\beta^\lambda$	The prediction function of RFFs approximation
$\tilde{k}$	The random Fourier feature (RFF) approximation of kernel k
ξ	The feature noise
$d_{\mathbf{K}}^\lambda$	The effective dimension of a learning problem with kernel k

$f_*(x)$ The optimal estimating regression function of a kernel ridge regression problem

$f_{\mathcal{H}}$ The estimator with minimal expected risk in RKHS $\mathcal{H}$

k A kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$

l The loss function

$L_2(d\tau)$ The space of square integrable functions with respect to the measure $d\tau$

$l_\lambda(v)$ The ridge leverage function

$p(v)$ The plain RFF sampling distribution

$P(x, y)$ The data generating distribution

$q^*(v)$ The optimal RFF sampling distribution

$R_n(\mathcal{H})$ The expected Rademacher complexity of the function class $\mathcal{H}$

$R_X(f)$ The excess learning risk of an estimator f

X The covariate matrix $[x_1, \ldots, x_n]^T$

Y The label vector $[y_1, \ldots, y_n]^T$

$z(v, x)$ The random feature of kernel k with frequency v: $k(x, y) = \int z(v, x)z(v, y)d\tau(v)$

Chapter 1 Introduction

This thesis attempts to provide a detailed study of the theoretical properties of random feature methods. The first part is dedicated to understanding the efficiency of the random features in various kernel supervised learning frameworks. In the second part, treating the random feature model as a two-layer neural network with fixed first layer weights, we explore the generalization properties of the overparameterized models as well as some unusual phenomena uncovered by modern deep learning.

Recent advances in computer hardware and information technology have dramatically decreased the cost of data collection and storage. This has led us to the era of "Big Data", where hundreds of terabytes of data are generated every day. The need to understand the massive data has motivated us to use much more complicated learning machinery in order to extract knowledge that could potentially impact not only commercial industries like healthcare, transportation, sports and advertising, but also our daily lives. When processing such large amount of data, a key quantity is the computational time, as we often employ complex models that would, if applied in their original form, require extremely large amounts of computational resources and time to train. A notable example is that of kernel methods (Cortes and Vapnik, 1995; Saunders et al., 1998; Schölkopf and Smola, 2001; Schölkopf et al., 2004; Shawe-Taylor and Cristianini, 2004; Szabó et al., 2016), which require at least $O(n^2)$ (where n is the sample size), or sometimes $O(n^3)$ computational cost. On the other hand, kernel methods have many desirable properties, including a flexibility to provide rich representations of our data as well as a complete theory on their statistical properties and theoretical guarantees (Caponnetto and De Vito, 2007; Steinwart and Christmann, 2008).

In light of this, the so called random Fourier features (RFFs) method (Rahimi and Recht, 2007) is proposed for the purpose of speeding up kernel methods. As we shall see in Chapter 2, the key idea of kernel methods is the "kernel trick", where we implicitly map our data $x \in \mathcal{X}$ into some high and often infinite dimensional Hilbert space $\mathcal{H}$ through a feature map $\varphi(\cdot) : \mathcal{X} \to \mathcal{H}$, and compute its inner product through the kernel evaluation $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$. While the feature map $\varphi(\cdot)$ gives us rich representations of the data, it also creates the computation bottleneck as it is typically infinite dimensional. As such, the RFFs propose to construct an explicit feature map $\phi(\cdot) : \mathcal{X} \to \mathbb{R}^s$, such that $\langle \phi(x), \phi(x') \rangle_{\mathbb{R}^s}$ is a good approximation of $\langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$. The construction of the explicit feature map can drive down the computation from $O(n^3)$ to $O(ns^2)$ provided $s \ll n$.

A number of empirical results have verified that the RFFs can significantly reduce the computational time in various kernel learning settings such as kernel ridge regression and kernel classification (Rahimi and Recht, 2007, 2009), hypothesis testing (Zhang et al., 2018), and principal component analysis (Ullah et al., 2018). However, the theoretical properties of the RFFs are only partially understood. Some outstanding questions remain as follows

- Are the estimators which arise from the RFFs approximation consistent, i.e., would they asymptotically give the same function as the original kernel methods?
- What is the trade-off between the computational time and statistical accuracy, i.e., to what extent can the RFFs save the computational cost without compromising statistical properties of the original kernel methods?

Apart from being a kernel approximation method, the RFFs have recently gained much research interest as they can also be treated as a simple two-layer neural network model with fixed first layer weights, and hence they can be used as an

initial step in the study of the generalization properties of modern deep learning models (Belkin et al., 2018; Jacot et al., 2018; Liao et al., 2020; Mei and Montanari, 2019) which reveals a surprising statistical phenomenon that overparameterized models can have optimal prediction accuracy. This finding contradicts classical learning theory which states that when choosing a model, one should always balance between the training loss and the model complexity in order to avoid overfitting and to achieve the best possible generalization, i.e., prediction accuracy on previously unseen examples. However, deep neural networks seem to have superior generalization ability even when they are overparameterized with millions of parameters and possess zero training loss. This *benign overfitting* phenomenon is recently characterized via the *double descent* (Belkin et al., 2019a) curves where the risk undergoes another descent (in addition to the classical U-shaped learning curve of classical learning theory when the number of parameters is small) as we increase the number of parameters beyond a certain threshold. The double descent and benign overfitting phenomena have spurred much research interest, (see, e.g., Bartlett et al., 2020; Belkin et al., 2019b; Chatterji and Long, 2020; Chinot and Lerasle, 2020; Liao et al., 2020; Mahdaviyeh and Naulet, 2019; Mei and Montanari, 2019). However, existing results only provide partial answers to how benign overfitting can happen with many restrictive assumptions. For example, many studies on the finite sample learning risk behavior or asymptotic risk behavior (Bartlett et al., 2020; Belkin et al., 2019b; Liao et al., 2020; Mei and Montanari, 2019) rely on the key assumption of either Gaussian data distribution or Gaussian kernel. Hence, we still lack a general understanding of how these phenomena happen.

In this thesis, we attempt to address the problems mentioned above in detail. In particular, we make the following contributions

- Adopting the kernel Gram-matrix approximation perspective, we first demon-

strate that the RFFs estimators are indeed consistent in various kernel learning frameworks.

- By directly studying the learning risk of the RFFs estimators, we devise a simple framework that allows us to provide a unified analysis of the trade-off between computational time and statistical accuracy for both regression and classification.
- Treating the random feature model as a simple two-layer neural network, we examine the conditions under which the double descent and benign overfitting phenomena can happen and point out that the feature noise serves as the implicit regularizer in supervised learning and plays an important role in these phenomena.

Thesis Structure

This thesis contains six chapters and follows an integrated format, where Chapter 4 and Chapter 5 contain either published papers or projects that are currently under review.

Chapter 2 introduces some background on kernel methods and its low rank approximation: the RFFs. We start with the definitions and properties of the reproducing kernel and its reproducing kernel Hilbert space (RKHS) and explain how these concepts can be employed in the supervised learning framework. After introducing the loss function and the learning risk, we formally define the kernel ridge regression and distribution regression frameworks. We also illustrate the concept of excess learning risk and the learning rate to evaluate the quality of these estimators. Finally, as kernel methods often suffer from prohibitive computational cost when applied to large datasets, we introduce the RFFs method, which is the core

study of this thesis, and explain how RFFs can be used to significantly reduce the computational cost.

Chapter 3 forms our initial attempt towards the theoretical understanding of the RFFs. We tackle the problem from the matrix approximation perspective, i.e., we look at the Gram-matrix approximation error due to the RFFs and analyze how this error is related to the number of features used in the approximation. In addition, we are interested in how this error can propagate into the learning risk both in the kernel ridge regression and distribution regression frameworks. The key questions that we would like to address in this chapter are: whether the RFFs estimators from the two regression settings are consistent and how the excess learning risk from the RFFs approximation behaves as a function of the number of features employed. Our results provide a basic analysis of the theoretical properties of the RFFs and are novel. However, soon after we derive the results of this chapter, some similar results[1] are published (Sutherland and Schneider, 2015). Moreover, our results derived later in Chapter 4 provide a more general and sharper analysis of the learning risk. Hence, we decide not to publish this analysis but leave this as an introductory chapter in our thesis. Under the guidance of Dino Sejdinovic, I contribute to the derivation of all the core theoretical results. The writing of the chapter is done by myself with revisions provided by Dino Sejdinovic.

Chapter 4 studies the RFFs estimation from the learning risk perspective, where we directly study the generalization properties of the RFFs in various kernel supervised learning setting. While our previous results in Chapter 3 do not provide any theoretical guarantees on the efficacy of the RFFs approximation, analysis in Chapter 4 derives a precise relationship between the learning rate of the RFFs estimators and the number of features required under both regression and classifi-

[1]To our knowledge, the published results are a weaker version of our results, please refer to Chapter 3 for more details.

cation frameworks. In addition, we also propose an algorithm that enables us to sample features according to the leverage score distribution. We both theoretically and empirically verify the efficiency of the algorithm. This is a joint work with Jean-Francois Ton, Dino Oglic and Dino Sejdinovic. Under the guidance of Dino Sejdinovic, I contribute all the derivations of the main theoretical results. The experiments in this chapter are a collaborative effort with Jean-Francois Ton. I draft the first version and edit the paper jointly with Dino Oglic and Dino Sejdinovic. Chapter 4 is based on the following published paper, which won the 2019 ICML Best Paper Honourable Mention Award

***Zhu Li**, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. "Towards a unified analysis of random Fourier features." In International Conference on Machine Learning, pp. 3905-3914. PMLR, 2019.*

The extended version of the paper (as presented in Chapter 4) has recently been published in the *Journal of Machine Learning Research (JMLR)*

***Zhu Li**, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards a unified analysis of random Fourier features. Journal of Machine Learning Research 22, no. 108 (2021): 1-51.*

In Chapter 5, we investigate the recently discovered phenomena of the double descent and benign overfitting in the context of random feature models. Modern deep learning has uncovered an interesting phenomenon that overparameterized or even interpolated models can provide optimal prediction accuracy, which contradicts classical learning theory. In this chapter, by incorporating the noise in the feature representations, we theoretically examine the conditions that possibly cause the double descent and benign overfitting phenomena. This is a joint work with Weijie Su and Dino Sejdinovic. Under the guidance of Dino Sejdinovic, I derive all the theoretical results and conduct the experiments. The first draft is written by myself

with revision provided by Weijie Su and Dino Sejdinovic. This chapter is based on the following project, which has been submitted to the *Journal of the Americal Statistical Association (JASA)* and is now under review.

***Zhu Li**, Weijie Su, and Dino Sejdinovic. Benign overfitting and noisy features. arXiv preprint arXiv:2008.02901, 2020.*

Chapter 6 provides a summary of our contributions towards the learning theory of random feature methods and points out some possible future research directions that might further our understanding of both random feature models and deep neural networks.

Finally, while I was reading DPhil at Oxford, I completed two other projects of a methodological nature: a project on fair learning with kernel methods which is under review at *Pattern Recognition (PR)*, and a project of distribution regression with correlated training samples, which will be submitted for publication in due course. However, for the sake of coherence, I decide not to include the two projects into the thesis and to maintain the focus on theoretical contributions to random feature methods.

Chapter 2 Background

In this chapter, we provide the necessary background for the kernel learning framework as well as the random Fourier features (RFFs) method. In particular, we start with introducing the concept of kernel and its corresponding reproducing kernel Hilbert space (RKHS), followed by a detailed exposition of kernel ridge regression and distribution regression. We close by explaining the theoretical background that the RFFs are based on.

2.1 Kernels and RKHS

In this section, we introduce the concept of kernel which has been widely used in the machine learning literature (see, e.g., Bishop, 2006; Boser et al., 1992; Cortes and Vapnik, 1995; Friedman et al., 2001; Hofmann et al., 2008; Rasmussen and Williams, 2006; Schölkopf et al., 2002; Vapnik, 2013). As is common in the machine learning applications, we restrict our attention to real-valued kernels.

2.1.1 Positive Definite Function and Kernels

Definition 2.1. *(Steinwart and Christmann, 2008, Definition 4.1)*
Let $\mathcal{X}$ be a non-empty set. The function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel if there exists a $\mathbb{R}$-Hilbert space and a map $\Phi : \mathcal{X} \to \mathcal{H}$ such that for all $x, y \in \mathcal{X}$ we have

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}.$$

In addition, we call Φ the feature map and $\mathcal{H}$ the feature space.

While there are various ways to construct kernels (refer to Steinwart and Christmann (2008) and Berlinet and Thomas-Agnan (2011)), in general, to see whether a given

function is a kernel through finding a feature space is difficult. Hence, we rely on the notion of positive definite function which characterizes kernel functions through certain inequalities. Below we present the definition of a positive definite function.

Definition 2.2. *(Steinwart and Christmann, 2008, Definition 4.15)*
A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called positive definite if, for all $n > 1$, $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and $x_1, \ldots x_n \in \mathcal{X}$, we have

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0.$$

In addition, k is said to be strictly positive definite if for all mutually distinct x_i, the equality only holds when all α_i are zero.

It turns out that the two notions are equivalent as given by the following theorem.

Theorem 2.1. *(Steinwart and Christmann, 2008, Theorem 4.16)*
The following two statements are equivalent

1. *$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel;*
2. *k is symmetric and positive definite.*

One advantage of employing kernel is that we can avoid constructing the explicit feature map Φ, a property called *kernel trick* (Muandet et al., 2016; Schölkopf et al., 2002). In addition, the kernel trick can not only be applied to Euclidean data, but also to non-Euclidean structured data, functional data, and other domains (Gärtner, 2003). Because of these nice properties, kernels have been widely used in machine learning community. A review of classes of kernel functions exploited in practice can be found in Genton (2001). For particular applications of kernels such as support vector machine or Gaussian processes, please refer to Burges (1998);

Cucker and Smale (2002); Rasmussen and Williams (2006).

2.1.2 Reproducing Kernel and RKHS

We introduce the reproducing kernel and the RKHS (Aronszajn, 1950; Berlinet and Thomas-Agnan, 2011; Steinwart and Christmann, 2008) in this section.

Definition 2.3. *(Reproducing Kernel (Steinwart and Christmann, 2008))*
Let $\mathcal{H}$ be a real-valued Hilbert function space over $\mathcal{X}$, i.e., it is a space that consists of functions mapping from $\mathcal{X}$ into $\mathbb{R}$. Then a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a reproducing kernel of $\mathcal{H}$ if it satisfies that

1. $\forall x \in \mathcal{X}, k(\cdot, x) \in \mathcal{H}$;
2. $\forall x \in \mathcal{X}, \forall f \in \mathcal{H}, f(x) = \langle f, k(\cdot, x)\rangle_{\mathcal{H}}$.

The definition of an RKHS is presented below

Definition 2.4. *(RKHS (Steinwart and Christmann, 2008))*
A Hilbert space $\mathcal{H}$ of real-valued functions $f : \mathcal{X} \to \mathbb{R}$ over a non-empty set $\mathcal{X}$ is said to be an RKHS if the evaluation functional δ_x is continuous $\forall x \in \mathcal{X}$.

After we describe the reproducing kernel and the RKHS, we would like to investigate their relationship. The following lemma from Steinwart and Christmann (2008) indicates this relationship.

Lemma 2.1. *(Steinwart and Christmann, 2008, Lemma 4.19)*
$\mathcal{H}$ is an RKHS if and only if $\mathcal{H}$ is a Hilbert function space over a non-empty space $\mathcal{X}$ that has a reproducing kernel k. In addition, $\mathcal{H}$ is the feature space of k with the feature map $\Phi : \mathcal{X} \to \mathcal{H}$ defined as

$$\Phi(x) = k(\cdot, x), x \in \mathcal{X}.$$

Lemma 2.1 shows that the reproducing kernels are kernels, where for any $x, y \in \mathcal{X}, k(x, y) = \langle k(\cdot, x), k(\cdot, y)\rangle_{\mathcal{H}}$. It further proves that a Hilbert space is an RKHS if and only if it has a reproducing kernel. Moreover, one can show that there is a one-to-one relationship between reproducing kernel and RKHS (Aronszajn, 1950; Steinwart and Christmann, 2008): for every reproducing kernel, there is a unique RKHS and every RKHS has a unique reproducing kernel.

A kernel often has properties such as boundedness, measurability, and continuity. Hence, we are also interested in whether functions in its associated RKHS share the same properties, which the next two lemmas illustrate.

Lemma 2.2. *(Steinwart and Christmann, 2008, Lemma 4.23)*
Let $\mathcal{X}$ be a non-empty set and $\mathcal{H}$ be an RKHS with reproducing kernel k. Then the following two statements are equivalent

1. *k is bounded ($\exists M > 0$, such that $|k(x, y)| \leq M, \forall x, y \in \mathcal{X}$);*
2. *f is bounded, $\forall f \in \mathcal{H}$ ($\exists M > 0$, such that $|f(x)| \leq M, \forall x \in \mathcal{X}$).*

Lemma 2.3. *(Steinwart and Christmann, 2008, Lemma 4.29)*
Let $\mathcal{X}$ be a non-empty set and k be a kernel on $\mathcal{X}$ with feature space $\mathcal{H}$ and feature map $\Phi : \mathcal{X} \to \mathcal{H}$, then the following statements are equivalent

1. *k is continuous.*
2. *Φ is continuous.*

2.1.3 The Characteristic Kernel

One of the application of kernel methods is to compute the distance between probability measures, which is of paramount importance in probability theory and statistics, and has many applications including hypothesis testing (Gretton et al., 2012), density estimation (Song et al., 2008) and hidden Markov models (Song

et al., 2013). One application of our thesis, i.e., distribution regression in Section 2.2.3, relies heavily on computing the distance between probability measures. As a result, this section introduces one popular way of characterizing distances between probability measures: the Hilbert space embedding of probability measure and some properties of the distance.

Let $\mathcal{X}$ be a non-empty set and $\mathcal{A}$ the associated σ-algebra. Denote by $\mathsf{M}_1^+(\mathcal{X})$ the set of all Borel probability measures on $(\mathcal{X}, \mathcal{A})$. Let $\mathcal{H}$ be an RKHS with k as its reproducing kernel, then we compute the distance between $P, Q \in \mathsf{M}_1^+(\mathcal{X})$ as

$$\gamma_{\mathcal{H}}(P, Q) = \sup_{||f||_{\mathcal{H}} \leq 1} \left| \int_{\mathcal{X}} f dP - \int_{\mathcal{X}} f dQ \right|.$$

Since there is a one-to-one relationship between RKHS and kernel k, we denote the distance by $\gamma_k(P, Q)$. The computation of γ_k can be further simplified, provided that k is bounded (Sriperumbudur et al., 2010).

Theorem 2.2. *(Sriperumbudur et al., 2010, Theorem 1 and Proposition 2)*
Let $\mathcal{X}$ be a non-empty set, k is a bounded and measurable kernel defined on $\mathcal{X}$ with $\mathcal{H}$ as the associated RKHS. Then for any $P, Q \in \mathsf{M}_1^+(\mathcal{X})$,

$$\gamma_k(P, Q) = \left\| \int_{\mathcal{X}} k(\cdot, x) dP(x) - \int_{\mathcal{X}} k(\cdot, x) dQ(x) \right\|_{\mathcal{H}}.$$

After we can compute the distance between probability measures, i.e., γ_k, a natural question to ask is whether we can choose a kernel k such that γ_k is a metric on $\mathsf{M}_1^+(\mathcal{X})$. In other words, $\gamma_k(P, Q) = 0$ implies $P = Q$. Fortunately, extensive studies (see, e.g., Fukumizu et al., 2009, 2007; Gretton et al., 2006, and references therein) are devoted to finding such a kernel (the characteristic kernel). Below we formally define the characteristic kernel and state the theorems to help us examine whether a kernel is characteristic.

Definition 2.5. *(Characteristic kernel, Sriperumbudur et al. (2010))*
Let $\mathcal{X}$ be a non-empty set, k be a bounded measurable kernel defined on $\mathcal{X}$ and $\mathsf{M}_1^+(\mathcal{X})$ be the set of all probability measures defined on $(\mathcal{X}, \mathcal{A})$. Then k is said to be characteristic to a set $\mathcal{Q} \subset \mathsf{M}_1^+(\mathcal{X})$, if for $P, Q \in \mathcal{Q}$, $\gamma_k(P, Q) = 0 \Leftrightarrow P = Q$. The RKHS induced by such a kernel k is called a characteristic RKHS.

Sriperumbudur et al. (2010) thoroughly examine the conditions under which a kernel k is characteristic. Since our primary interest is in translation invariant kernels, we only provide the conditions that are related to these kernels.

Theorem 2.3. *(Sriperumbudur et al., 2010, Theorem 9)*
Let k be a translation invariant kernel and has Fourier transform Λ, i.e., $k(x, y) = \psi(x - y)$, where ψ is a real-valued bounded continuous positive definite function defined on $\mathbb{R}^d$, then k is characteristic if and only if supp$(\Lambda) = \mathbb{R}^d$.

Building on Theorem 2.3, Corollary 10 in Sriperumbudur et al. (2010) further improves the results and demonstrates that the translation invariant kernel k is characteristic even if supp(Λ) is compact.

2.1.4 Kernel Mean Embedding

One of the advantage of characteristic kernels is that they enable a one-to-one embedding of a probability distribution into the RKHS, i.e., the kernel mean embedding. The idea behind kernel mean embedding is to transform the distribution into one element of the RKHS such that all equipment in kernel methods can be extended into dealing with probability measures. Kernel mean embeddings are used in many applications, which include independence and conditional independence tests (Doran et al., 2014; Fukumizu et al., 2008; Zhang et al., 2011), adaptive MCMC (Sejdinovic et al., 2014), filtering for dynamical system (Song et al., 2009) and kernel Bayes' Rule (Fukumizu et al., 2011). Because of its enormous

applications, we introduce the kernel mean embedding in this section.

We let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel defined on the input space, called the **data kernel**. Let $\mathcal{H}_k$ be the associated RKHS (Steinwart and Christmann, 2008). With the data kernel k, we define the set of mean embeddings for probability measures in $\mathsf{M}_1^+(\mathcal{X})$ by

$$\mu_{\mathcal{X}} = \mu(\mathsf{M}_1^+(\mathcal{X})) = \{\mu_P : P \in \mathsf{M}_1^+(\mathcal{X})\},$$

where the mean embedding for a particular distribution P is defined as

$$\mu_P = \int_{\mathcal{X}} k(\cdot, x) dP(x) = \mathbb{E}_P[k(\cdot, x)]. \tag{2.1}$$

The use of mean embedding is only interesting if it is in the RKHS. Hence, we would like to investigate the conditions to satisfy that. It turns out that the conditions are easy to check, as is illustrated in the following lemma.

Lemma 2.4. *(Smola et al., 2007)*
If $\mathbb{E}_P[\sqrt{k(x,x)}] < \infty$*, then we have* $\mu_P \in \mathcal{H}_k$ *and* $\mathbb{E}_P[f(x)] = \langle f, \mu_P \rangle_{\mathcal{H}_k}$.

As we can see, the kernel mean embedding is a vector in the Hilbert space which preserves all the information about the distribution. In addition, it represents the expectation of any function in the RKHS in the form of the inner product $\int h(x)P(dx) = \langle h, \mu_P \rangle_{\mathcal{H}_k}$. In the case where we use the *characteristic* kernel (Sriperumbudur et al., 2011), every probability measure has a unique embedding, and hence, μ_P completely determines a probability measure.

In practice, we often do not have access to the true underlying distribution P, which causes difficulties in estimating the mean embedding. As a result, we rely on the samples generated from P to perform inference. Given an i.i.d sample $\{x_i\}_{i=1}^n$,

the most common one that we use is

$$\mu_{\hat{P}} := \frac{1}{n}\sum_{i=1}^{n} k(x_i, \cdot),$$

where the use of $\hat{P}$ indicates the estimation comes from the empirical distribution. It is easy to see that the empirical estimator is unbiased and by the law of large numbers, $\mu_{\hat{P}}$ converges to μ_P as $n \to \infty$. The convergence rate is thoroughly studied by Song (2008) and Lopez-Paz et al. (2015), which state that the convergence happens at a rate of $O(n^{-1/2})$. Moreover, $\mu_{\hat{P}}$ is demonstrated to be the minimax optimal estimator of μ_P by Tolstikhin et al. (2017).

To conclude, we introduce the real-valued reproducing kernel and its associate RKHS in this section. However, we remark that the vector-valued RKHS is proposed in literature for finite dimensional spaces (Micchelli and Pontil, 2005), infinite dimensional spaces (Caponnetto et al., 2008; Carmeli et al., 2010), and is widely used in many applications (see, e.g., Kadri et al., 2016, 2012; Lian, 2007). For interested readers, we suggest Alvarez et al. (2012) for a comprehensive review. Due to the nature of the problems studied in the thesis, we will hereafter mainly focus on real-valued RKHS unless specified.

2.2 Supervised Learning with Kernels

Kernel methods have wide applications in supervised learning, which is one of the most important task in machine learning. As a result, we formally introduce supervised learning with kernels in this section. Let $\mathbf{x}$ and $\mathbf{y}$ be two random variables with $P(x, y) = P_{\mathbf{x}}(x)P_{\mathbf{y}}(y|x)$ as its joint probability distribution on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y}$ is a label space. A training data is a set of examples $\{(x_i, y_i)\}_{i=1}^{n}$

sampled independently from $P(x, y)$. The distribution $P_{\mathbf{x}}$ [1] is called the marginal distribution of a data-generating process. The goal of supervised learning is to find a function $f\colon \mathcal{X} \to \mathcal{Y}$ from a hypothesis space $\mathcal{H}$, such that $f(x)$ is a good estimate of the label $y \in \mathcal{Y}$ corresponding to a previously unseen instance $x \in \mathcal{X}$. In this section, we briefly introduce the concept of supervised learning as well as the associated kernel framework.

2.2.1 Risk and Regularization

Since supervised learning aims to find the best $f(x)$ to estimate y, we need a formal way to define the quality of the estimation. For the assessment of the estimation at a particular point x, this is done by the concept of the loss function.

Definition 2.6. *(Definition 2.1 Steinwart and Christmann (2008))*
Let $(\mathcal{X}, \mathcal{A})$ be a measurable space, and $\mathcal{Y} \subset R$ is a closed subset. Then $l : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$ is called a loss function, if it is measurable.

The loss function provides a formal way to assess the quality of predicting y_i by $f(x_i)$. The smaller the loss, the better the prediction at a given input x_i. However, the true goal of learning is to find the estimator $f \in \mathcal{H}$ that possesses the lowest loss across the whole instance space $\mathcal{X}$. Hence, we define the learning risk below.

Definition 2.7. *(Definition 2.2 Steinwart and Christmann (2008))*
Let $l : \mathcal{X} \times \mathcal{Y} \to [0, \infty)$ be a loss function and denote $l_f = l(f(x), y)$. Then if the function $f : \mathcal{X} \to \mathbb{R}$ is measurable, the risk of f is defined as

$$\mathbb{E}(l_f) := \int_{\mathcal{X}\times\mathcal{Y}} l(f(x), y) dP(x, y).$$

[1] Later in the thesis, when the context is clear, we will drop the math font and use $P(x, y)$ and P_x to denote the data generating distribution and the marginal distribution respectively.

As discussed, $\mathbb{E}(l_f)$ gives a global quality assessment of the estimator f. Hence, we would like to choose the f from the hypothesis space $\mathcal{H}$ with the lowest $\mathbb{E}(l_f)$ possible. However, in many learning tasks, the access to P is only through the training samples $\{(x_i, y_i)\}_{i=1}^n$. Hence we rely on the empirical measures to define the empirical risk

$$\mathbb{E}_n(l_f) := \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i).$$

The empirical risk $\mathbb{E}_n(l_f)$ provides an estimate of the quality of f through the training samples. Therefore, in supervised learning, we often formulate the learning task as an optimization problem that finds the estimator $f \in \mathcal{H}$ with minimal empirical risk. However, the hypothesis space $\mathcal{H}$ is usually quite large, and potentially can be infinite dimensional. As a result, simply reducing the training loss often leads to the problem of overfitting (estimator with low or zero training loss but with poor prediction on unseen data). In order to avoid this problem, the regularization technique is introduced so that the objective function in learning becomes

$$\mathbb{E}_n(l_f) + \lambda\Omega(f), \tag{2.2}$$

where $\Omega(f)$ is the penalty function that prevents the estimation function f becoming too complex. A typical example is $\|f\|_{\mathcal{H}}^2$. The parameter λ controls the relative importance of the empirical risk term $\mathbb{E}_n(l_f)$ and the regularization term $\Omega(f)$, i.e., if λ gets larger, the estimation prefers functions with small complexity. Learning problems defined as Eq. (2.2) are often referred as the regularized empirical risk minimization (ERM). For more details of the regularized ERM, please refer to Chapter 3 in Steinwart and Christmann (2008).

Depending on the loss function used, this regularized ERM framework can be utilized for both regression and classification. In regression, we often use the square loss function: $l(y, f(x)) = (y - f(x))^2$, absolute loss function $l(y, f(x)) = |y - f(x)|$, etc., whereas for classification, loss functions can be hinge loss $l(y, f(x)) = (1 - yf(x))_+$, logistic loss $l(y, f(x)) = \log(1 + \exp(-yf(x)))$, 0-1 loss $l(y, f(x)) = \mathbf{1}_{\{f(x)=y\}}$, etc. In this thesis, we will mainly focus on square loss function in the regression setting, and Lipschitz continuous loss and 0-1 loss (refer to Chapter 4 for more details) in the classification setting.

2.2.1.1 Kernel Supervised Learning

Kernel supervised learning is defined as a supervised learning task with a kernel function k and the associated RKHS $\mathcal{H}$. The objective is similar, where we find a hypothesis $f : \mathcal{X} \to \mathcal{Y}$ such that $f \in \mathcal{H}$ and $f(x)$ is a good estimate of the label $y \in \mathcal{Y}$. While in regression tasks $\mathcal{Y} \subset \mathbb{R}$, in classification tasks we typically have $\mathcal{Y} = \{-1, 1\}$. Kernel supervised learning can be described as the following empirical risk minimization problem

$$\begin{aligned} \hat{f}^{\lambda} &:= \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} l(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}}^2 \\ &= \arg\min_{\alpha} \frac{1}{n} \sum_{i=1}^{n} l(y_i, (\mathbf{K}\alpha)_i) + \lambda \alpha^T \mathbf{K} \alpha, \end{aligned} \tag{2.3}$$

where $f = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$ with $\alpha \in \mathbb{R}^n$, $\mathbf{K}$ is the kernel Gram-matrix, and λ is the regularization parameter. Note that Eq. (2.3) is due to the representer theorem (Schölkopf and Smola, 2001).

In kernel supervised learning, similar to Rudi and Rosasco (2017) and Caponnetto and De Vito (2007), we often assume [2] the existence of $f_{\mathcal{H}} \in \mathcal{H}$ such that $f_{\mathcal{H}} =$

[2] The existence of $f_{\mathcal{H}}$ depends on the complexity of $\mathcal{H}$ which is related to the data distribution $P(x, x)$. For more details, please see Caponnetto and De Vito (2007) and Rudi and Rosasco (2017).

$\inf_{f\in\mathcal{H}} \mathbb{E}(l_f)$, in order to analyze the statistical property of the estimator $\hat{f}^\lambda$. The assumption implies that there exists some ball of radius $R > 0$ containing $f_\mathcal{H}$ in its interior. As we can see later, our theoretical results in Chapter 3 and 4 do not require prior knowledge of this constant and hold uniformly over all finite radii.

Furthermore, for all the estimators returned by supervised learning algorithm, we assume that they have bounded RKHS norm. As a result, to simplify our derivations and constant terms in our bounds, we assume (without loss of generality) that all the estimators appear in the rest of the thesis are within the **unit ball** of the RKHS.

Note that $\mathbb{E}(l_{f_\mathcal{H}})$ is the lowest learning risk one can achieve in the RKHS $\mathcal{H}$. Hence, the theoretical studies of the estimator $\hat{f}^\lambda$ often concern how fast its learning risk $\mathbb{E}(l_{\hat{f}^\lambda})$ converges to $\mathbb{E}(l_{f_\mathcal{H}})$, in other words, how fast the excess risk $\mathbb{E}(l_{\hat{f}^\lambda}) - \mathbb{E}(l_{f_\mathcal{H}})$ converges to zero. In the remainder of the thesis, we will treat the rate at which the excess risk converges to zero as the learning rate. The convergence rate of excess risk is extensively studied in literature. For details please refer to Bartlett et al. (2006); Caponnetto and De Vito (2007); Friedman et al. (2001); Steinwart and Christmann (2008) and references therein.

2.2.2 Kernel Ridge Regression

In this section, we consider the supervised learning problem where response variable y is real-valued and the loss function l is the squared loss.

Recall that we denote x and y to be the two random variables defined on $\mathcal{X} \times \mathcal{Y}$. In the regularized ERM with the squared loss, the optimal estimating regression function is given by

$$f_*(x) = \mathbb{E}(\mathsf{y}|\mathsf{x} = x).$$

Let $X = [x_1, \ldots, x_n]^T$ and $Y = [y_1, \cdots, y_n]^T$ denote the training inputs and

outputs. An important class of regularized ERM problems is the kernel ridge regression (KRR) which we describe next.

Definition 2.8. *(Kernel Ridge Regression (KRR))*
Given training example $\{(x_i, y_i)\}_{i=1}^n$ *from* P*, a kernel* k *and its corresponding RKHS* $\mathcal{H}$*, KRR problem is the regularized ERM in Eq.(2.2) with* l *being the squared loss and* $\Omega(\cdot)$ *being the squared RKHS norm*

$$\hat{f}^\lambda := \arg\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\|f\|_{\mathcal{H}}^2. \tag{2.4}$$

Applying the representer theorem (Steinwart and Christmann, 2008, Theorem 5.5), the solution to Eq.(2.4) can be written as $\hat{f}(x) = k(x, X)(\mathbf{K} + n\lambda I)^{-1}Y$, where $k(x, X) = [K(x, x_1), \ldots, K(x, x_n)]^T$ and $\mathbf{K}_{i,j} = k(x_i, x_j)$ is the Gram-matrix.

The convergence of kernel ridge regression is studied extensively in the literature. See for example De Vito et al. (2005); Geer (2000); Kohler and Krzyzak (2001); Vito et al. (2005); Wahba (1990) and references therein. In particular, Caponnetto and De Vito (2007) provide a comprehensive analysis of the learning rate for KRR by restricting the data generating distribution to come from the $\mathcal{P}(b, c)$ family, where $b \in [1, \infty)$ and $c \in [1, 2]$ are some parameters concerning the effective dimension and complexity of the RKHS corresponding to the kernel k (for details of the $\mathcal{P}(b, c)$ family, please refer to Caponnetto and De Vito (2007)).

2.2.3 Distribution Regression

In many machine learning applications, modelling the training data as a probability measure is often more convenient than points in some space. Hence, in this section, we will consider the task of learning a regression function from a set to a single label. For example, we might want to learn the sentiment based on each paragraph

where we treat each paragraph as a specific distribution of words. Another setting would be in classical regression problem where we have noisy input. In fact, many learning problem can be categorized as learning from distributions such as multiple instance learning (Doran, 2013) and group anomaly detection (Muandet and Schölkopf, 2013; Póczos et al., 2012; Xiong et al., 2011). Due to its broad applications, we formally introduce the distribution regression framework in this section.

2.2.3.1 Notation

Let $(\mathcal{X}, \tau)$ be a topological space and let $\mathcal{B}(\mathcal{X})$ be the Borel σ-algebra induced by the topology τ. Let $\mathsf{M}_1^+(\mathcal{X})$ denote the set of Borel probability measures on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. Given measurable spaces $(\mathcal{X}_1, \mathcal{B}(\mathcal{X}_1))$, $(\mathcal{X}_2, \mathcal{B}(\mathcal{X}_2))$, denote by $\mathcal{B}(\mathcal{X}_1) \bigotimes \mathcal{B}(\mathcal{X}_2)$ the product σ-algebra (Steinwart and Christmann, 2008). For data $\{x_1, x_2, \ldots, x_n\} \overset{\text{i.i.d}}{\sim} P$, denote by $\hat{P} := \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$ as the empirical measure, where δ_x is a Dirac measure supported on $x \in \mathcal{X}$.

2.2.3.2 Kernels on Probability Distributions

As we shall see in Section 2.2.3.3, distribution regression is solved via kernel mean embedding introduced in Section 2.1.4. The basic idea is to map a distribution P into an RKHS and to treat the embedding μ_P as the feature to perform kernel ridge regression. Hence, on top of the kernel k (**data kernel**) used to conduct kernel mean embedding, we also need another layer of kernel K, often termed as **bag kernel**, for the purpose of regression.

Specifically, let $K : \mu_{\mathcal{X}} \times \mu_{\mathcal{X}} \to \mathbb{R}$ be the kernel defined on the set of mean embeddings $\mu_{\mathcal{X}} = \{\mu_P : P \in \mathsf{M}_1^+(\mathcal{X})\}$, where the associated RKHS is $\mathcal{H}_K$. We

refer to this kernel as the bag kernel. One typical example of the bag kernel is

$$K(P,Q) := \langle \mu_P, \mu_Q \rangle_{\mathcal{H}_K} = \int \int_{\mathcal{X}} k(x,y) dP(x) dQ(y). \tag{2.5}$$

And the empirical estimation of (2.5) is

$$K(\hat{P},\hat{Q}) \approx \langle \mu_{\hat{P}}, \mu_{\hat{Q}} \rangle_{\mathcal{H}_K} = \frac{1}{N^2} \sum_{i,j=1}^{N} k(x_i, y_j),$$

assuming we have samples $x_1, \cdots, x_N \overset{\text{i.i.d}}{\sim} P$ and $y_1, \cdots, y_N \overset{\text{i.i.d}}{\sim} Q$ respectively.

2.2.3.3 Learning in Distribution Regression

We assume that there is a meta distribution $\mathcal{M}$ defined on the product space $(\mathsf{M}_1^+(\mathcal{X}) \times Y, S_1 \bigotimes \mathcal{B}(Y))$ where S_1 is the σ-algebra for $\mathsf{M}_1^+(\mathcal{X})$. The data $\mathbf{z} = \{(P_i, y_i)\}_{i=1}^l$ are drawn i.i.d from the distribution $\mathcal{M}$ with $P_i \in \mathsf{M}_1^+(\mathcal{X}), y_i \in Y$. We collect data $x_{i,1}, \ldots, x_{i,N_i} \overset{\text{i.i.d}}{\sim} P_i$. Hence, the actual dataset is $\hat{\mathbf{z}} = \{(\{x_{i,n}\}_{n=1}^{N_i}, y_i)\}_{i=1}^m$. For notational simplicity, we assume that the number of the collected data for each distribution is the same. Our goal is to estimate f based on $\hat{\mathbf{z}}$.

As in the classical regression problem, distribution regression can be solved by the ridge regression method. However, a key difference is that the distribution regression problem involves two levels of randomness, where the first level is that $\mathbf{z}$ is randomly drawn from $\mathcal{M}$ and the second level is that the observed sample is generated by sampling N data points from distribution P_i. As such, the Mean Embedding based Ridge Regression (MERR) method is used to address the two-stage sampling problem. Such a method is thoroughly analyzed in Szabó et al. (2016). Specifically, we first map each distribution P_i into the mean embedding set $\mu_{\mathcal{X}}$ via the mean embedding function μ with the data kernel k. We then obtain

the new dataset $\mathbf{z}_{\text{new}} = (\mu_{P_i}, y_i)_{i=1}^m$. Finally, we conduct the usual kernel ridge regression via the bag kernel K.

If we use the squared loss function and define the expected risk for a measurable function $f : \mathcal{X} \to Y$ as

$$\mathbb{E}(l_f) = \mathbb{E}_{\mathcal{M}} \|f(\mu_P) - y\|^2,$$

then the MERR is defined as following

Definition 2.9. *(Mean Embedding based Ridge Regression (MERR))*
Given training example $\mathbf{z} = \{(P_i, y_i)\}_{i=1}^m$*, a data kernel* k *and the bag kernel* K*, MERR problem is the regularized ERM in Eq.(2.2) with the square loss defined below*

$$f_z^\lambda = \arg\min_{f \in \mathcal{H}_K} \frac{1}{m} \sum_{i=1}^m \|f(\mu_{P_i}) - y_i\|^2 + \lambda \|f\|_{\mathcal{H}_K}^2.$$

However, as we mentioned before, direct access to the sample $\mathbf{z}$ is not possible as each P_i is a distribution. Instead, we minimize the function defined through the observed sample $\hat{\mathbf{z}}$

$$f_{\hat{z}}^\lambda = \text{argmin}_{f \in \mathcal{H}_K} \frac{1}{m} \sum_{i=1}^m \|f(\mu_{\hat{P}_i}) - y_i\|^2 + \lambda \|f\|_{\mathcal{H}_K}^2. \tag{2.6}$$

This ridge regression problem in (2.6) admits an analytical solution. For the new empirical test distribution $\hat{P}_{m+1}$, the prediction is

$$f_{\hat{z}}^\lambda(\mu_{\hat{P}_{m+1}}) = \mathbf{k}_{\hat{P}_{m+1}} (\mathbf{K} + m\lambda I)^{-1} Y, \tag{2.7}$$

where $\mathbf{k}_{\hat{P}_{m+1}} = [K(\mu_{\hat{P}_1}, \mu_{\hat{P}_{m+1}}), \ldots, K(\mu_{\hat{P}_m}, \mu_{\hat{P}_{m+1}})]^T$, $\mathbf{K} = [K(\mu_{\hat{P}_i}, \mu_{\hat{P}_j})]_{i,j \in m}$,

and $Y = [y_1, \ldots, y_m]^T$.

After we have the prediction for the new distribution $\hat{P}_{m+1}$, one natural question to ask is what the inferential properties of such prediction are. Szabó et al. (2016) provide a detailed analysis of the excess learning risk of the MERR. By assuming that the probability distribution $\mathcal{M}$ is within the $\mathcal{P}(b, c)$ class and under some mild conditions, Szabó et al. (2016) demonstrate the statistical consistency of the MERR and provide an exact computational-statistical efficiency trade-off for the MERR estimator.

2.3 Random Feature Methods

Kernel methods are very flexible in terms of modelling nonlinear functional relationships. However, kernel methods often require the storing and inverting the Gram-matrix $\mathbf{K}$, which are $O(n^2)$ storage and $O(n^3)$ computation operations respectively. This is not scalable even for a moderately sized dataset. Hence, numerous methods have been developed to address this problem. Among them, one of the possible solutions is the RFFs method proposed by Rahimi and Recht (2007). We provide an introduction to the RFFs method in the next section, before we thoroughly investigate its theoretical properties in Chapter 3 and 4.

2.3.1 Bochner's Theorem and RFFs Approximation

RFFs is a widely used, simple, and effective technique for scaling up kernel methods. The underlying principle of the approach is a consequence of the classical Bochner's theorem (Bochner, 1932), which states that any bounded, continuous and shift-invariant kernel is the Fourier transform of a bounded positive measure. This measure can be transformed/normalized into a probability measure which is typically called the spectral measure of the kernel. Assuming the spectral measure

$d\tau$ has a density function $p(\cdot)$, the corresponding shift-invariant kernel can be written as

$$k(x,y) = \int_{\mathcal{V}} e^{-2\pi i v^T(x-y)} d\tau(v) = \int_{\mathcal{V}} \left(e^{-2\pi i v^T x}\right)\left(e^{-2\pi i v^T y}\right)^* p(v)dv, \tag{2.8}$$

where c^* denotes the complex conjugate of $c \in \mathbb{C}$. Typically, the kernel is real valued and we can ignore the imaginary part in this equation (Rahimi and Recht, 2007).

KERNEL NAME	$k(\Delta)$	$\mu(v)$
GAUSSIAN	$e^{-\frac{\|\Delta\|_2^2}{2}}$	$(2\pi)^{-\frac{D}{2}} e^{\frac{\|v\|_2^2}{2}}$
LAPLACIAN	$e^{-\|\Delta\|_1}$	$\prod_d \frac{1}{\pi(1+v_d^2)}$
CAUCHY	$\prod_d \frac{2}{\pi(1+\Delta_d^2)}$	$e^{-\|v\|_1}$

Table 2.1: Fourier Transform of Typical Stationary Kernels.

The principle can be further generalized into the following format

$$k(x,y) = \int_{\mathcal{V}} z(v,x)z(v,y)d\tau(v) = \int_{\mathcal{V}} z(v,x)z(v,y)p(v)dv, \tag{2.9}$$

where $z\colon \mathcal{V} \times \mathcal{X} \to \mathbb{R}$ is a continuous and bounded function with respect to v and x. The main idea behind the RFFs is to approximate the kernel function by its Monte-Carlo estimate

$$\tilde{k}(x,y) = \frac{1}{s}\sum_{i=1}^{s} z(v_i,x)z(v_i,y), \tag{2.10}$$

with the reproducing kernel Hilbert space $\tilde{\mathcal{H}}$ (note that in general $\tilde{\mathcal{H}} \nsubseteq \mathcal{H}$) and $\{v_i\}_{i=1}^s$ independently sampled from the spectral measure.

Based on Eq. (2.10), Rahimi and Recht (2007) propose an algorithm to approximate the kernel function with RFFs listed in Algorithm 1.

Algorithm 1 THE CONSTRUCTION OF RFFS FOR $k(x, y)$

Input: A positive definite shift-invariant kernel k
Output: The RFFs approximation of the kernel $\tilde{k}$
1: Compute the spectral measure τ of the kernel k

$$\tau(v) = \frac{1}{2\pi} \int e^{iv^T \Delta} k(\Delta) d\Delta, \quad \Delta = x - y$$

2: Randomly draw s iid samples $v_1, \cdots, v_s$ from τ
3: Let $\mathbf{z}_x(\mathbf{v}) = \sqrt{\frac{1}{s}}[z(v_1, x), \cdots, z(v_s, x)]^T$
4: $\tilde{k}(x, y) = \mathbf{z}_x(\mathbf{v})^T \mathbf{z}_y(\mathbf{v})$

Apart from approximating kernel function k, RFFs is shown to be related to any function f in the RKHS. In Bach (2017a, Appendix A), we know that a function $f \in \mathcal{H}$ can be expressed as [3]

$$f(x) = \int_{\mathcal{V}} g(v) z(v, x) p(v) dv \qquad (\forall x \in \mathcal{X}), \tag{2.11}$$

where $g \in L_2(d\tau)$ is a real-valued function such that $\|g\|^2_{L_2(d\tau)} < \infty$ and $\|f\|_{\mathcal{H}} = \min_g \|g\|_{L_2(d\tau)}$, where the minimum is taken over all possible decompositions of f. Thus, one can take an independent sample $\{v_i\}_{i=1}^s \sim p(v)$ (we refer to this sampling scheme as *plain RFF*) and approximate a function $f \in \mathcal{H}$ at a point $x_j \in \mathcal{X}$ by

$$\tilde{f}(x_j) = \sum_{i=1}^{s} \alpha_i z(v_i, x_j) \coloneqq \mathbf{z}_{x_j}(\mathbf{v})^T \alpha, \quad \text{with } \alpha \in \mathbb{R}^s. \tag{2.12}$$

In standard estimation problems, it is typically the case that for a given set of instances $\{x_i\}_{i=1}^n$ one approximates $\mathbf{f}_x = [f(x_1), \cdots, f(x_n)]^T$ by

$$\tilde{\mathbf{f}}_x = [\mathbf{z}_{x_1}(\mathbf{v})^T \alpha, \cdots, \mathbf{z}_{x_n}(\mathbf{v})^T \alpha]^T \coloneqq \mathbf{Z}\alpha, \tag{2.13}$$

where $\mathbf{Z} = [\mathbf{z}_{x_1}(\mathbf{v}), \ldots, \mathbf{z}_{x_n}(\mathbf{v})]^T \in \mathbb{R}^{n \times s}$.

As the latter approximation is simply a Monte-Carlo estimate, one could also

[3] It is not necessarily true that for any $g \in L_2(d\tau)$, there exists a corresponding $f \in \mathcal{H}$.

pick an importance weighted probability density function $q(\cdot)$ and sample features $\{v_i\}_{i=1}^s$ from q (we refer to this sampling scheme as *weighted RFF*). Then, the function value $f(x_j)$ can be approximated by

$$\tilde{f}_q(x_j) = \sum_{i=1}^{s} \beta_i z_q(v_i, x_j) \coloneqq \mathbf{z}_{q,x_j}(\mathbf{v})^T \beta,$$

with $z_q(v_i, x_j) = \sqrt{\frac{p(v_i)}{q(v_i)}} z(v_i, x_j)$ and $\mathbf{z}_{q,x_j}(\mathbf{v}) = [z_q(v_1, x_j), \cdots, z_q(v_s, x_j)]^T$. Hence, a Monte-Carlo estimate of $\mathbf{f}_x$ can be written in a matrix form as $\tilde{\mathbf{f}}_{q,x} = \mathbf{Z}_q \beta$, where $\mathbf{Z}_q \in \mathbb{R}^{n \times s}$ and $\mathbf{z}_{q,x_j}(\mathbf{v})^T$ is its jth row.

Let $\tilde{\mathbf{K}}$ and $\tilde{\mathbf{K}}_q$ be Gram-matrices with entries $\tilde{\mathbf{K}}_{ij} = \tilde{k}(x_i, x_j)$ and $\tilde{\mathbf{K}}_{q,ij} = \tilde{k}_q(x_i, x_j)$ such that

$$\tilde{\mathbf{K}} = \frac{1}{s} \mathbf{Z}\mathbf{Z}^T, \quad \tilde{\mathbf{K}}_q = \frac{1}{s} \mathbf{Z}_q \mathbf{Z}_q^T.$$

If we now denote the jth column of $\mathbf{Z}$ by $\mathbf{z}_{v_j}(\mathbf{x})$ and the jth column of $\mathbf{Z}_q$ by $\mathbf{z}_{q,v_j}(\mathbf{x})$, then the following equalities can be derived easily from Eq. (2.10)

$$\mathbb{E}_{v\sim p}(\tilde{\mathbf{K}}) = \mathbf{K} = \mathbb{E}_{v\sim q}(\tilde{\mathbf{K}}_q), \quad \mathbb{E}_{v\sim p}\left[\mathbf{z}_v(\mathbf{x})\mathbf{z}_v(\mathbf{x})^T\right] = \mathbf{K} = \mathbb{E}_{v\sim q}\left[\mathbf{z}_{q,v}(\mathbf{x})\mathbf{z}_{q,v}(\mathbf{x})^T\right].$$

Sampling features from the importance weighted probability density function $q(\cdot)$ has led to much interest in the literature (Alaoui and Mahoney, 2015; Avron et al., 2017; Bach, 2017b; Rudi and Rosasco, 2017) as it often leads to computation savings provided that we have access to q. In particular, an importance weighted density function based on the notion of *ridge leverage scores* is introduced in Alaoui and Mahoney (2015) for landmark selection in the Nyström method (Nyström, 1930; Smola and Schölkopf, 2000; Williams and Seeger, 2001b). For landmarks selected using that sampling strategy, Alaoui and Mahoney (2015) establish a sharp convergence rate of the low-rank estimator based on the Nyström method. This

result motivates the pursuit of a similar notion for RFFs. Indeed, Bach (2017b) propose a leverage score function based on an integral operator defined using the kernel function and the marginal distribution of a data-generating process. Building on this work, Avron et al. (2017) propose the ridge leverage function with respect to a fixed input dataset, i.e.,

$$l_\lambda(v) = p(v)\mathbf{z}_v(\mathbf{x})^T(\mathbf{K} + n\lambda\mathbf{I})^{-1}\mathbf{z}_v(\mathbf{x}). \tag{2.14}$$

From our assumption on the decomposition of a kernel function, it follows that there exists a constant z_0 such that $|z(v,x)| \leq z_0$ (for all v and x) and $\mathbf{z}_v(\mathbf{x})^T\mathbf{z}_v(\mathbf{x}) \leq nz_0^2$. We can now deduce the following inequality using a result from Avron et al. (2017, Proposition 4)

$$l_\lambda(v) \leq p(v)\frac{z_0^2}{\lambda} \qquad \text{with} \qquad \int_{\mathcal{V}} l_\lambda(v)dv = \mathrm{Tr}\big[\mathbf{K}(\mathbf{K} + n\lambda\mathbf{I})^{-1}\big] := d_{\mathbf{K}}^{\lambda}.$$

The quantity $d_{\mathbf{K}}^{\lambda}$ is known for implicitly determining the number of independent parameters in a learning problem and is thus called the *effective dimension of the problem* (Caponnetto and De Vito, 2007) or the *number of effective degrees of freedom* (Bach, 2013; Hastie, 2017).

We can now observe that $q^*(v) = l_\lambda(v)/d_{\mathbf{K}}^{\lambda}$ is a probability density function. From now on, we refer to $q^*(v)$ as the *empirical ridge leverage score distribution* and this sampling strategy as *leverage weighted RFF* in the remainder of the thesis. Avron et al. (2017) establish that sampling according to $q^*(v)$ requires fewer Fourier features compared to the standard spectral measure sampling. The reason for this is that when sampling according to $p(v)$, the samples obtained are often clustered around the leading/top eigenvalues of the corresponding kernel matrix $\mathbf{K}$. In contrast, sampling according to a re-weighted distribution is likely to obtain the feature vectors that span the whole eigenspectrum of $\mathbf{K}$.

We remark that the leverage weighted RFF is an example of data dependent sampling scheme for random features. There are many other data dependent sampling strategies in literature (see, e.g., Agrawal et al., 2019; Liu et al., 2020; Shahrampour et al., 2018) which we review below.

Surrogate Leverage Score To avoid inverting the kernel Gram-matrix in Eq. (2.14), Liu et al. (2020) propose a simple surrogate leverage function

$$l_\lambda(v) = p(v)\mathbf{Z}\left(\frac{1}{n^2\lambda}\left(YY^T + nI\right)\right)\mathbf{Z},$$

where $Y = [y_1, \ldots, y_n]^T$. The sampling distribution is then proportional to $l_\lambda(v)$. The surrogate leverage score sampling is empirically verified to be effective in reducing the computation time in many applications such as KRR (Liu et al., 2020) and Canonical Correlation Analysis (Wang and Shahrampour, 2019).

Kernel Polarization The idea of kernel polarization RFFs (KP-RFFs) is to associate each random feature a score. Specifically, it first generates a large number of random features and computes the energy-based score for each random feature as following

$$\tilde{S}(v) = \frac{1}{n}\sum_{i=1}^{n} y_i z(x_i, v).$$

We then select the random feature v with high $\tilde{S}(v)$ value. Ideally, we would like to select v so that the feature vector $\mathbf{z}_v(\mathbf{x}) = [z(v, x_1), \ldots, z(v, x_n)]^T$ aligns well with the label Y. Shahrampour et al. (2018) demonstrate the efficiency of KP-RFFs through empirical experiments.

Kernel Alignment RFFs For Kernel Alignment RFFs (KA-RFFs) (Sinha and Duchi, 2016), it generates a large number of random features (denote as set J), and selects a subset of the features from J by solving the following optimization

problem

$$\max_{\mathbf{a}\in\mathcal{Q}_J} \sum_{i,j=1}^{n} y_i y_j \sum_{t=1}^{J} a_t z(x_i, v_t) z(x_j, v_t),$$

where $\mathbf{a}$ is a weight vector. Here $\mathcal{Q}_J$ is the set of distributions such that

$$\mathcal{Q}_J := \{\mathbf{a} : D_f(\mathbf{a}||1/J) \leq c\},$$

where $D_f(P||Q)$ is the χ^2-divergence and $c > 0$ is some pre-specified constant. The idea of KA-RFFs is to generate the random features so that the kernel matrix is close to YY^T. For more details, please refer to Sinha and Duchi (2016).

2.3.2 Fast Construction of RFFs

Suppose we sample s number of d-dimensional random features to form the transformation matrix $\mathbf{V} = [v_1, \ldots, v_s]^T \in \mathbb{R}^{s\times d}$. The random feature vector for a training data x would be $\varphi(\mathbf{V}x)$ in many applications, where $\varphi(\cdot)$ is some non-linear function such as $\cos$ and $\sin$ functions. Hence, direct computation of $\mathbf{V}x$ would cost $O(sd)$ for both computation and memory cost. When the dimension d is high, computing the feature vector can be costly. Therefore, how to speed up the construction of $\mathbf{V}x$ has attracted much research interest in the machine learning community (see, e.g., Bojarski et al., 2017; Choromanski et al., 2017; Choromanski and Sindhwani, 2016; Feng et al., 2015; Le et al., 2013; Li, 2017; Lyu, 2017; Yu et al., 2016). We provide a review on these fast construction methods below and list the computation and memory cost in Table 2.2.

Fastfood When Gaussian kernel is used for RFFs, the transformation matrix $\mathbf{V}$ is a dense Gaussian matrix. In this case, Le et al. (2013) propose the Fastfood algorithm to fast construct the dense Gaussian matrix using the Hadamard and

diagonal matrices. Specifically, the transformation matrix can be written as

$$\mathbf{V}_{Fastfood} = \theta \mathbf{B}_1 \mathbf{H} \mathbf{G} \mathbf{\Gamma} \mathbf{H} \mathbf{B}_2,$$

where θ is the kernel bandwidth, $\mathbf{H}$ is the Hadamard matrix which allows fast matrix multiplication, and $\mathbf{\Gamma}$ is a $d \times d$ permutation matrix that can decorrelate the two Hadamard matrices. $\mathbf{B}_1, \mathbf{B}_2$ and $\mathbf{G}$ are three diagonal matrices constructed in the following way: $\mathbf{G}_{ii}$ is an i.i.d sample from standard Gaussian distribution; $\mathbf{B}_{1,ii} = \|v_i\|_2 / \|\mathbf{G}\|_F$, where v_i is a sample from the kernel spectral distribution $p(v)$ from Eq. (2.9); $\mathbf{B}_{2,ii}$ is an i.i.d Radamacher random variable, i.e., $P(\mathbf{B}_{2,ii} = \{\pm 1\}) = 1/2$. The Fastfood algorithm utilizes the property of the Hadamard matrix to speed up the matrix multiplication by reducing the computation and the memory cost.

Signed Circulant Random Feature (SCRF) Feng et al. (2015) propose the SCRF algorithm using the circulant matrices. Given s, they let $t = \lceil s/d \rceil$ and construct the transformation matrix as

$$\mathbf{V}_{SCRF} = [\boldsymbol{\nu} \otimes \mathcal{C}(v_1), \ldots, \boldsymbol{\nu} \otimes \mathcal{C}(v_t)]^T \in \mathbb{R}^{td \times d},$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is a Rademacher random vector, $\otimes$ is the tensor product and $\mathcal{C}(v_i)$ is a circulant matrix generated by the random feature v_i sampled from $p(v)$ in Eq. (2.9). Feng et al. (2015) show that the special structure of the circulant matrix allows one to use $O(s)$, instead of $O(sd)$, memory to store the transformation matrix.

$\mathcal{P}$-Model Choromanski and Sindhwani (2016) propose the $\mathcal{P}$-Model, which includes the Fastfood and SCRF as special cases. Specifically, given a constant t,

it constructs the transformation matrix as following

$$\mathbf{V}_{\mathcal{P}} = [\mathbf{g}^T\mathbf{P}_1, \ldots, \mathbf{g}^T\mathbf{P}_s]^T \in \mathbb{R}^{s\times d},$$

where $\mathbf{g} \in \mathbb{R}^t$ is a standard Gaussian random vector and each $\mathbf{P}_i \in \mathbb{R}^{t\times d}$ is a matrix where each of its column has unit l_2 norm. The $\mathcal{P}$-model is designed so that the Gaussian random vector $\mathbf{g}$ can be effectively recycled in order to reduce computation and memory cost.

Structured Orthogonal Random Feature (SORF) Yu et al. (2016) and Bojarski et al. (2017) propose to use a class of structured matrices that are similar to those matrices in Fastfood to speed up the construction. The transformation matrix is given by

$$\mathbf{V}_{SORF} = \sqrt{d}\theta\mathbf{H}\mathbf{D}_1\mathbf{H}\mathbf{D}_2\mathbf{H}\mathbf{D}_3,$$

where θ is the kernel bandwidth, $\mathbf{H}$ is the normalized Hadamard matrix and each $\mathbf{D}_i \in \mathbb{R}^{d\times d}$ is a diagonal matrix where its diagonal entry is generated according to the Radamacher distribution. Later, Choromanski et al. (2017) generalize this idea and propose the random orthogonal emeddings (ROM) algorithm. Given a constant t, the transformation matrix is constructed as

$$\mathbf{V}_{ROM} = \sqrt{d}\theta\prod_{i=1}^{t}\mathbf{H}\mathbf{D}_i,$$

where in ROM, $\mathbf{H}$ can either be the normalized Hadamard matrix or the Walsh matrix. Empirical experiments (Bojarski et al., 2017; Choromanski et al., 2017; Yu et al., 2016) demonstrate that SORF and ROM not only accelerate the computation of the transformation matrix $\mathbf{V}$, but also have the effect of variance reduction

compared to plain RFFs.

Quasi-Monte Carlo (QMC) We have seen that plain RFFs is a direct Monte Carlo sampling method which has the undesired feature clustering effect and hence, has slow error convergence rate. To alleviate these issues, Yang et al. (2014) propose the QMC method which outputs low discrepancy sequences. Specifically, QMC assumes that the spectral distribution $p(v)$ from Eq. (2.8) admits the factorization with respect to each dimension, i.e., $p(v) = \prod_{i=1}^{d} p_i(v_i)$. It then transforms the integral in Eq. (2.8) to an integral on the unit cube $[0,1]^d$ in the following way

$$k(x,y) = \int_{[0,1]^d} \exp\left(-i(x-y)^T \Phi^{-1}(\mathbf{t})\right) d\mathbf{t},$$

where $\Phi^{-1}(\mathbf{t}) = [\Phi_1^{-1}(\mathbf{t}_1), \ldots, \Phi_d^{-1}(\mathbf{t}_d)]$ with Φ_i being the cumulative distribution function of p_i. The transformation matrix for QMC is constructed by first generating a low discrepancy sequence $\mathbf{t}_1, \ldots, \mathbf{t}_s \in [0,1]^d$ and formulated as below

$$\mathbf{V}_{QMC} = [\Phi^{-1}(\mathbf{t}_1), \ldots, \Phi^{-1}(\mathbf{t}_s)].$$

Gaussian Quadrature (GQ) RFFs can be treated as an approximation of the integral in Eq. (2.8), which connects to a long line of work on numerical quadrature for estimating integrals. In light of this, Dao et al. (2017) propose to utilize GQ to construct features. Specifically, they assume that the kernel function k can factorize with respect to each dimension and the spectral distribution is subgaussian. Eq. (2.8) can now be written as

$$k(x,y) = \prod_{i=1}^{d} \left(\int_{-\infty}^{\infty} \exp\left(-i(x_i - y_i)^T v_i\right) p(v_i) dv_i \right),$$

where x_i, y_i, v_i are the i-th coordinate of x, y and v. GQ then approximates the one dimensional integral using one-dimensional quadrature rule. Dao et al. (2017) utilize the Gaussian quadrature with orthogonal polynomials

$$\int_{-\infty}^{\infty} \exp\left(-i(x_i - y_i)v_i\right) p(v_i) dv_i = \sum_{i=1}^{L} a_i \exp(-i(x_i - y_i)\gamma_i),$$

where each γ_i is a chosen location point with weight a_i and L controls the accuracy level in the sense that a univariate GQ with L points is exact for polynomials up to $2L - 1$ degrees.

In the above, we briefly introduce the current various methods to construct the RFFs approximation. We list the computation and memory cost for each method in Table 2.2 below.

ALGORITHM	COMPUTATION TIME	MEMORY COST
PLAIN RFFS	$O(sd)$	$O(sd)$
FASTFOOD	$O(s \log d)$	$O(s)$
SCRF	$O(s \log d)$	$O(s)$
$\mathcal{P}$-Model	$O(s \log d)$	$O(s)$
SORF	$O(s \log d)$	$O(s)$
QMC	$O(sd)$	$O(sd)$
GQ	$O(d \log d)$	$O(d)$

Table 2.2: The computation and memory cost of fast RFFs construction algorithms.

2.3.3 Kernel Low Rank Approximation

RFFs can be broadly treated as a kernel low rank approximation technique. There are a large number of methods aiming to reduce the computation cost of kernel methods. Hence, in this section, we provide a brief review of such methods.

2.3.3.1 Nyström Method

Among the kernel low rank approximation methods, there is a class of subsampling methods, that we broadly refer to as the Nyström methods (Smola and Schölkopf, 2000; Williams and Seeger, 2001a). These methods are closely related to RFFs. Both of the two methods use subsampling technique to produce a new matrix to replace the larger original kernel Gram-matrix. The difference is that RFFs try to subsample in the frequency domain while the Nyström methods conduct the subsampling from the original data domain.

We now explain how the Nyström methods work using the KRR (see Definition 2.8) as an example. The prediction function from KRR can be written as

$$\hat{f}(x) = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, x), \quad \text{where } \hat{\alpha} = (\mathbf{K} + n\lambda I)^{-1} Y. \tag{2.15}$$

This further implies that the function space that our estimator $\hat{f}$ comes from is

$$\mathcal{H}_n := \{f \in \mathcal{H} \mid f = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot), \alpha \in \mathbb{R}^n\}.$$

We can now easily see that once n becomes large, the storage of $\mathbf{K}$ and computation of $\hat{\alpha}$ is cumbersome. In light of this, Nyström methods propose to find the estimator in the following reduced space

$$\mathcal{H}_m := \{f \in \mathcal{H} \mid f = \sum_{i=1}^{m} \alpha_i k(\tilde{x}_i, \cdot), \alpha \in \mathbb{R}^m\},$$

where $m < n$ and $\{\tilde{x}_1, \ldots, \tilde{x}_m\}$ is the subset of covariates from the training sample $\{x_1, \ldots, x_n\}$ selected according to certain sampling strategy. As a result of this

approximation, the solution of the KRR problem now becomes

$$\hat{f}_m(x) = \sum_{i=1}^{m} \hat{\alpha}_{m,i} k(x_i, x), \quad \text{where } \hat{\alpha}_m = (\mathbf{K}_{nm}^T \mathbf{K}_{nm} + n\lambda \mathbf{K}_{mm})^\dagger \mathbf{K}_{nm}^T Y,$$

where $(\mathbf{K}_{nm})_{ij} = k(x_i, \tilde{x}_j)$, $(\mathbf{K}_{mm})_{ij} = k(\tilde{x}_i, \tilde{x}_j)$, and $A^\dagger$ denotes the pseudoinverse of a matrix A. Since m is chosen to be smaller than n, now the storage and computation can be reduced to $O(nm)$ and $O(m^3)$ (or $O(nm^2)$) respectively.

The efficacy of Nyström methods have been both empirically and theoretically examined. From the theoretical perspective, the study of the Nyström methods starts from investigating the approximation error between the original kernel matrix and its subsampled version (Drineas et al., 2012; Drineas and Mahoney, 2005; Gittens and Mahoney, 2013; Wang and Zhang, 2013, 2014). While these results provide sharp analysis on the matrix approximation error due to subsampling, they do not shed light on the generalization properties of Nyström methods. Consequently, Cortes et al. (2010), Bach (2013), Alaoui and Mahoney (2015), and Rudi et al. (2015) thoroughly investigate the algorithm from a learning risk perspective and prove that the Nyström methods can indeed provide a computational gain for kernel methods. We list the sharpest findings in Table 2.3 at the end of the section.

2.3.3.2 Iterative Methods

Apart from Nyström methods, there is another class of algorithms which aims to reduce the computation of kernel methods, which we call *iterative methods* (Caponnetto and Yao, 2010; Engl et al., 1996; Gerfo et al., 2008; Ong and Canu, 2004; Ong et al., 2004). Instead of directly performing the matrix inverse as in Eq. (2.15), iterative methods seeks to compute the estimating function iteratively, where the computation cost is reduced within each iteration. In this section, we briefly introduce the iterative methods in the context of KRR.

The key of iterative methods relies on viewing the prediction function in Eq. (2.15) as a filter on the eigenvalues of the kernel matrix. Starting from Eq. (2.15), the in-sample prediction of kernel ridge regression can be written as

$$\begin{aligned}\hat{\mathbf{f}} := [\hat{f}(x_1), \ldots, \hat{f}(x_n)]^T &= \mathbf{K}(\mathbf{K} + n\lambda I)^T Y \\ &= \frac{\mathbf{K}}{n}(\frac{\mathbf{K}}{n} + \lambda I)^{-1} Y.\end{aligned}$$

Hence, if u_i is an eigenvector of $\mathbf{K}$ with eigenvalue λ_i, then $\frac{\mathbf{K}}{n}(\frac{\mathbf{K}}{n} + \lambda I)^{-1} u = \frac{\lambda_i}{\lambda_i + \lambda} u$. As a result, applying the regularization in regression amounts to employing a filter on the eigenvalues of the kernel matrix. Rosasco et al. (2005) demonstrate that the filter $\frac{\lambda_i}{\lambda_i + \lambda}$ guarantees a strong generalization properties and provides the necessary numerical stability. In light of this, a new class of algorithms can be defined by adopting the following view of Eq. (2.15)

$$\hat{f}(x) = \sum_{i=1}^{n} \hat{\alpha}_i k(x_i, x), \quad \text{where } \hat{\alpha} = \frac{1}{n} g_\lambda \left(\frac{\mathbf{K}}{n}\right) Y,$$

where $g_\lambda : [0, \infty) \to \mathbb{R}$ is some properly defined function and $g_\lambda \left(\frac{\mathbf{K}}{n}\right)$ is defined on the eigenvalues of $\mathbf{K}$. Depending on the specific g_λ function used, there are many different iterative algorithms. Below, we briefly discuss two popular methods, please refer to Gerfo et al. (2008) for more detailed iterative algorithms.

Iterative Landweber: The iterative Landweber uses the following filter function

$$g_t(\lambda_j) = \tau \sum_{i=0}^{t-1} (1 - \tau\lambda_j)^i,$$

where t is a positive integer. In each iteration, the computation of the $\hat{\alpha}$ is through the following update step

$$\hat{\alpha}^{(k)} = \hat{\alpha}^{(k-1)} + \frac{\tau}{n}(Y - \mathbf{K}\hat{\alpha}^{(k-1)}) \quad k = 1, \ldots, t.$$

The iterative Landweber has been shown to be connected to the gradient method where size of t controls the regularization (Gerfo et al., 2008; Yao et al., 2007).

Iterated Tikhonov: Iterated Tikhonov employs the following filter function

$$g_\lambda(\lambda_i) = \frac{(\lambda + \lambda_i)^v - \lambda^v}{\lambda(\lambda + \lambda_i)^v}, v \in \mathbb{N}.$$

In addition, the algorithm is defined by the following iterative procedure

$$(\mathbf{K} + n\lambda I)\hat{\alpha}^{(k)} = Y + n\lambda\hat{\alpha}^{(k-1)}, \quad k = 1, \ldots, v,$$

where we could simply choose the starting point as $\hat{\alpha}^{(0)} = 0$. Note that if we choose $v = 1$, then this reduces to the Tikhonov regularization used in KRR.

2.3.3.3 Divide and Conquer

The divide and conquer is a data-splitting approach that also aims to address the computation issues for kernel methods. Below we illustrate its idea using KRR as an example

1. Given the training set $\{(x_i, y_i)\}_{i=1}^n$, we divide the dataset evenly and uniformly at random into m subsets $S_1, \ldots, S_m$;
2. For each subset S_i, we perform the KRR estimation and obtain a local estimator $\hat{f}_i$;
3. Average all the local estimators and the final estimation function is $\tilde{f} = \frac{1}{m}\sum_{i=1}^m \hat{f}_i$.

One can see that if $m < n$, then the computational cost can be saved in the KRR setting. The theoretical properties of divide and conquer are rigorously studied in Zhang et al. (2013, 2015). They demonstrate that the divide and conquer can

indeed provide us with computational savings. We list its computational cost in Table 2.3.

2.3.3.4 RFFs with Stochastic Gradient Descent

Stochastic gradient descent (SGD) methods allow one to process data points individually, or in small batches to reduce computational complexity while preserving fast convergence rate (Bottou and Bousquet, 2007). RFFs can reduce the dimension of the feature to overcome the memory bottleneck for a typical kernel learning problem. Therefore, Carratino et al. (2018) and Yashima et al. (2021) design and study the RFFs with SGD algorithm by combining SGD and RFFs together in order to obtain fast computation.

Given training sample $\{(x_i, y_i)\}_{i=1}^n$, RFFs with SGD starts with sampling s frequencies $v_1, \ldots, v_s$ from the spectral distribution and forms the RFFs feature vector $\mathbf{z}_x(\mathbf{v}) = [z(v_1, x), \ldots, z(v_s, x)]^T$. The prediction function is parameterized as $\tilde{f}(x) = \mathbf{z}_x(\mathbf{v})^T\beta$, where β is the regression coefficient. The algorithm then chooses a subset of the training samples to update the gradient. Specifically, it chooses a subset S_t from the training sample at time t, and updates with a fixed learning rate η as following

$$\beta^1 = 0, \quad \beta^{t+1} = \beta^t - \frac{\eta}{|S_t|} \sum_{x_i, y_i \in S_t} \left(\mathbf{z}_{x_i}(\mathbf{v})^T\beta^t - y_i\right) \mathbf{z}_{x_i}(\mathbf{v}), \quad t = 1, \ldots, T.$$

RFFs with SGD is shown to effectively reduce the computational cost while preserving the same statistical prediction accuracy as KRR (Carratino et al., 2018; Yashima et al., 2021). We list the computational cost in Table 2.3.

2.3.3.5 Distributed Learning with RFFs

Recently, a line of research combines the RFFs approximation with the distributed learning to obtain fast approximation methods (see, e.g., Li et al., 2019a; Liu et al.,

2021; Richards et al., 2020). In this section, we briefly introduce these algorithms and discuss their computational gains.

Distributed KRR with Random Features The simplest way to combine the RFFs with distributed learning is the Distributed KRR with Random Features (DKRR-RF) proposed by Li et al. (2019a). The algorithm is similar to the Divide and Conquer method in KRR problems. We list the steps below

1. Given the training set $\{(x_i, y_i)\}_{i=1}^n$, we divide the dataset evenly and uniformly at random into m subsets $S_1, \ldots, S_m$;
2. For each subset S_i, we perform the local RFFs regression estimator $\tilde{f}_i$;
3. Average all the local estimators and the final estimation function is $\tilde{f} = \frac{1}{m}\sum_{i=1}^m \tilde{f}_i$.

The theoretical properties of DKRR-RF are studied in Li et al. (2019a). Through a novel error decompostion method, they demonstrate that the DKRR-RF can provide computational savings without hurting the prediction accuracy. The detailed computation cost is listed in Table 2.3.

DKRR-RF With Communications A key restriction for the DKRR-RF is that the number of subsets of the data m has to remain constant to guarantee no loss of prediction accuracy. Therefore, Liu et al. (2021) propose the DKRR-RF With Communications (DKRR-RF-CM) to enlarge the number of subsets of data. The DKRR-RF-CM algorithm can be summarized as

1. Given the training set $\{(x_i, y_i)\}_{i=1}^n$, we divide the dataset evenly and uniformly at random into m subsets $S_1, \ldots, S_m$;
2. Draw s random features, and formulate the feature matrix for each subset S_i as $\mathbf{Z}_i$;

3. Initialize the global regression coefficient $\beta^0 = 0$;

4. At time t, compute the gradient of the regularized empirical risk

$$g_{S_i,s} = (\mathbf{Z}_i^T \mathbf{Z}_i + \lambda I)^{-1} \beta^{t-1} + \mathbf{Z}_i Y_i,$$

   where Y_i are the response vector from S_i;

5. Average all the gradient $\bar{g}_s = \frac{1}{m} \sum_{i=1}^{m} g_{S_i,s}$;

6. Update the regression coefficient for each subset as

$$\beta_i^t = (\mathbf{Z}_i^T \mathbf{Z}_i + \lambda I)^{-1} \bar{g}_s;$$

7. Update the global regression coefficient

$$\beta^t = \beta^{t-1} - \frac{1}{m} \sum_{i=1}^{m} \beta_i^{t-1};$$

8. Repeat Step $4 - 6$ for p times.

The key idea of DKRR-RF-CM is to adopt a Newton Raphson iteration-based communication strategy (steps $4 - 6$) which allows one to increase the size of the subsets of the data while preserving the same trade-offs between computational costs and prediction accuracy as in DKRR-RF. The theoretical properties of DKRR-RF-CM are studied in Liu et al. (2021). We list its computation savings in Table 2.3.

Decentralized Learning with RFFs and Distributed Gradient Descent While the above two centralized approaches are effective in reducing the computational cost, a key problem is that a single global agent is required to collect and process information from each local agent. For example, in the above two distributed

learning regime, one agent is responsible for collecting all the estimators from each local agent in order to compute the global average. However, in many applications, such a centralized approach is not feasible due to various reasons (e.g., privacy issue). Therefore, Richards et al. (2020) propose a decentralized learning with RFFs and distributed gradient descent (DC-RF-GD).

Consider a connected network of m agents $G = (V, E)$ with $|V| = m$ and edges $E \subseteq V \times V$. We assume that each agent has a collection of l ($n = ml$) i.i.d training points $\{(x_{i,v}, y_{i,v})\}_{i=1}^{l}$. The DC-RF-GD states that each local agent wishes to solve the RFFs regression problem

$$\underset{\beta_v, v \in V}{\arg\min} \frac{1}{n} \sum_{v \in V} \|Y_v - \mathbf{Z}_v \beta_v\|_2^2 + s\lambda \|\beta_v\|_2^2,$$

where $Y_v = [y_{1,v}, \ldots, y_{l,v}]^T$, $\mathbf{Z}_v$ is the RFFs feature matrix for agent v, and s is the number of RFFs.

The DC-RF-GD uses two step to solve the optimization problem. In the first step, any local agent w performs a local gradient descent with stepsize η

$$a_w^t = \beta_w^t - \frac{\eta}{l} \sum_{i=1}^{l} \left(\mathbf{z}_{x_{i,w}}(\mathbf{v})^T \beta_w^t - y_{i,w} \right) \mathbf{z}_{x_{i,w}}(\mathbf{v}).$$

In the second step, an agent v communicates with its neighbors to further update the gradient $\beta_v^{t+1} = \sum_{w \in V} P_{vw} a_w^t$, where $P \in \mathbb{R}^{n \times n}$ is a doubly stochastic matrix such that $P_{i,j} \neq 0$ only if $(i, j) \in E$. Table 2.3 lists the computational cost for the DC-RF-GD.

Besides the algorithms introduced above, there are also many other methods which combine the above algorithms to pursue further computational or memory savings. Examples include Nyström with iterative method (Camoriano et al., 2016; Tu

et al., 2016), Nyström with stochastic gradient descendent (Dieuleveut and Bach, 2016), Nyström with preconditioning (Rudi et al., 2017, FALKON), etc. Below we provide a table to summarize the computational gain for methods we have introduced.

ALGORITHM	TRAINING TIME	TESTING TIME	MEMORY
PLAIN KERNEL METHOD	$O(n^3)$	$O(n^2)$	$O(n^2)$
RFFS	$O(n^2)$	$O(\sqrt{n})$	$O(n\sqrt{n})$
RFFS+ SGD	$O(n\sqrt{n})$	$O(\sqrt{n})$	$O(n)$
ITERATIVE METHODS	$O(n^{2.25})$	$O(n)$	$O(n^2)$
DIVIDE AND CONQUER	$O(n^2)$	$O(\sqrt{n})$	$O(n)$
DKRR-RF	$O(n\sqrt{n})$	$O(\sqrt{n})$	$O(n)$
DKRR-RF-CM	$O(n\sqrt{n})$	$O(\sqrt{n})$	$O(n)$
DC-RF-GD	$O(n)$	$O(\sqrt{n})$	$O(n)$
NYSTRÖM	$O(n^2)$	$O(\sqrt{n})$	$O(n\sqrt{n})$
NYSTRÖM + ITERATIVE	$O(n^2)$	$O(\sqrt{n})$	$O(n)$
NYSTRÖM + SGD	$O(n^2)$	$O(\sqrt{n})$	$O(n)$
FALKON	$O(n\sqrt{n})$	$O(\sqrt{n})$	$O(n)$

Table 2.3: The computation time of fast kernel algorithms.

Chapter 3 Kernel Matrix Approximation with RFFs

In this chapter, we provide our first attempt to develop the theoretical understanding of the RFFs, where we adopt the matrix approximation perspective, and study the behaviour of $\sup |\tilde{k}(x,y) - k(x,y)|$ as a function of the number of features used. In addition, we are interested in how this error propagates to $\|\tilde{\mathbf{K}} - \mathbf{K}\|$ as well as the learning risk in various kernel learning settings. We start with analyzing the kernel matrix approximation error in kernel ridge regression in Section 3.2, followed by providing the error convergence rate in the distribution regression setting in Section 3.3.

3.1 Introduction

Kernel methods are one of the pillars of machine learning (Schölkopf and Smola, 2001; Schölkopf et al., 2004), as they give us a flexible framework to model complex functional relationships in a principled way and also come with well-established statistical properties and theoretical guarantees (Caponnetto and De Vito, 2007; Steinwart and Christmann, 2008). The key ingredient, known as *kernel trick*, allows implicit computation of an inner product between rich feature representations of data through the kernel evaluation $k(x,x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{H}}$, while the actual feature mapping $\varphi : \mathcal{X} \to \mathcal{H}$ between a data domain $\mathcal{X}$ and some high and often infinite dimensional Hilbert space $\mathcal{H}$ is never computed. However, such convenience comes at a price: due to operating on all pairs of observations, kernel methods inherently require computation and storage which is at least quadratic in the number of observations, and hence often prohibitive for large datasets. In particular, the kernel matrix has to be computed, stored, and often inverted. As a

result, a flurry of research into scalable kernel methods and the analysis of their performance emerged (Alaoui and Mahoney, 2015; Bach, 2013; Mahoney and Drineas, 2009; Rahimi and Recht, 2007; Rudi et al., 2015, 2017; Rudi and Rosasco, 2017; Zhang et al., 2015). Among the most popular frameworks for fast approximations to kernel methods are RFFs due to Rahimi and Recht (2007). The idea of RFFs is to construct an explicit feature map which is of a dimension much lower than the number of observations, but with the resulting inner product which approximates the desired kernel function $k(x, y)$. In particular, RFFs rely on Bochner's theorem (Bochner, 1932; Rudin, 2017) which tells us that any bounded, continuous and shift-invariant kernel is the Fourier transform of a bounded positive measure, called spectral measure. The feature map is then constructed using samples drawn from the spectral measure. Essentially, any kernel method can then be adjusted to operate on these explicit feature maps (i.e., primal representations), greatly reducing the computational and storage costs, while in practice mimicking performance of the original kernel method. Despite the empirical success of RFFs, the theoretical properties of RFFs remains unclear. Hence, in this chapter, we provide our first attempt to address this issue from the kernel matrix approximation perspective.

Before we present our analysis, we review some current results on the uniform convergence property of the RFFs. Since this chapter serves as the introductory analysis of the RFFs, we will only provide some basic results from literature and leave the extensive review in Chapter 4.

3.1.1 A Review on the Uniform Convergence Property of RFFs

We have seen that RFFs can generate low-dimensional (of dimension s) feature map z to approximate the often infinite dimensional feature map in kernel methods. After that, we can apply fast liner solvers to efficiently approximate the solution from the original kernel methods. However, we might also want to explore the

properties of the estimator $\tilde{k}$ and the accuracy loss due to the RFFs approximation. Below we present two theorems demonstrating the almost sure (a.s.) convergence of this estimator. The following theorem applies to kernels with spectral measures which have finite second moments.

Theorem 3.1 (Rahimi and Recht (2007)). *Let $\mathbb{S}$ be a compact subset of $\mathcal{X}$ with diameter $|\mathbb{S}|$. $k(x,y) = \psi(x-y)$, $x, y \in \mathbb{S}$ be a continuous bounded positive definite function. Assume that k is such that $\sigma_k^2 := \int \|v\|^2 d\tau(v) < \infty$, where $d\tau$ is the spectral measure defined in Chapter 2.3. Then we have*

$$P\left[\sup_{x,y\in\mathbb{S}} |\tilde{k}(x,y) - k(x,y)| \geq \epsilon\right] \leq 256 \left(\frac{|\mathbb{S}|\sigma_k}{\epsilon}\right)^2 exp\left(-\frac{s\epsilon^2}{8(d+2)}\right) \quad (3.1)$$

Let $A_s := \sup_{x,y\in\mathbb{S}} |\tilde{k}(x,y) - k(x,y)|$. Then, Eq. (3.1) has shown that $\tilde{k}$ is a consistent estimator of k, i.e., with high probability

$$A_s = O\left(|\mathbb{S}|\sqrt{\frac{\log(s)}{s}}\right).$$

However, as pointed out by Sriperumbudur and Szabó (2015), this convergence rate is not optimal. The reason is that the RFF estimator is an empirical approximation of the Fourier integral and can be treated as an *empirical characteristic function* (ECF). The ECF is extensively studied in statistics literature (Csorgo and Totik, 1983; Feuerverger and Mureika, 1977; Giné and Zinn, 1984) and it can be shown that the optimal convergence rate of A_s is $s^{-\frac{1}{2}}$. In addition, the a.s. convergence is guaranteed as long as the diameter of $\mathbb{S}$ is $o(e^s)$. Therefore, Sriperumbudur and Szabó (2015) give an improved result regarding to the convergence rate of RFFs estimator $\tilde{k}(x,y)$, presented below.

Theorem 3.2 (Sriperumbudur and Szabó (2015)). *Let $\mathbb{S}$, k, σ_k be as in Theorem*

3.1. Then

$$P\left[\sup_{x,y\in\mathbb{S}}|\tilde{k}(x,y)-k(x,y)|\geq\epsilon\right]\leq\exp\left(-\frac{s(\epsilon-\frac{h(d,|\mathbb{S}|,\sigma_k)}{\sqrt{s}})^2}{2}\right)$$

where we have

$$h(d,|\mathbb{S}|,\sigma_k):=32\sqrt{2d\log(|\mathbb{S}|+1)}+32\sqrt{2d\log(\sigma_k+1)}$$
$$+16\sqrt{2d(\log(|\mathbb{S}|+1))^{-1}}.$$

Theorem 3.2 clearly shows that $\tilde{k}$ is a consistent estimator of k with the a.s. convergence at the rate $\sqrt{s^{-1}\log|\mathbb{S}|}$. Comparing to Theorem 3.1, it provides an improved convergence rate. In addition, Theorem 3.2 shows that $\tilde{k}$ is consistent as long as $\sqrt{s^{-1}\log|\mathbb{S}|}\to 0$ as $s\to\infty$. It can be shown that the findings match the result from Csorgo and Totik (1983); Feuerverger and Mureika (1977); Giné and Zinn (1984).

3.2 Kernel Matrix Approximation in Kernel Ridge Regression

In this section, we present our contribution to the error analysis of kernel approximation from a Gram-matrix approximation perspective. To accomplish this, we will first proceed by revisiting and improving a result due to Sutherland and Schneider (2015). After that, we obtain the error convergence rate for the kernel ridge regression setting with RFFs approximation.

3.2.1 The Expected Max Error of RFFs

Define $\|k\|_\infty = \sup_{x,y\in\mathbb{S}} |\tilde{k}(x,y) - k(x,y)|$. The tail behaviour of $\|k\|_\infty$ has already been studied in Theorem 3.1 and 3.2. Here, we are interested in investigating the expected worst case error, i.e., $\mathbb{E}\|k\|_\infty$. The reason we study this is that in many machine learning problems, our goal is to minimize the learning risk, which is the expected value of loss functions plus the expected error due to RFFs kernel approximation. In the following, we first provide an initial derivation of the expected max error due to Sutherland and Schneider (2015). We then present our analysis which improves their result.

Theorem 3.3. *(Sutherland and Schneider, 2015) Let $\mathbb{S}$, k be as in Theorem 3.1. Define $\mathcal{X}_\Delta := \{x - y | x, y \in \mathcal{X}\}$, suppose k is L-Lipschitz on $\mathcal{X}_\Delta$. Let $R := \mathbb{E}\max_{i=1,\ldots,s} \|v_i\|$, $\gamma \approx 0.964$, is a constant. Then*

$$\mathbb{E}\left[\|k\|_\infty\right] \leq \frac{24\gamma\sqrt{d}|\mathbb{S}|}{\sqrt{s}}(R + L) \tag{3.2}$$

Theorem 3.3 provides a first analysis of the expected error in RFF ridge regression, which shows that the expected max error decays at the rate up to $O(s^{-1/2})$. Sutherland and Schneider (2015) comment that the direct integral of the expected max error diverges, hence, they employ a more sophisticated tool to obtain the upper bound. However, the upper bound is problematic due to the R term which is difficult to compute and it is also unclear how R grows with s. Thus the bound in Eq. (3.2) is possibly vacuous for a large number of kernel functions. Also, the R term varies with different kernels. On the other hand, our contribution in the following theorem is a more direct way to obtain the upper bound and is consistent across many different kernels. The proof is in Section 3.4.

Theorem 3.4. *Let $\mathbb{S}$, k, σ_k be as in Theorem 3.1. Assuming $s > [8e(d+2)]$, then*

we have

$$\mathbb{E}\left[\|k\|_\infty\right] \leq \left[1 + 128\sqrt{2\pi}\sigma_k^2 \frac{|\mathbb{S}|^2}{\sqrt{\log(s)}}\right]\sqrt{8(d+2)}\sqrt{\frac{\log(s)}{s}}$$

Theorem 3.4 shows that the expected max error converges to 0 almost surely with convergence rate $\sqrt{\frac{\log(s)}{s}}$. However, as discussed in Sriperumbudur and Szabó (2015), Theorem 3.1 does not provide the optimal convergence rate. Instead of using Theorem 3.1, we could easily obtain a sharper expected max error using Theorem 3.2. The proof of the next result is in Section 3.4.

Theorem 3.5. *Let $\mathbb{S}$, k, σ_k be as in Theorem 3.1. Let $h(d, |\mathbb{S}|, \sigma_k)$ be as in Theorem 3.2. If we assume that $\sqrt{\frac{\log|\mathbb{S}|}{s}}$ is upper bounded, i.e., $|\mathbb{S}| = o(\exp(s))$. we have*

$$\mathbb{E}(\|k\|_\infty) \leq \sqrt{2\pi}s^{-\frac{1}{2}}.$$

Theorem 3.5 shows a sharper convergence rate for the expected max error. Moreover, it also improves the dependence on $|\mathbb{S}|$. Suppose the diameter of $\mathbb{S}$ can grow with s ($|\mathbb{S}_s| \to \infty$ as $s \to \infty$), then we can see Theorem 3.5 states that the expected max error converge to 0 even when $|\mathbb{S}_s|$ grows exponentially with s, comparing with $[\log(s)]^{\frac{1}{4}}$ in Theorem 3.4.

3.2.2 RFFs in Kernel Ridge Regression

We are now ready to present our analysis on the convergence rate of the learning risk in kernel ridge regression with RFFs. The proof of the next theorem can be found in Section 3.4.

Theorem 3.6. *Suppose we have training set $\{(x_i, y_i)\}_{i=1}^n$, with $x_i \in \mathcal{X}$, $y_i \in \mathbb{R}$. Let $\hat{f}^\lambda(x)$ be the kernel ridge regression solution. Recall we denote $\mathbf{K}$ as the Gram-matrix and $K(x, X) = [k(x, x_1), \ldots, k(x, x_n)]$, let $\tilde{f}(x)$ be the RFFs*

approximation of $\hat{f}^{\lambda}(x)$*, where* $\mathbf{K}$ *and* $K(x, X)$ *are approximated by* $\tilde{\mathbf{K}}$ *and* $\tilde{K}(x, X)$ *respectively. Moreover, we assume that there exists* $\kappa < \infty$ *such that* $k(x, x) \leq \kappa, \tilde{k}(x, x) \leq \kappa \ \forall x \in \mathcal{X}$*. Suppose there also exists* $M < \infty$*, such that* $\|\mathbf{y}\|_2 \leq M$*. Let* $\lambda = n\lambda_0$*, then we have*

$$\begin{aligned}\mathbb{E}(l_{\tilde{f}}) &\leq \mathbb{E}(l_{\hat{f}^{\lambda}}) + \mathbb{E}(|\hat{f}^{\lambda}(x) - \tilde{f}(x)|) \\ &\leq \mathbb{E}(l_{\hat{f}^{\lambda}}) + \sqrt{2\pi} M \frac{\lambda_0 + \kappa}{\lambda_0^2} s^{-\frac{1}{2}}\end{aligned}$$

Theorem 3.6 provides an initial analysis of the learning risk for RFFs estimator $\tilde{f}(x)$ and demonstrates that it is a consistent estimator with excess risk converging at $O(s^{-1/2}) + O(n^{-1/2})$. [1] However, this worst case analysis gives a pessimistic bound as it requires $s = n$ for the RFFs estimator to have the same learning rate as the original KRR estimator. This means that we cannot enjoy any computation gain with RFFs. In the next chapter, we will try to address this problem from a different perspective.

3.3 Kernel Matrix Approximation in Distribution Regression

In this section, we analyze the RFFs approximation problem in distribution regression. Section 3.3.1 explains how the RFFs is employed in approximating the solution of distribution regression, Section 3.3.2 presents an analysis of the RFFs approximation error.

3.3.1 RFFs in Distribution Regression

Let $k(x, y) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a bounded, continuous, positive definite and translation invariant kernel. Then we know that its Fourier transform is a probability

[1] Note that the $O(n^{-1/2})$ term is for the excess risk convergence rate of kernel ridge regression estimator $\hat{f}^{\lambda}$ in the minimax sense, please refer Caponnetto and De Vito (2007) for details.

measure and hence the kernel can be approximated by $\tilde{k}(x, y) = \mathbf{z}_x(\mathbf{v})^T \mathbf{z}_y(\mathbf{v})$ through Algorithm 1. We would like to extend the approximation to the kernel mean embedding in the distribution regression setting. As the definition of the mean embedding μ_P for a probability distribution P depends on the probability measure and the associated kernel, we have various ways of approximating a mean embedding. Specifically, let $\tilde{k}(\cdot, x) := \mathbf{z}_x(\mathbf{v})$, we denote

$$\mu_P^{\tilde{k}} = \int_{\mathcal{X}} \tilde{k}(\cdot, \mathbf{x}) dP(x) = \int_{\mathcal{X}} \mathbf{z}_x(\mathbf{v}) dP(\mathbf{x});$$
$$\mu_{\hat{P}}^{k} = \frac{1}{n} \sum_{i=1}^{n} k(\cdot, \mathbf{x}_i);$$
$$\mu_{\hat{P}}^{\tilde{k}} = \frac{1}{n} \sum_{i=1}^{n} \tilde{k}(\cdot, \mathbf{x}_i) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_x(\mathbf{v}).$$

Similar to kernel ridge regression, the distribution regression analysis is computationally prohibitive as both prediction and storage cost grow at least quadratically with n. As a result, the two layer random features have been applied to reduce computation cost. For example, in Jitkrittum et al. (2015), the two layer RFFs is employed to speed up the learning process of message operator in expectation propagation. The FastFood random feature representation of kernel embeddings is used to solve the Gaussian process distribution regression problem while learning the individual-level associations from aggregate data in the context of ecological inference (Flaxman et al., 2015). In approximating kernel based distances between probability densities, RFFs is exploited to convert problem into primal space, avoiding the computation of the Gram-matrix (Sutherland et al., 2016). Finally, distribution regression with RFFs has also been used for summary statistic selection in Approximate Bayesian Computation to approximate the true likelihood in complex generative model (Mitrovic et al., 2016).

In order to reduce the computational cost, we utilize the two layer RFFs to approximate the two different kernels in MERR: the data kernel k and the bag kernel K.

The data kernel k is used to calculate the mean embedding μ_P^k for distribution P. Then with the new batch of dataset $\{(\mu_{P_i}^k, y_i)\}_{i=1}^l$, we perform the kernel ridge regression with kernel K. The goal of using RFFs is to approximate the kernel between two mean embeddings $K(\mu_{P_i}^k, \mu_{P_j}^k)$. As discussed previously, the RFFs can be utilized to approximate a translation invariant kernel with random features. Here, we will follow a similar approach to derive the two stage random features to approximate $K(\mu_{P_i}^k, \mu_{P_j}^k)$. We first assume that the bag kernel K is a radial kernel (Steinwart and Christmann, 2008): $K(\mu_{P_i}^k, \mu_{P_j}^k) = K(\|\mu_{P_i}^k - \mu_{P_j}^k\|_{\mathcal{H}}^2)$. By expanding the kernel, we get

$$\begin{aligned} K(\mu_{P_i}^k, \mu_{P_j}^k) &= K(\|\mu_{P_i}^k - \mu_{P_j}^k\|_{\mathcal{H}}^2) \\ &= K(\langle \mu_{P_i}^k, \mu_{P_i}^k \rangle_{\mathcal{H}} - 2\langle \mu_{P_i}^k, \mu_{P_j}^k \rangle_{\mathcal{H}} + \langle \mu_{P_j}^k, \mu_{P_i}^k \rangle_{\mathcal{H}}) \end{aligned} \tag{3.3}$$

Now we assume that the data kernel k is translation invariant. According to Rahimi and Recht (2007), there is a vector of random features $\mathbf{z}_x(\mathbf{v})$ such that $k(x_i, x_j) \approx \mathbf{z}_{x_i}(\mathbf{v})^T \mathbf{z}_{x_j}(\mathbf{v})$ where $x_i, x_j \in \mathcal{X}$ and $\mathbf{z}_x(\mathbf{v}) \in \mathbb{R}^{s_k}$. Therefore, we can calculate the inner product between mean embeddings in the following way

$$\begin{aligned} \langle \mu_{P_i}^k, \mu_{P_j}^k \rangle_{\mathcal{H}} &= \langle \mathbb{E}_{P_i}[k(\cdot, x_i)], \mathbb{E}_{P_j}[k(\cdot, x_j)] \rangle_{\mathcal{H}} \\ &= \mathbb{E}_{P_i} \mathbb{E}_{P_j} [\langle k(\cdot, x_i), k(\cdot, x_j) \rangle_{\mathcal{H}}] \\ &= \mathbb{E}_{P_i} \mathbb{E}_{P_j} [k(x_i, x_j)] \\ &= \mathbb{E}_{P_i} \mathbb{E}_{P_j} [k(x_i - x_j)] \\ &\approx \mathbb{E}_{P_i} \mathbb{E}_{P_j} [\mathbf{z}_{x_i}(\mathbf{v})^T \mathbf{z}_{x_j}(\mathbf{v})] \\ &= \mathbb{E}_{P_i} (\mathbf{z}_{x_i}(\mathbf{v}))^T \mathbb{E}_{P_j} (\mathbf{z}_{x_j}(\mathbf{v})) \\ &:= \tilde{\mu_{P_i}^k}^T \tilde{\mu_{P_j}^k} \end{aligned} \tag{3.4}$$

Here the mapping $\tilde{\mu_{P_i}^k}$ is the mean embedding of the RFFs vector $\mathbf{z}_x(\mathbf{v})$. However, since the underlying distribution P_i, P_j are not observable, we will use the empirical

Algorithm 2 CONSTRUCTION OF THE TWO-STAGE RANDOM FOURIER FEATURES FOR $K(\mu^k_{P_i}, \mu^k_{P_j})$

Input: Data kernel k, Bag kernel K, Number of data features s_k, Number of bag features s_K, $\{x_{i1}, \cdots x_{iN}\}$ and $\{x_{j1}, \cdots x_{jN}\}$ from distribution P_i, P_j respectively.
Output: Random features $Z_{\mu^{\tilde{k}}_{\hat{P}_i}}(\mathbf{w}), Z_{\mu^{\tilde{k}}_{\hat{P}_j}}(\mathbf{w}) \in R^{s_K}$.

1: Compute the Fourier transform Q_k and Q_K of kernel k and K
2: Sample $\{v_i\}_{i=1}^{s_k} \overset{\text{i.i.d}}{\sim} Q_k$ and $\{w_i\}_{i=1}^{s_K} \overset{\text{i.i.d}}{\sim} Q_K$
3: $\mu^{\tilde{k}}_{\hat{P}_i} = \sqrt{\frac{1}{s_k}}[\mathbb{E}_{\hat{P}_i} z(v_1, x), \ldots, \mathbb{E}_{\hat{P}_i} z(v_{s_k}, x)]$
4: $Z_{\mu^{\tilde{k}}_{\hat{P}_i}}(\mathbf{w}) = \sqrt{\frac{1}{s_K}}[Z(w_1, \mu^{\tilde{k}}_{\hat{P}_i}), \ldots, Z(w_{s_K}, \mu^{\tilde{k}}_{\hat{P}_i})]$
5: $\tilde{K}(\mu^{\tilde{k}}_{\hat{P}_i}, \mu^{\tilde{k}}_{\hat{P}_j}) = Z_{\mu^{\tilde{k}}_{\hat{P}_i}}(\mathbf{w})^T Z_{\mu^{\tilde{k}}_{\hat{P}_j}}(\mathbf{w})$.

distribution to approximate the expectation, i.e., we approximate $\mu^{\tilde{k}}_{P_i}$ with $\mu^{\tilde{k}}_{\hat{P}_i}$. With the RFFs mean embedding defined as in Eq. (3.4), the kernel between mean embeddings in Eq. (3.3) is now

$$\begin{aligned} K(\mu^k_{P_i}, \mu^k_{P_j}) &= K(\|\mu^k_{P_i} - \mu^k_{P_j}\|^2_{\mathcal{H}}) \\ &\approx K(\|\mu^{\tilde{k}}_{\hat{P}_i} - \mu^{\tilde{k}}_{\hat{P}_j}\|^2_{s_k}) \end{aligned}$$

Since the bag kernel K is also translation invariant, we can apply the RFFs on K to get another vector of random features $Z(\cdot) \in \mathbb{R}^{s_K}$ such that

$$K(\|\mu^{\tilde{k}}_{\hat{P}_i} - \mu^{\tilde{k}}_{\hat{P}_j}\|^2_{s_k}) \approx \tilde{K}(\|\mu^{\tilde{k}}_{\hat{P}_i} - \mu^{\tilde{k}}_{\hat{P}_j}\|^2_{s_k}) = Z_{\mu^{\tilde{k}}_{\hat{P}_i}}(\mathbf{w})^T Z_{\mu^{\tilde{k}}_{\hat{P}_j}}(\mathbf{w}).$$

We then use the random feature vector $Z_{\mu^{\tilde{k}}_{\hat{P}_i}}(\mathbf{w})$ to conduct the regression analysis. The detailed algorithm for generating the random feature vector is listed in Algorithm 2.

3.3.2 Expected Max Error of RFF in Distribution Regression

After introducing the RFFs approximation in the distribution regression setting, we are now concerned with the matrix approximation error in this scheme. We will

first derive the expected max error for the kernel approximation

$$\|K\|_\infty = \sup_{P_i,P_j\in \mathsf{M}_1^+(\mathcal{X})} |\tilde{K}(\mu^{\tilde{k}}_{\hat{P}_i}, \mu^{\tilde{k}}_{\hat{P}_j}) - K(\mu^{k}_{\hat{P}_i}, \mu^{k}_{\hat{P}_j})|,$$

followed by the expected error for the distribution regression prediction.

Theorem 3.7. *Let the data kernel k be continuous, shift-invariant and its Fourier transform $p(v)$ has finite second moment $\sigma_k := \mathbb{E}(\|v\|^2) < \infty$. Let the bag kernel K be a radial kernel. Assume K is Lipschitz continuous with constant L and its Fourier transform $q(v)$ also has finite second moment $\sigma_K := \mathbb{E}(\|w\|^2) < \infty$. Then we have the following upper bound for the expected max error*

$$\mathbb{E}(\|K\|_\infty) \leq 4L\sqrt{2\pi}s_k^{-\frac{1}{2}} + \sqrt{2\pi}s_K^{-\frac{1}{2}}$$

where s_k and s_K represent the dimension of the random feature vector for data kernel k and bag kernel K respectively.

Theorem 3.7 provides a first uniform convergence analysis of the expected max error in the distribution regression setting. Our result demonstrates that the approximation error due to the RFFs applied in distribution regression converges to zero at rate of $O(s_k^{-1/2} + s_K^{-1/2})$, implying that the RFFs estimator is consistent.

Now that we have the upper bound for the expected max error for kernel approximation, we are ready to derive the upper bound for the learning risk.

Theorem 3.8. *Let kernels k and K to be the same as in Theorem 3.7. In addition, we assume that there is $B_K < \infty$ such that $K(\mu^k_P, \mu^k_P) \leq B_K, \forall \mu^k_P \in \mathcal{H}$. Let $f^\lambda_z(\mu_{\hat{P}}), f^{\tilde{k}}_z(\mu_{\hat{P}})$ denote the MERR prediction for test distribution P and the RFF approximation of MERR prediction for P respectively. Then we have*

$$\mathbb{E}|f^\lambda_z(\mu_{\hat{P}}) - f^{\tilde{k}}_z(\mu_{\hat{P}})| \leq \sqrt{2\pi}M\frac{\lambda + B_K}{\lambda^2}(4Ls_k^{-\frac{1}{2}} + s_K^{-\frac{1}{2}})$$

Theorem 3.8 demonstrate that the RFFs estimator for distribution regression is consistent and the learning rate is on the order of $O(s_k^{-1/2} + s_K^{-1/2})$. However, as

discussed before, our theorem indicates that the number of features required by the RFFs estimator has to be on the order of n and m in order to match the minimax learning rate for distribution regression. This means that the computational cost is not reduced. Therefore, to demonstrate the efficacy of RFFs, a refined analysis is needed. We leave this as an interesting future direction.

3.4 Proofs

3.4.1 Proof of Theorem 3.4

Proof. Note that $\mathbb{E}\|k\|_\infty = \int_0^\infty P(\|k\|_\infty \geq \epsilon)\, d\epsilon$, and we utilize Theorem 3.1 as follows

$$\begin{aligned}\mathbb{E}(\|k\|_\infty) &= \int_0^\infty P(\|k\|_\infty > \epsilon) d\epsilon \\ &= \int_0^{\epsilon_0} P(\|k\|_\infty > \epsilon) d\epsilon + \int_{\epsilon_0}^\infty P(\|k\|_\infty > \epsilon) d\epsilon \\ &\leq \epsilon_0 + \int_{\epsilon_0}^\infty 256(\frac{\sigma_k|\mathbb{S}|}{\epsilon})^2 \text{exp}(-\frac{s\epsilon^2}{8(d+2)})\, d\epsilon \end{aligned} \tag{3.5}$$

for any positive ϵ_0.

We now define

$$A = 256(\sigma_k|\mathbb{S}|)^2,\ B = \frac{s}{4(d+2)},$$

note that A and B do not depend on ϵ, then Eq. (3.5) can be expressed as

$$\mathbb{E}(\|k\|_\infty) \leq \epsilon_0 + \int_{\epsilon_0}^\infty A\epsilon^{-2}\text{exp}(-B\epsilon^2) d\epsilon$$

We let $\epsilon = t + \epsilon_0$, then we have

$$\begin{aligned}\mathbb{E}(\|k\|_\infty) &\leq \epsilon_0 + \int_0^\infty A(\epsilon_0 + t)^{-2}\text{exp}(-B(\epsilon_0 + t)^2) dt \\ &= \epsilon_0 + \int_0^\infty A\epsilon_0^{-2} e^{-B\epsilon_0^2}(1 + \frac{t}{\epsilon_0})^{-2} e^{-2B\epsilon_0 t} e^{-Bt^2} dt \end{aligned} \tag{3.6}$$

Since we can choose ϵ_0 arbitrarily, we choose ϵ_0 such that $\epsilon_0^{-2}e^{-B\epsilon_0^2} = 1$. Therefore, Eq. (3.6) becomes

$$\begin{aligned}\mathbb{E}(\|k\|_\infty) &\leq \epsilon_0 + \int_0^\infty A(1+\frac{t}{\epsilon_0})^{-2}e^{-2B\epsilon_0 t}e^{-Bt^2}dt \\ &\leq \epsilon_0 + A\int_0^\infty e^{-Bt^2}dt \\ &= \epsilon_0 + \frac{1}{2}A\sqrt{2\pi}B^{-\frac{1}{2}} \\ &= \epsilon_0 + 128\sqrt{2\pi}\sigma_k^2\sqrt{8(d+2)}\sqrt{\frac{\log(s)}{s}}\frac{|\mathbb{S}|^2}{\sqrt{\log(s)}} \end{aligned} \tag{3.7}$$

On the other hand,

$$\epsilon_0^{-2}e^{-B\epsilon_0^2} = 1 \Rightarrow \epsilon_0^{-2} - e^{B\epsilon_0^2} = 0$$

Define

$$f(\epsilon_0) = \epsilon_0^{-2} - e^{B\epsilon_0^2},$$

we can see that f is a decreasing function, and $f(\sqrt{\frac{\log(B)}{B}}) = B(\frac{1}{\log(B)} - 1)$. Since B can be arbitrarily large, we have $f(\sqrt{\frac{\log(B)}{B}}) < 0$. But $f(\epsilon_0) = 0$, we can deduce that

$$\begin{aligned}\epsilon_0 &< \sqrt{\frac{\log(B)}{B}} \\ &= \sqrt{8(d+2)}\sqrt{\frac{\log(s) - \log[8(d+2)]}{s}} \\ &< \sqrt{8(d+2)}\sqrt{\frac{\log(s)}{s}} \end{aligned} \tag{3.8}$$

Combining Eqs. (3.7) and (3.8) and if we require that $|\mathbb{S}| = O(\sqrt{\log(s)})$, we obtain

$$\mathbf{E}(\|k\|_\infty) \leq \left[1 + 128\sqrt{2\pi}\sigma_k^2\frac{|\mathbb{S}|^2}{\sqrt{\log(s)}}\right]\sqrt{8(d+2)}\sqrt{\frac{\log(s)}{s}}.$$

□

3.4.2 Proof of Theorem 3.5

Proof. Using Theorem 3.2, we have that

$$\mathbb{E}(\|k\|_\infty) = \int_0^\infty P(\|k\|_\infty > \epsilon)d\epsilon \leq \int_0^\infty \exp\left(- \frac{s(\epsilon - \frac{h(d,|\mathbb{S}|,\sigma_k)}{\sqrt{s}})^2}{2} \right) d\epsilon.$$

Given any particular s, $\sqrt{\frac{\log|\mathbb{S}|}{s}}$ is upper bounded, we can see that $\frac{h(d,|\mathbb{S}|,\sigma_k)}{s}$ is also upper bounded. Hence, there exists M, such that $\frac{h(d,|\mathbb{S}|,\sigma_k)}{s} < M$. Therefore, the above equation can be rewritten as

$$\begin{aligned}\mathbb{E}(\|k\|_\infty) &\leq \int_0^\infty \exp\left(- \frac{s(\epsilon - M)^2}{2} \right) d\epsilon \\ &\leq \int_{-\infty}^\infty \exp\left(- \frac{s(\epsilon - M)^2}{2} \right) d\epsilon = \sqrt{2\pi} s^{-\frac{1}{2}}.\end{aligned}$$

□

3.4.3 Proof of Theorem 3.6

Proof. By proposition 9 in Sutherland and Schneider (2015). we can see that

$$|\hat{f}^\lambda(x) - \tilde{f}(x)| \leq \frac{M\kappa}{n\lambda_0^2}\|\mathbf{K} - \tilde{\mathbf{K}}\|_2 + \frac{M}{\sqrt{n}\lambda_0}\|K(x,X) - \tilde{K}(x,X)\|_2 \quad (3.9)$$

In addition, we notice that

$$\frac{\|K(x,X) - \tilde{K}(x,X)\|_2}{\sqrt{n}} \leq \sup|\tilde{k}(x,y) - k(x,y)| = \|k\|_\infty \quad (3.10)$$

$$\frac{\|\mathbf{K} - \tilde{\mathbf{K}}\|_2}{n} \leq \|k\|_\infty \quad (3.11)$$

Combining (3.9), (3.10), (3.11), Theorem 3.4 and notice that $\mathbb{E}(l_{\tilde{f}}) \leq \mathbb{E}(l_{\hat{f}^\lambda}) + \mathbb{E}(|\hat{f}^\lambda(x) - \tilde{f}(x)|)$ yields the final result. □

3.4.4 Proof of Theorem 3.7

Proof. Let $\tilde{k}(x,y)$ and $\tilde{K}(\mu^{\tilde{k}}_{\hat{P}_i}, \mu^{\tilde{k}}_{\hat{P}_j})$ be the RFFs approximation of the data kernel $k(x,y)$ and bag kernel $K(\mu^k_{\hat{P}_i}, \mu^k_{\hat{P}_j})$ respectively. By triangle inequality

$$
\begin{aligned}
&\sup_{P_i,P_j\in\mathsf{M}_1^+(\mathcal{X})} |\tilde{K}(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j}) - K(\mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_j})| \\
&= \sup |\tilde{K}(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j}) - K(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j}) + K(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j}) - K(\mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_j})| \\
&\leq \sup |K(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j}) - K(\mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_j})| \qquad (3.12)\\
&\quad + \sup |\tilde{K}(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j}) - K(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j})|. \qquad (3.13)
\end{aligned}
$$

For Eq.(3.13), it is the standard RFF, and hence we have

$$
\mathbb{E}\big[\sup |\tilde{K}(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j}) - K(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j})|\big] \leq \sqrt{2\pi} s_K^{-\frac{1}{2}} \qquad (3.14)
$$

by Theorem 3.5 again.

Since we assume that K is radial kernel and Lipschitz continuous, (3.12) becomes

$$
\begin{aligned}
&\sup |K(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j}) - K(\mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_j})| \\
&\leq \sup L \big| \|\mu^{\tilde{k}}_{\hat{P}_i} - \mu^{\tilde{k}}_{\hat{P}_j}\|^2_{R^{s_k}} - \|\mu^{k}_{\hat{P}_i} - \mu^{k}_{\hat{P}_j}\|^2_{\mathcal{H}} \big| \\
&= \sup L \Big| \big(\langle \mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_i}\rangle_{R^{s_k}} - \langle \mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_i}\rangle_{\mathcal{H}}\big) - 2\big(\langle \mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j}\rangle_{R^{s_k}} - \langle \mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_i}\rangle_{\mathcal{H}}\big) \\
&\quad + \big(\langle \mu^{\tilde{k}}_{\hat{P}_j},\mu^{\tilde{k}}_{\hat{P}_j}\rangle_{R^{s_k}} - \langle \mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_i}\rangle_{\mathcal{H}}\big) \Big| \\
&= \sup L \Big| \mathbb{E}_{\hat{P}_i}\mathbb{E}_{\hat{P}_j}\big[\tilde{k}(x,y) - k(x,y)\big] - 2\mathbb{E}_{\hat{P}_i}\mathbb{E}_{\hat{P}_j}\big[\tilde{k}(x,y) - k(x,y)\big] \\
&\quad + \mathbb{E}_{\hat{P}_j}\mathbb{E}_{\hat{P}_j}\big[\tilde{k}(x,y) - k(x,y)\big] \Big| \\
&\leq \sup L \Big[\big|\mathbb{E}_{\hat{P}_i}\mathbb{E}_{\hat{P}_j}\big[\tilde{k}(x,y) - k(x,y)\big]\big| + \big|2\mathbb{E}_{\hat{P}_i}\mathbb{E}_{\hat{P}_j}\big[\tilde{k}(x,y) - k(x,y)\big]\big| \\
&\quad + \big|\mathbb{E}_{\hat{P}_j}\mathbb{E}_{\hat{P}_j}\big[\tilde{k}(x,y) - k(x,y)\big]\big| \Big] \\
&\leq L\mathbb{E}_{\hat{P}_i}\mathbb{E}_{\hat{P}_j} \sup \big|\tilde{k}(x,y) - k(x,y)\big| + 2\mathbb{E}_{\hat{P}_i}\mathbb{E}_{\hat{P}_j} \sup \big|\tilde{k}(x,y) - k(x,y)\big| \\
&\quad + \mathbb{E}_{\hat{P}_j}\mathbb{E}_{\hat{P}_j} \sup \big|\tilde{k}(x,y) - k(x,y)\big| \qquad (3.15)
\end{aligned}
$$

Hence, the upper bound for the expected value of Eq. (3.12) is

$$\begin{aligned}
&\mathbb{E}\big[\sup|K(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j})-K(\mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_j})|\big]\\
\leq L\Big\{&\mathbb{E}_{\hat{P}_i}\mathbb{E}_{\hat{P}_j}\mathbb{E}\big[\sup\big|\tilde{k}(x,y)-k(x,y)\big|\big]+2\mathbb{E}_{\hat{P}_i}\mathbb{E}_{\hat{P}_j}\mathbb{E}\big[\sup\big|\tilde{k}(x,y)-k(x,y)\big|\big]\\
&+\mathbb{E}_{\hat{P}_j}\mathbb{E}_{\hat{P}_j}\mathbb{E}\big[\sup\big|\tilde{k}(x,y)-k(x,y)\big|\big]\Big\}
\end{aligned}\tag{3.16}$$

Note from Theorem 3.5, we have that $\mathbb{E}\big[\sup\big|k(x,y)-\tilde{k}(x,y)\big|\big]\leq\sqrt{2\pi}s_k^{-\frac{1}{2}}$. Therefore, Eq.(3.16) becomes

$$\mathbb{E}\big[\sup|K(\mu^{\tilde{k}}_{P_i},\mu^{\tilde{k}}_{P_j})-K(\mu^{k}_{P_i},\mu^{k}_{P_j})|\big]\leq 4L\sqrt{2\pi}s_k^{-\frac{1}{2}}\tag{3.17}$$

Combining (3.17) and (3.14) yields our result. □

3.4.5 Proof of Theorem 3.8

Proof. By Eq. (2.7), we have $f_z^\lambda(\mu_{\hat{P}})=\mathbf{k}_{\hat{P}}(\mathbf{K}+m\lambda I)^{-1}y$. By proposition 9 in Sutherland and Schneider (2015) and recall we have $\|\mathbf{y}\|_2\leq M$, we thus have

$$|f_z^\lambda(\mu_{\hat{P}})-f_z^{\tilde{k}}(\mu_{\hat{P}})|\leq\frac{MB_K}{m\lambda^2}\|\mathbf{K}-\tilde{\mathbf{K}}\|_2+\frac{M}{\sqrt{m\lambda}}\|\mathbf{k}_{\hat{P}}-\tilde{\mathbf{k}}_{\hat{P}}\|_2\tag{3.18}$$

In addition, we notice that

$$\frac{\|\mathbf{K}-\tilde{\mathbf{K}}\|_2}{m}\leq\sup|\tilde{K}(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j})-K(\mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_j})|\tag{3.19}$$

$$\frac{\|\mathbf{k}_{x_t}-\tilde{\mathbf{k}}_{x_t}\|_2}{\sqrt{m}}\leq\sup|\tilde{K}(\mu^{\tilde{k}}_{\hat{P}_i},\mu^{\tilde{k}}_{\hat{P}_j})-K(\mu^{k}_{\hat{P}_i},\mu^{k}_{\hat{P}_j})|\tag{3.20}$$

Applying Theorem 3.7 into Eq.(3.19) and (3.20), and combining with Eq.(3.18) proved our claim. □

3.5 Conclusion

In this chapter, we derive a very first result on the properties of the RFFs approximation under kernel ridge regression and distribution regression settings. By adopting the kernel matrix approximation approach and conducting analysis in the worst case scenario, we show that the learning risk of the RFF estimators converge at the rate of $O(s^{-\frac{1}{2}})$ in the KRR setting, and $O(s_k^{-\frac{1}{2}} + s_K^{-\frac{1}{2}})$ in distribution regression setting. The results guarantee the statistical consistency of the RFFs approximation. However, in order to ensure the $O(n^{-\frac{1}{2}})$ risk convergence rate, the number of features required is close to n, the number of data points. As a result, it looks like we do not obtain any computational gain by using RFFs approximation, which is not the case in practice. Therefore, in next chapter, we employ a different approach and borrow some advanced machinery from learning theory to overcome this issue.

Chapter 4 Towards a Unified Analysis of RFFs

This chapter is based on the following self-contained paper Li et al. (2019b):

***Zhu Li**, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards a unified analysis of random Fourier features. In International Conference on Machine Learning (ICML), pages 3905–3914, 2019.*

The extended version of the paper which includes new results given here in Section 4.3.2.2 and Theorem 4.9 is recently published in the *Journal of Machine Learning Research* (Li et al., 2021).

Random Fourier features is a widely used, simple, and effective technique for scaling up kernel methods. The existing theoretical analysis of the approach, however, remains focused on specific learning tasks and typically gives pessimistic bounds which are at odds with the empirical results. We tackle these problems and provide the first unified risk analysis of learning with RFFs using the squared error and Lipschitz continuous loss functions. In our bounds, the trade-off between the computational cost and the learning risk convergence rate is problem specific and expressed in terms of the regularization parameter and the *number of effective degrees of freedom*. We study both the standard RFFs method for which we improve the existing bounds on the number of features required to guarantee the corresponding minimax risk convergence rate of kernel ridge regression, as well as a data-dependent modification which samples features proportional to *ridge leverage scores* and further reduces the required number of features. As ridge leverage scores are expensive to compute, we devise a simple approximation scheme which provably reduces the computational cost without loss of statistical efficiency. Our empirical results illustrate the effectiveness of the proposed scheme relative to the standard RFFs method.

4.1 Introduction

In Chapter 3, we have presented our first analysis of RFFs from the kernel approximation perspective. However, the theoretical understanding of statistical properties of RFFs is incomplete, and the question of how many features are needed, in order to obtain a method with performance provably comparable to the original one, remains without a definitive answer. Currently, there are two main lines of research addressing this question. The first line considers the approximation error of the kernel matrix itself (e.g., see Rahimi and Recht, 2007; Sriperumbudur and Szabó, 2015; Sutherland and Schneider, 2015, and references therein) and bases performance guarantees on the accuracy of this approximation. However, all of these works require $\Omega(n)$ features (n being the number of observations), which translates to no computational savings at all and is at odds with empirical findings. Realizing that the approximation of kernel matrices is just a means to an end, the second line of research aims at directly studying the risk and generalization properties of RFFs in various supervised learning scenarios. Arguably, first such result is already in Rahimi and Recht (2009), where supervised learning with Lipschitz continuous loss functions is studied. However, the bounds therein still require a pessimistic $\Omega(n \log n)$ number of features and cannot demonstrate the efficiency of RFFs theoretically. In Bach (2017b), the generalization properties are studied from a function approximation perspective, showing for the first time that fewer features could preserve the statistical properties of the original method, but in the case where a certain data-dependent sampling distribution is used instead of the spectral measure. These results also do not apply to kernel ridge regression and the mentioned sampling distribution is typically itself intractable. Avron et al. (2017) study the RFFs for kernel ridge regression in the fixed design setting. They show that it is possible to use $o(n)$ features and have the risk of the linear ridge regression

estimator based on RFFs close to the risk of the original kernel estimator, also relying on a modification to the sampling distribution. However, their result restricts the data distribution to have finite support, and a tractable method to sample from a modified distribution is proposed for the Gaussian kernel only. A highly refined analysis of kernel ridge regression is given by Rudi and Rosasco (2017), where it is shown that $\Omega(\sqrt{n}\log n)$ features suffices for an optimal $O(1/\sqrt{n})$ learning rate in a minimax sense (Caponnetto and De Vito, 2007). Moreover, the number of features can be reduced even further if a data-dependent sampling distribution is employed. While these are groundbreaking results, guaranteeing computational savings without any loss of statistical efficiency, they require some technical assumptions that are difficult to verify. Moreover, to what extent the bounds can be improved by utilizing data-dependent distributions still remains unclear. Finally, it does not seem straightforward to generalize the approach of Rudi and Rosasco (2017) to kernel support vector machines (SVM) and/or kernel logistic regression (KLR). Recently, Sun et al. (2018) have provided novel bounds for RFFs in the SVM setting, assuming the Massart's low noise condition and that the target hypothesis lies in the corresponding reproducing kernel Hilbert space (RKHS). The bounds, however, require the sample complexity and the number of features to be exponential in the dimension of the instance space and this can be problematic for high dimensional instance spaces. The theoretical results are also restricted to the hinge loss and the 0-1 loss (without means to generalize to other loss functions) and require optimized features. For a comprehensive review on the theoretical analysis of the RFFs, we recommend the interested readers to refer to Liu et al. (2020) for more details.

In this chapter, we address the gaps mentioned above by making the following contributions:

- We devise a simple framework for the unified analysis of generalization

properties of RFFs, which applies to kernel ridge regression, as well as to kernel support vector machines and logistic regression.

- For the plain RFFs sampling scheme (Section 4.3.1.1), we provide, to the best of our knowledge, the sharpest results on the number of features required. In particular, we show that already with $\Omega(\sqrt{n}\log d_{\mathbf{K}}^{\lambda})$ random features one can obtain the minimax learning rate of kernel ridge regression (Caponnetto and De Vito, 2007), where $d_{\mathbf{K}}^{\lambda}$ corresponds to the notion of *the number of effective degrees of freedom* (Bach, 2013) with $d_{\mathbf{K}}^{\lambda} \ll n$ and $\lambda := \lambda(n)$ is the regularization parameter.

- In the case of a modified data-dependent sampling distribution (Section 4.3.1.2) (the so called *empirical ridge leverage score distribution*), we demonstrate that $\Omega(d_{\mathbf{K}}^{\lambda})$ features suffice for the learning risk to converge at $O(\lambda)$ rate in kernel ridge regression. In addition, we show that the excess risk convergence rate of the estimator based on RFFs can (depending on the decay rate of the spectrum of the kernel function) be upper bounded by $O(\frac{\log n}{n})$ or even $O(\frac{1}{n})$, which implies much faster convergence than the standard $O(\frac{1}{\sqrt{n}})$ rate featuring in the majority of previous bounds.

- For plain RFFs in the Lipschitz continuous loss and 0-1 loss setting (Section 4.3.2.1), we show that $\Omega(1/\lambda)$ features are sufficient to ensure $O(\sqrt{\lambda})$ learning risk rate in kernel support vector machines and kernel logistic regression. Moreover, using the empirical ridge leverage score distribution, we show that $\Omega(d_{\mathbf{K}}^{\lambda})$ features are sufficient to guarantee $O(\sqrt{\lambda})$ risk convergence rate in these two learning settings.

- Similarly, under the low noise assumption (Section 4.3.2.2), our refined analysis for the Lipschitz continuous loss function demonstrates that it is possible to achieve $O(1/n)$ excess risk convergence rate. The required number of

features can be $\Omega(\log n \log\log n)$ when learning using the empirical leverage score distribution, or even constant in some benign cases.

- Finally, as the empirical ridge leverage scores distribution is typically costly to compute, we give a fast algorithm to generate samples from the approximated empirical leverage distribution (Section 4.4). Utilizing these samples one can significantly reduce the computation time during the in-sample prediction and testing stages, i.e., $O(n \log n \log\log n)$ and $O(\log n \log\log n)$, respectively. We also include a proof that characterizes the trade-off between the computational cost and the learning risk of the algorithm, showing that the statistical efficiency can be preserved while provably reducing the required computational cost.

4.1.1 Additional Literature

Recent studies reveal that the random feature models have many connections with deep learning models (see, e.g., Ba et al., 2019; d'Ascoli et al., 2020; Dhifallah and Lu, 2020; Gerace et al., 2020; Hastie et al., 2019; Liao et al., 2020; Mei and Montanari, 2019). As a result, the research interest of random feature methods shifts from investigating the trade-off between the computational cost and the prediction accuracy to understanding the generalization properties of overparameterized models. We briefly remark that utilizing the connection between random feature methods and neural networks allows us to obtain many interesting insights on the generalization properties of overpameterized models such as a precise description of the double descent learning curve (Adlam and Pennington, 2020b; Belkin et al., 2019b; d'Ascoli et al., 2020; Hastie et al., 2019; Mei and Montanari, 2019), the implicit regularization of random features (Jacot et al., 2020), the fitting capability of random feature methods and neural networks (Ghorbani et al., 2021; Mei et al., 2021; Yehudai and Shamir, 2019). Understanding this connection and

the generalization phenomenon of overparameterized models is the focus of our next chapter.

4.2 Background

In this section, we provide some notations and preliminary results that will be used throughout the chapter. Henceforth, we denote the Euclidean norm of a vector $\mathbf{a} \in \mathbb{R}^n$ with $\|\mathbf{a}\|_2$ and the operator norm of a matrix $A \in \mathbb{R}^{n_1 \times n_2}$ with $\|A\|_2$. Let $\mathcal{H}$ be a Hilbert space with $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ as its inner product and $\| \cdot \|_{\mathcal{H}}$ as its norm. We use $\mathrm{Tr}(\cdot)$ to denote the trace of an operator or a matrix. Given a measure $d\rho$, we use $L_2(d\rho)$ to denote the space of square-integrable functions with respect to $d\rho$. Finally, for the notations and definitions that are related to kernel supervised learning and RFFs, please refer to Chapter 2.

4.2.1 Rademacher Complexity

To characterize the performance of a learning algorithm, we need to take into account the complexity of its hypothesis space. Below, we first introduce a particular measure of the complexity over function spaces known as *Rademacher complexity* (Bartlett and Mendelson, 2002). Then, we give two lemmas that demonstrate how Rademacher complexity of a RKHS can be linked to the corresponding kernel and how the excess risk can be computed via Rademacher complexity.

Definition 4.1. *Suppose that $\{x_1 \cdots, x_n\}$ are independent samples selected according to P_x. Let $\mathcal{H}$ be a class of functions mapping $\mathcal{X}$ to $\mathbb{R}$. Then, the random variable known as the empirical Rademacher complexity is defined as*

$$\hat{R}_n(\mathcal{H}) = \mathbb{E}_\sigma \left[\sup_{f \in \mathcal{H}} \left| \frac{2}{n} \sum_{i=1}^{n} \sigma_i f(x_i) \right| \mid x_1, \cdots, x_n \right],$$

where $\sigma_1, \cdots, \sigma_n$ are independent samples from the uniform distribution over the two element set $\{\pm 1\}$. The corresponding Rademacher complexity is then defined as the expectation of the empirical Rademacher complexity

$$R_n(\mathcal{H}) = \mathbb{E}\Big[\hat{R}_n(\mathcal{H})\Big] ,$$

where the expectation is taken with respect to n-element sets of indepedent samples from P_x.

The following lemma provides an upper bound on the Rademacher complexity of a hypothesis space that is a subspace of the RKHS with a kernel k.

Lemma 4.1. *(Bartlett and Mendelson, 2002) Let $\mathcal{H}_0$ be the unit ball that is centered at the origin of the RKHS $\mathcal{H}$ associated with a kernel k. Then, we have that $R_n(\mathcal{H}_0) \leq (1/n)\mathbb{E}_X[\sqrt{Tr(\mathbf{K})}]$, where $\mathbf{K}$ is the Gram-matrix for kernel k over an independent and identically distributed sample $X = \{x_1, \cdots, x_n\}$.*

Lemma 4.2 states that the expected excess risk convergence rate of a particular estimator in $\mathcal{H}$ not only depends on the number of data points, but also on the complexity of $\mathcal{H}$ and how it interacts with the loss function.

Lemma 4.2. *(Bartlett and Mendelson, 2002, Theorem 8) Let $\{x_i, y_i\}_{i=1}^n$ be i.i.d samples from $P(x, y)$ and let $\mathcal{H}$ be the space of functions mapping from $\mathcal{X}$ to $\mathbb{R}$. Denote a loss function with $l : \mathcal{Y} \times \mathbb{R} \to [0, 1]$ and recall that, for all $f \in \mathcal{H}$, the expected and corresponding empirical learning risk functions are denoted with $\mathbb{E}[l_f]$ and $\mathbb{E}_n[l_f] = (1/n)\sum_{i=1}^n l(y_i, f(x_i))$, respectively. Then, for a sample of size n, for all $f \in \mathcal{H}$ and $\delta \in (0, 1)$, with probability $1 - \delta$, we have that*

$$\mathbb{E}[l_f] \leq \mathbb{E}_n[l_f] + R_n(l \circ \mathcal{H}) + \sqrt{\frac{8\log(2/\delta)}{n}} ,$$

where $l \circ \mathcal{H} = \{(x, y) \mapsto l(y, f(x)) - l(y, 0) \mid f \in \mathcal{H}\}$.

Note that the risk bound is given by the Rademacher complexity term $R_n(l \circ \mathcal{H})$ defined on the transformed space $l \circ \mathcal{H}$, which is obtained via composition of $f \in \mathcal{H}$ and the loss function l. This term is, in general, different from $R_n(\mathcal{H})$. However, in the case when l is Lipschitz continuous with constant L_l, then $R_n(l \circ \mathcal{H}) \leq 2L_l R_n(\mathcal{H})$ by Theorem 12 in Bartlett and Mendelson (2002).

4.2.2 Local Rademacher Complexity

When characterizing the finite sample behavior of learning risk, the notion of Rademacher complexity introduced in the previous section does not typically give the optimal convergence rates. This is because Rademacher complexity considers the behavior of the empirical learning risk over the whole hypothesis space, while the estimator returned by the regression is typically in a neighborhood around the optimal estimator. Hence, in our refined analysis we rely on the so called *local Rademacher complexity*. Before illustrating this concept, we first recall that given a hypothesis $f \in \mathcal{H}$, we denote its expectation and finite sample average with $\mathbb{E}[f]$ and $\mathbb{E}_n[f]$, respectively. The notion of local Rademacher complexity is typically introduced via the so called *sub-root function*. This sub-root function is used to obtain a fixed point of the local Rademacher complexity, which gives a sharper convergence rate than the notion introduced in previous section. Below, we first give the definition and a useful property of the sub-root function. We then review a theorem that relates the notion of local Rademacher complexity and learning risk.

Definition 4.2. *Let* $\psi : [0, \infty) \to [0, \infty)$ *be a function. Then,* $\psi(r)$ *is called a sub-root function if, for all* $r > 0$, $\psi(r)$ *is non-decreasing and* $\frac{\psi(r)}{r}$ *is non-increasing.*

A sub-root function has the following property.

Lemma 4.3. *(Bartlett et al., 2005, Lemma 3.2) If* $\psi(r)$ *is a sub-root function, then*

$\psi(r) = r$ has a unique positive solution r^*. In addition, we have that $r \geq \psi(r)$ if and only if $r \geq r^*$.

In Lemma 4.2, we can see that the difference between the expected and empirical learning risks, $\mathbb{E}[l_f]$ and $\mathbb{E}_n[l_f]$, is upper bounded by $O(\frac{1}{\sqrt{n}})$. This rate can be further improved with local Rademacher complexity. The reason for the slow learning rate is because the bound accounts for the difference between $\mathbb{E}[l_f]$ and $\mathbb{E}_n[l_f]$ using the global Rademacher complexity. Inspecting the definition of $R_n(\mathcal{H})$ (Definition 4.1), we can see that $R_n(\mathcal{H})$ is defined by considering the whole hypothesis space, as the supremum operator is applied over all functions in $\mathcal{H}$. However, as discussed before, learning algorithms typically return functions that are in the neighborhood around the optimal estimator. Hence, using $R_n(\mathcal{H})$ unnecessarily enlarges the space that we are interested in.

As empirical estimators returned by learning algorithms typically have low learning risk as well as low variance, we could instead consider the alternative space $\mathcal{H}_r := \{f \in \mathcal{H} : \mathbb{E}[f^2] \leq r\}$ for some given value $r \in \mathbb{R}$. In this way, we greatly reduce the complexity of the function space at hand and can provide a sharper convergence rate. The following results from Bartlett et al. (2005) details how this idea can be used to describe the learning risk behavior.

Lemma 4.4. *(Bartlett et al., 2005, Theorem 4.1) Let $\mathcal{H}$ be a class of functions with bounded ranges and assume that there is some constant $B > 0$ such that, for all $f \in \mathcal{H}$, $\mathbb{E}[f^2] \leq B\mathbb{E}[f]$. Let $\hat{\psi}_n$ be a sub-root function and let $\hat{r}^*$ be the fixed point of $\hat{\psi}_n$, i.e., $\hat{\psi}_n(\hat{r}^*) = \hat{r}^*$. Fix any $\delta \in (0, 1)$, and assume that for any $r \geq \hat{r}^*$,*

$$\hat{\psi}_n(r) \geq c_1 \hat{R}_n\{f \in \text{star}(\mathcal{H}, 0) \mid \mathbb{E}_n[f^2] \leq r\} + \frac{c_2}{n} \log \frac{1}{\delta},$$

where

$$\text{star}(\mathcal{H}, f_0) = \{f_0 + \alpha(f - f_0) \mid f \in \mathcal{H} \wedge \alpha \in [0, 1]\} .$$

Then for all $D > 1$ *and* $f \in \mathcal{H}$*, with probability greater than* $1 - \delta$*,*

$$\mathbb{E}[f] \leq \frac{D}{D-1}\mathbb{E}_n[f] + \frac{6D}{B}\hat{r}^* + \frac{c_3}{n}\log\frac{1}{\delta} ,$$

where c_1*,* c_2 *and* c_3 *are some constants.*

Note that this theorem bounds the difference between $\mathbb{E}[f]$ and $\mathbb{E}_n[f]$. We will show later (Section 4.6.3), with a simple transformation, that this result can be used to bound the difference between the expected and empirical learning risks for estimators based on RFFs.

We have seen that in the above theorem, we can use the fixed point of the sub-root function to upper bound the learning rate. It is, however, not clear how to obtain the explicit formula for the fixed point using this result. Fortunately, in the setting of learning with kernel k and the corresponding RKHS, we can derive such results. The following lemma provides us with an upper bound on local Rademacher complexity through the eigenvalues of the Gram-matrix.

Lemma 4.5. *(Bartlett et al., 2005, Lemma 6.6) Let* k *be a positive definite kernel function with RKHS* $\mathcal{H}$ *and let* $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_n$ *be the eigenvalues of the normalized Gram-matrix* $(1/n)\mathbf{K}$*. Then, for all* $r > 0$ *and* $f \in \mathcal{H}$*,*

$$\hat{R}_n\{f \in \mathcal{H} \mid \mathbb{E}_n[f^2] \leq r\} \leq \left(\frac{2}{n}\sum_{i=1}^{n}\min\{r, \hat{\lambda}_i\}\right)^{1/2} .$$

4.3 Theoretical Analysis

In this section, we provide a unified analysis for the generalization properties of learning with RFFs. Our analysis is split into two cases/settings: *i*) we start with a bound for learning with the squared error loss function (Section 4.3.1) and *ii*) then extend these results to learning problems with Lipschitz continuous loss functions (Section 4.3.2). In addition, in each of the cases, we will present two different analyses. In the worst case analysis, we provide the conditions for the estimator to achieve the minimax learning rate of the corresponding kernel-based estimator. After that, we present a refined analysis and show that the estimators based on RFFs are able to achieve faster learning rates if the learning problem exhibits certain benign properties. Before proceeding with our theoretical contributions, we first enumerate the generic assumptions that we made in Section 4.2.

1. For a learning problem with kernel k (and corresponding RKHS $\mathcal{H}$) defined as in Eq. (2.3), we assume that $f_{\mathcal{H}} = \arg\inf_{f\in\mathcal{H}} \mathbb{E}[l_f]$ always exists and has a bounded $\mathcal{H}$-norm. Moreover, we (without further loss of generality) restrict our analysis to the unit ball of $\mathcal{H}$, i.e., the hypothesis space is given by $\|f\|_{\mathcal{H}} \leq 1$;

2. We assume that the kernel k has the decomposition as in Eq. (2.9) with $|z(w,x)| < z_0 \in (0,\infty)$;

3. For kernel k, denote with $\lambda_1 \geq \cdots \geq \lambda_n$ the eigenvalues of the kernel matrix $\mathbf{K}$. We assume that the regularization parameter satisfies $0 \leq n\lambda \leq \lambda_1$.

Intuitively, Assumption 3 requires that the signal λ_1 is stronger than the added regularization term $n\lambda$. More specifically, the in-sample prediction of a kernel ridge regression problem is $\mathbf{K}(\mathbf{K}+n\lambda I)^{-1}Y$. The largest eigenvalue of $\mathbf{K}(\mathbf{K}+n\lambda I)^{-1}$ is $\frac{\lambda_1}{(\lambda_1+n\lambda)}$. If $n\lambda > \lambda_1$, then the in-sample prediction is essentially dominated by

$n\lambda$, which could lead to under-fitting.

Throughout our analysis, we will use the assumptions listed above and will not be restating them unless problem-specific clarifications are required.

4.3.1 Learning with the Squared Error Loss

In this section, we consider learning with the squared error loss, i.e., $l(y, f(x)) = (y - f(x))^2$, where the learning is known as KRR. We make the following assumption specific for the KRR problem.

A.1 $y = f_*(x) + \epsilon$ with $\mathbb{E}[\epsilon] = 0$ and $\text{Var}[\epsilon] = \sigma^2$. Furthermore, we assume that y is a bounded random variable, i.e., $|y| \leq y_0$;

For regression problem, we have the target regression function $f_* = \mathbb{E}[y \mid x]$. Note that f_* may be different from $f_{\mathcal{H}}$ as it is not necessarily contained in our hypothesis space $\mathcal{H}$.

In the RFFs setting, the KRR problem can be reduced to solving a linear system $(\tilde{\mathbf{K}} + n\lambda\mathbf{I})\alpha = Y$, with $Y = [y_1, \cdots, y_n]^T$. Typically, an approximation of the kernel function based on RFFs is employed in order to effectively reduce the computational cost and scale kernel ridge regression to problems with a large number of examples. More specifically, for a vector of observed labels Y the goal is to find a hypothesis $\tilde{\mathbf{f}}_x = \mathbf{Z}\beta$ that minimizes $\|Y - \tilde{\mathbf{f}}_x\|_2^2$ while having good generalization properties. In order to achieve this, one needs to control the complexity of hypotheses defined by RFFs and avoid over-fitting. Hence, we would like to estimate the norm of a function $\tilde{f} \in \tilde{\mathcal{H}}$ for the purpose of regularization. The following proposition (originally from Bach, 2017b) gives an upper bound on that norm (a proof is given in Section 4.6).

Proposition 4.1. *Assume that the RKHS $\mathcal{H}$ with kernel k admits a decomposition*

as in Eq. (2.9) and denote by $\tilde{\mathcal{H}} := \{\tilde{f} \mid \tilde{f} = \sum_{i=1}^{s} \alpha_i z(v_i, \cdot), \alpha_i \in \mathbb{R}\}$ *the RKHS with kernel* $\tilde{k}$ *(see Eq. 2.10). Then, for all* $\tilde{f} \in \tilde{\mathcal{H}}$ *it holds that* $\|\tilde{f}\|_{\tilde{\mathcal{H}}}^2 \leq s\|\alpha\|_2^2$.

According to Proposition 4.1, the learning problem with RFFs and the squared error loss can be cast as

$$\beta_\lambda := \underset{\beta \in \mathbb{R}^s}{\arg\min} \ \frac{1}{n}\|Y - \mathbf{Z}_q\beta\|_2^2 + \lambda s\|\beta\|_2^2 \,. \tag{4.1}$$

This is simply a linear ridge regression problem in the space of Fourier features. We denote the optimal hypothesis function returned by Eq. (4.1) to be $\tilde{f}_\beta^\lambda$. The function can be parameterized by β_λ and its in-sample evaluation is given by $\tilde{\mathbf{f}}_\beta^\lambda = \mathbf{Z}_q\beta_\lambda$, where $\beta_\lambda = (\mathbf{Z}_q^T\mathbf{Z}_q + ns\lambda\mathbf{I})^{-1}\mathbf{Z}_q^T Y$. Since $\mathbf{Z}_q \in \mathbb{R}^{n\times s}$, the computational and space complexities are $O(s^3 + ns^2)$ and $O(ns)$. Thus, significant savings can be achieved using estimators with $s \ll n$. To assess the effectiveness of such estimators, it is important to understand the relationship between the excess learning risk and the choice of s.

4.3.1.1 Worst Case Analysis

In this section, we provide a bound on the required number of RFFs with respect to the worst case (in the minimax sense) of the corresponding kernel ridge regression problem, i.e., learning rate $O(\frac{1}{\sqrt{n}})$. The following theorem gives a general result while taking into account both the number of features s and a sampling strategy for selecting them.

Theorem 4.1. *Suppose that Assumption A.1 holds and let* $\tilde{l} : \mathcal{V} \to \mathbb{R}$ *be a measurable function such that* $\tilde{l}(v) \geq l_\lambda(v)$ $(\forall v \in \mathcal{V})$ *with* $d_{\tilde{l}} = \int_{\mathcal{V}} \tilde{l}(v)dv < \infty$. *Suppose also that* $\{v_i\}_{i=1}^s$ *are sampled independently from the probability density function*

$q(v) = \tilde{l}(v)/d_{\tilde{l}}$. *If*

$$s \geq 5d_{\tilde{l}} \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta} ,$$

then for all $\delta \in (0,1)$*, with probability* $1-\delta$*, the excess risk of* $\tilde{f}_{\beta}^{\lambda}$ *can be upper bounded by*

$$\mathbb{E}[l_{\tilde{f}_{\beta}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \leq 4\lambda + O\left(\frac{1}{\sqrt{n}}\right) + \mathbb{E}[l_{\hat{f}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] . \tag{4.2}$$

Theorem 4.1 expresses the trade-off between the computational and statistical efficiency through the regularization parameter λ, the effective dimension of the problem $d_{\mathbf{K}}^{\lambda}$, and the normalization constant $d_{\tilde{l}}$ of the sampling distribution. The decay rate of the regularization parameter is used as a key quantity (Caponnetto and De Vito, 2007; Rudi and Rosasco, 2017) and its choice can be linked to the complexity of the target regression function $f_*(x) = \int y d\rho(y \mid x)$. In particular, Caponnetto and De Vito (2007) show that the minimax risk convergence rate for kernel ridge regression is $O(\frac{1}{\sqrt{n}})$. Setting $\lambda \propto \frac{1}{\sqrt{n}}$, we observe that the estimator $\tilde{f}_{\beta}^{\lambda}$ attains the worst case minimax rate of kernel ridge regression. As a consequence of Theorem 4.1, we have the following bounds on the number of required features for the two strategies: *leverage weighted* RFFs (Corollary 4.1) and *plain* RFFs (Corollary 4.2).

Corollary 4.1. *If the probability density function from Theorem 4.1 is the empirical ridge leverage score distribution* $q^*(v)$*, then the upper bound on the risk from Eq. (4.2) holds for all* $s \geq 5d_{\mathbf{K}}^{\lambda} \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta}$.

Proof. For this corollary, we set $\tilde{l}(v) = l_{\lambda}(v)$ and deduce $d_{\tilde{l}} = \int_{\mathcal{V}} l_{\lambda}(v) dv = d_{\mathbf{K}}^{\lambda}$. □

Theorem 4.1 and Corollary 4.1 have several implications on the choice of λ and

s. First, we could pick $\lambda \in O(n^{-1/2})$ that implies the worst case minimax rate for kernel ridge regression (Bartlett et al., 2005; Caponnetto and De Vito, 2007; Rudi and Rosasco, 2017) and observe that in this case s is proportional to $d_{\mathbf{K}}^{\lambda} \log d_{\mathbf{K}}^{\lambda}$. As $d_{\mathbf{K}}^{\lambda}$ is determined by the learning problem (i.e., the marginal distribution P_x), we can consider several different cases. In the best case, where the number of positive eigenvalues is finite, implying that $d_{\mathbf{K}}^{\lambda}$ does not grow with n, we then have that even with a constant number of features, we are able to achieve the $O(1/\sqrt{n})$ learning rate. Next, if the eigenvalues of $\mathbf{K}$ exhibit a geometric/exponential decay, i.e., $\lambda_i \propto R_0 r^i$ with a constant $R_0 > 0$ (this can happen in scenario where we have a Gaussian kernel and a sub-Gaussian marginal distribution P_x), we then know that $d_{\mathbf{K}}^{\lambda} \leq \log(R_0/\lambda)$ (Bach, 2017b), implying $s \geq \log n \log\log n$. Hence, significant savings can be obtained with $O(n \log^4 n + \log^6 n)$ computational and $O(n \log^2 n)$ storage complexities of linear ridge regression over RFFs, as opposed to $O(n^3)$ and $O(n^2)$ costs (respectively) in the kernel ridge regression setting.

In the case of a slower decay (e.g., $\mathcal{H}$ is a Sobolev space of order $t \geq 1$) with $\lambda_i \propto R_0 i^{-2t}$ (R_0 is some constant), we have $d_{\mathbf{K}}^{\lambda} \leq (R_0/\lambda)^{1/(2t)}$ and $s \geq n^{1/(4t)} \log n$. Hence, substantial computational savings can be achieved even in this case. Furthermore, in the worst case with λ_i close to $R_0 i^{-1}$, our bound implies that $s \geq \sqrt{n} \log n$ features are sufficient, recovering the result from Rudi and Rosasco (2017).

Corollary 4.2. *If the probability density function from Theorem 4.1 is the spectral measure $p(v)$ from Eq. (2.9), then the upper bound on the learning risk from Eq. (4.2) holds for all $s \geq 5\frac{z_0^2}{\lambda} \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta}$.*

Proof. We set $\tilde{l}(v) = p(v)\frac{z_0^2}{\lambda}$ and obtain $d_{\tilde{l}} = \int_{\mathcal{V}} p(v)\frac{z_0^2}{\lambda} dv = \frac{z_0^2}{\lambda}$. □

Corollary 4.2 addresses plain RFFs and states that if s is chosen to be greater than $\sqrt{n} \log d_{\mathbf{K}}^{\lambda}$ and $\lambda \propto \frac{1}{\sqrt{n}}$ then the minimax risk convergence rate is guaranteed.

SAMPLING SCHEME	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
WEIGHTED RFFS	finite rank	$s \in \Omega(1)$	$O(1/\sqrt{n})$
	$\lambda_i \propto A^i$	$s \in \Omega(\log n \cdot \log\log n)$	
	$\lambda_i \propto i^{-2t}$ $(t \geq 1)$	$s \in \Omega(n^{1/2t} \cdot \log n)$	
	$\lambda_i \propto i^{-1}$	$s \in \Omega(\sqrt{n} \cdot \log n)$	
PLAIN RFFS	finite rank	$s \in \Omega(\sqrt{n})$	$O(1/\sqrt{n})$
	$\lambda_i \propto A^i$	$s \in \Omega(\sqrt{n} \cdot \log\log n)$	
	$\lambda_i \propto i^{-2t}$ $(t \geq 1)$	$s \in \Omega(\sqrt{n} \cdot \log n)$	
	$\lambda_i \propto i^{-1}$	$s \in \Omega(\sqrt{n} \cdot \log n)$	

Table 4.1: The worst case trade-offs between computational complexity and statistical efficiency for the squared error loss.

In the case of finitely many positive eigenvalues, $s \geq \sqrt{n}$ features are needed to obtain $O(\frac{1}{\sqrt{n}})$ convergence rate. When the eigenvalues have an exponential decay, we obtain the same convergence rate with only $s \geq \sqrt{n}\log\log n$ features, which is an improvement compared to a result by Rudi and Rosasco (2017) where $s \geq \sqrt{n}\log n$ is needed. For the other two cases, we derive $s \geq \sqrt{n}\log n$ and recover the results from Rudi and Rosasco (2017). Table 4.1 provides a summary of the trade-offs between computational complexity and statistical efficiency for the worst case scenario.

4.3.1.2 Refined Analysis

In this section, we provide a more refined analysis with risk convergence rates faster than $O(\frac{1}{\sqrt{n}})$, depending on the spectrum decay of the kernel and/or the complexity of the target regression function. The main reason for obtaining a faster rate compared to the previous section is the reliance on local Rademacher complexity, instead of the global one (detailed proofs can be found in Section 4.6.3).

Theorem 4.2. *Suppose that Assumption A.1 holds and that the conditions on sampling measure $\tilde{l}$ from Theorem 4.1 apply to this setting. If*

$$s \geq 5d_{\tilde{l}} \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta}$$

then for all $D > 1$ and $\delta \in (0,1)$, with probability $1-\delta$, the excess risk of $\tilde{f}_{\beta}^{\lambda}$ can be bounded by

$$\mathbb{E}[l_{\tilde{f}_{\beta}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \leq \frac{12D}{B}\hat{r}_{\mathcal{H}}^{*} + 4\frac{D}{D-1}\lambda + O\left(\frac{1}{n}\right) + \mathbb{E}[l_{\hat{f}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \,. \quad (4.3)$$

Furthermore, denoting the eigenvalues of the normalized kernel matrix $(1/n)\mathbf{K}$ with $\{\hat{\lambda}_i\}_{i=1}^{n}$, we have that

$$\hat{r}_{\mathcal{H}}^{*} \leq \min_{0 \leq h \leq n} \left(e_0 \frac{h}{n} + \sqrt{\frac{1}{n}\sum_{i>h} \hat{\lambda}_i} \right) , \quad (4.4)$$

where $B, e_0 > 0$ are some constant and $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_n$.

Theorem 4.2 covers a wide range of cases and can provide tight/sharp risk convergence rates. In particular, note that $\hat{r}_{\mathcal{H}}^{*}$ has an upper bound of $O(1/\sqrt{n})$ in all cases, which happens when we let $h = 0$ and the spectrum decays polynomially as $O(1/n^t)$ with $t > 1$. On the other hand, if the eigenvalues decay exponentially, then setting $h = \lceil \log n \rceil$ implies that $\hat{r}_{\mathcal{H}}^{*} \leq O(\log n / n)$. In the best case, when the kernel function has only finitely many positive eigenvalues, we have that $\hat{r}_{\mathcal{H}}^{*} \leq O(1/n)$ by letting h be any fixed value larger than the number of positive eigenvalues. These different upper bounds provide insights into various trade-offs between computational complexity and statistical efficiency. We now split the discussion into two cases: weighted sampling with empirical leverage score and plain sampling.

Under the weighted sampling scheme, if the eigenvalues decay polynomially, i.e., $\lambda_i \propto i^{-t}$ with $t > 1$, then the learning rate is upper bounded by $O(1/\sqrt{n})$. In this case, we have $d_{\mathbf{K}}^{\lambda} \leq (R_0/\lambda)^{1/t} \leq n^{1/2t}$ and consequently $s \geq n^{1/2t} \log n$. On the other hand, if the eigenvalues decay exponentially, we have $d_{\mathbf{K}}^{\lambda} \leq \log(R_0/\lambda)^{1/t} \leq \log n$. Hence, if $s \geq \log n \log\log n$ we achieve $O(\log n/n)$ learning rate. In the best case, where we have finitely many positive eigenvalues, then with a constant number of features we achieve $O(1/n)$ learning rate.

As for the plain sampling strategy, the learning rates and required numbers of features for the three above cases are: *i*) $O(1/\sqrt{n})$ and $s \geq \sqrt{n} \log n$ (polynomial decay), *ii*) $O(\log n/n)$ and $s \geq n$ (exponential decay), and *iii*) $O(1/n)$ and $s \geq n$ (finite many positive eigenvalues). Table 4.2 summarizes our results for the refined case.

Remark 4.3. *In Caponnetto and De Vito (2007), the convergence rate of the excess risk has been linked to two constants (b, c), where $b \in (1, \infty)$ represents the eigenvalue decay and $c \in [1, 2]$ measures the complexity of the target function $f_{\mathcal{H}}$. Essentially, c determines how fast the coefficients α_i of $f_{\mathcal{H}}$ decay, where α_i represents the coefficient of the expansion of $f_{\mathcal{H}}$ along the eigenfunctions of the integral operator defined by the kernel k and the data generating distribution $P(x, y)$. While $c = 1$ is equivalent to assuming $f_{\mathcal{H}}$ exists, in literature, it is typical that we have to assume benign cases (i.e., $c > 1$) to obtain fast learning rates.*

Our analysis is different from that in Caponnetto and De Vito (2007) in the sense that we only consider the worst case $c = 1$. Under this assumption, we compute excess learning risk of the RFFs estimator under various eigenvalue decays (the values of constant b). Our results demonstrate that even if we only consider case $c = 1$, we are still able to obtain the rate $O(1/\sqrt{n})$ in Theorem 1. This is aligned with the worst case rate in Caponnetto and De Vito (2007). In the refined analysis,

SAMPLING SCHEME	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
WEIGHTED RFFS	finite rank	$s \in \Omega(1)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega(\log n \cdot \log\log n)$	$O(\log n/n)$
	$\lambda_i \propto i^{-t}$ $(t > 1)$	$s \in \Omega(n^{1/2t} \cdot \log n)$	$O(1/\sqrt{n})$
PLAIN RFFS	finite rank	$s \in \Omega(n)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega(n)$	$O(\log n/n)$
	$\lambda_i \propto i^{-t}$ $(t > 1)$	$s \in \Omega(\sqrt{n} \cdot \log n)$	$O(1/\sqrt{n})$

Table 4.2: The refined case trade-offs between computational complexity and statistical efficiency for the squared error loss.

the local Rademacher complexity technique allows us to obtain sharper/better convergence rates without further assumptions on the constant such as $c > 1$, i.e., resulting in an improvement from $O(1/\sqrt{n})$ to $O(1/n)$ learning rate. Moreover, our fast rate range matches that in Caponnetto and De Vito (2007).

4.3.1.3 Comparison with the sharpest bounds from prior work

The trade-offs between the computational cost and prediction accuracy for RFFs in the squared error loss setting have been thoroughly studied in the literature (see, e.g., Avron et al., 2017; Rudi and Rosasco, 2017; Sriperumbudur and Szabó, 2015; Sutherland and Schneider, 2015). In the remainder of this section, we discuss our results relative to Rudi and Rosasco (2017), which provide the sharpest learning rates so far and demonstrate that it is possible to achieve computational savings without sacrificing the prediction accuracy when learning with RFFs.

Worst Case Analysis. We start with the worst case setting where the learning rate is *minimax optimal* $O(n^{-1/2})$. Table 4.3 provides a summary of our bounds for the plain RFFs sampling scheme relative to the results from Rudi and Rosasco (2017). Note that we omit the weighted sampling because Rudi and Rosasco (2017)

RESULTS	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
THIS WORK	finite rank	$s \in \Omega(\sqrt{n})$	$O(1/\sqrt{n})$
	$\lambda_i \propto A^i$	$s \in \Omega(\sqrt{n} \cdot \log\log n)$	
	$\lambda_i \propto i^{-2t}$ $(t \geq 1)$	$s \in \Omega(\sqrt{n} \cdot \log n)$	
	$\lambda_i \propto i^{-1}$		
RUDI & ROSASCO (2017)	finite rank	$s \in \Omega(\sqrt{n} \cdot \log n)$	$O(1/\sqrt{n})$
	$\lambda_i \propto A^i$		
	$\lambda_i \propto i^{-2t}$ $(t \geq 1)$		
	$\lambda_i \propto i^{-1}$		

Table 4.3: The comparison of our results to the sharpest learning rates from prior work (Rudi and Rosasco, 2017) in the worst case setting for plain RFFs and the squared error loss.

do not consider the worst case setting for leverage weighted RFFs. When the eigenspectrum has a finite rank or geometric decay, we can see that our results achieve sharper bound on the number of features, $s \in \Omega(\sqrt{n})$ versus $s \in \Omega(\sqrt{n} \cdot \log n)$ and $s \in \Omega(\sqrt{n} \cdot \log\log n)$ versus $s \in \Omega(\sqrt{n} \cdot \log n)$, respectively. When the eigenspectrum has a polynomial decay, we can see that the number of required features is the same.

Refined Analysis. We discuss here our results for the refined analysis, where learning algorithms with RFFs can achieve rates faster than $O(1/\sqrt{n})$. We split the comparison into two cases: plain and leverage weighted RFFs. Note that to obtain sharp learning rates, Rudi and Rosasco (2017) adopt the *source condition* assumption discussed in Remark 4.3. In particular, they assume that $c \in [1, 2]$. The assumption is convenient in that it allows one to derive a fast learning rate while at the same time providing a more flexible trade-off between the computational cost and prediction accuracy (please refer to Theorem 2 in Rudi and Rosasco (2017) for more details). To facilitate the comparison between our results and Rudi and

RESULTS	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
THIS WORK	finite rank	$s \in \Omega(n)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega(n)$	$O(\log n/n)$
	$\lambda_i \propto i^{-t}$ $(t > 1)$	$s \in \Omega(n^{1/2t} \cdot \log n)$	$O(1/\sqrt{n})$
RUDI & ROSASCO (2017)	finite rank	$s \in \Omega(n)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega(n)$	$O(\log n/n)$
	$\lambda_i \propto i^{-t}$ $(t > 1)$	$s \in \Omega(n^{\frac{2t}{1+2t}} \cdot \log n)$	$O(n^{-\frac{2t}{2t+1}})$

Table 4.4: The comparison of our results to the sharpest learning rates from prior work (Rudi and Rosasco, 2017) in the refined case for plain RFFs and the squared error loss.

RESULTS	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
THIS WORK	finite rank	$s \in \Omega(1)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega(\log n \cdot \log\log n)$	$O(\log n/n)$
	$\lambda_i \propto i^{-t}$ $(t > 1)$	$s \in \Omega(n^{1/2t} \cdot \log n)$	$O(1/\sqrt{n})$
RUDI & ROSASCO (2017)	finite rank	$s \in \Omega(1)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega((\frac{n}{\log n})^{\alpha} \cdot \log n), \alpha \in (0,1)$	$O(\log n/n)$
	$\lambda_i \propto i^{-t}$ $(t > 1)$	$s \in \Omega(n^{\frac{\alpha}{2t+1}} \cdot \log n), \alpha \in (0,1)$	$O(n^{-\frac{2t}{2t+1}})$

Table 4.5: The comparison of our results to the sharpest learning rates from prior work (Rudi and Rosasco, 2017) in the refined case for weighted RFFs and the squared error loss.

Rosasco (2017), we set $c = 1$ in both cases (plain and weighted random features). Table 4.4 provides a detailed comparison between the two works. One can see that when the eigenspectrum has a finite rank or exponential decay our results are the same as those in Rudi and Rosasco (2017). On the other hand, when eigenvalues have a fast polynomial decay i^{-t} with $t > 1$, it can be seen that the learning rates are different, i.e., $O(n^{-1/2})$ versus $O(n^{-2t/(2t+1)})$ obtained by Rudi and Rosasco (2017). However, the latter comes at the cost of requiring more features, i.e., $s \in \Omega(n^{2t/(2t+1)} \log n)$, whereas we only require $s \in \Omega(n^{1/2t} \log n)$.

We conclude this discussion with a comparison for the leverage weighted sampling scheme. To obtain fast learning rates in this setting, Rudi and Rosasco (2017) introduce a further *compatibility condition* which relates random features to the data generating distribution $P(x, y)$ through a parameter α. Table 4.5 below illustrates the trade-offs between the computational cost and the statistical learning accuracy in this case. We can see that when the eigenspectrum has a finite rank, our results are the same as those in Rudi and Rosasco (2017). However, when the eigenspectrum displays a slower decay the results are quite different. In particular, the computational cost and prediction accuracy trade-offs from Rudi and Rosasco (2017) depend heavily on the compatibility parameter α. Specifically, when the eigenspectrum exhibits an exponential decay, our results show that $s \in \Omega(\log n \log\log n)$ features is guaranteed to achieve the fast learning rate $O(\log n/n)$. In contrast to this, Rudi and Rosasco (2017) require $s \in \Omega((n/\log n)^{\alpha} \cdot \log n)$ features to achieve the same learning rate. When α is close to zero, the number of features required can be $\Omega(\log n)$, which is better than our results. However, when α is close to one, then the required number of features is $s \in \Omega(n)$, which is significantly worse than our results.

Additionally, when the eigenspectrum has a polynomial decay i^{-t} with $t > 1$, our results show that $s \in \Omega(n^{1/2t})$ features can guarantee the minimax optimal learning rate $O(1/\sqrt{n})$. For the same eigenspectrum decay, Rudi and Rosasco (2017) provide a more flexible trade-off that depends on α. For example, when $\alpha = 0$, then constant number of features is sufficient to obtain a fast learning rate $O(n^{-2t/(1+2t)})$. However, when $\alpha = 1$ then learning algorithms require $s \in \Omega(n^{1/(2t+1)})$ features to guarantee the same learning rate.

4.3.2 Learning with a Lipschitz Continuous Loss

We next consider kernel methods with Lipschitz continuous loss functions, examples of which include kernel support vector machines and kernel logistic regression. Similar to the squared error loss case, we approximate y_i with $g_\beta(x_i) = \mathbf{z}_{q,x_i}(\mathbf{v})^T\beta$ and formulate the following learning problem

$$\beta^\lambda := \underset{\beta\in\mathbb{R}^s}{\arg\min} \quad \frac{1}{n}\sum_{i=1}^{n} l(y_i, \mathbf{z}_{q,x_i}(\mathbf{v})^T\beta) + \lambda s\|\beta\|_2^2 \quad .$$

We let g_β^λ to be the prediction function defined based on β^λ and state an additional assumption that is specific to the Lipschitz continuous loss.

B.1 We assume that l is Lipschitz continuous with constant L,

$$(\forall g, g' \in \mathcal{H})(\forall x \in \mathcal{X})\colon |l_g - l_{g'}| \leq L|g(x) - g'(x)| \, .$$

Remark 4.4. *While the focus of our analysis in this section is on classification problems with a Lipschitz continuous loss function (e.g., hinge or logistic loss), the upper bounds on the excess risk and the resulting learning rates apply to the setting with* 0-1 *loss. The latter holds because the excess risk under* 0-1 *loss can be upper bounded by the excess risk under the hinge loss (see, e.g., Steinwart and Christmann, 2008; Sun et al., 2018, for more details).*

4.3.2.1 Worst Case Analysis

The following theorem describes the trade-off between the selected number of features s and the learning risk of the estimator, providing an insight into the choice of s for Lipschitz continuous loss.

Theorem 4.5. *Suppose that Assumption B.1 holds and that the conditions on sam-*

pling measure $\tilde{l}$ from Theorem 4.1 apply to the setting with a Lipschitz continuous loss. If

$$s \geq 5d_{\tilde{l}} \log \frac{16d_{\mathbf{K}}^{\lambda}}{\delta}$$

then for all $\delta \in (0,1)$, with probability $1-\delta$, the learning risk of g_{β}^{λ} can be upper bounded by

$$\mathbb{E}[l_{g_{\beta}^{\lambda}}] \leq \mathbb{E}[l_{g_{\mathcal{H}}}] + \sqrt{2\lambda} + O\left(\frac{1}{\sqrt{n}}\right). \tag{4.5}$$

Remark 4.6. *Just as Theorem 4.1 captures the trade-offs between computational complexity and statistical efficiency for the squared error loss, this theorem describes the relationship between s and $\mathbb{E}[l_{g_{\beta}^{\lambda}}]$ for the Lipschitz continuous setting. There is, however, an important difference between the two settings. More specifically, a property of the squared error loss function allows us to relate the RFF estimator $\tilde{f}_{\beta}^{\lambda}$ to the kernel ridge regression estimator $\hat{f}^{\lambda}$ by first computing the excess risk between $\tilde{f}_{\beta}^{\beta}$ and $\hat{f}^{\lambda}$ and then computing the same quantity for $\hat{f}^{\lambda}$ and $f_{\mathcal{H}}$. In contrast to this, the Lipschitz continuity property characteristic to the latter setting allows us to compute the excess learning risk directly by computing the risk difference between g_{β}^{λ} and $g_{\mathcal{H}}$. While the derivation in Lipschitz continuous loss is greatly simplified, the upper bound for the learning risk drops from $O(\lambda)$ in the regression case to $O(\sqrt{\lambda})$ in the classification case.*

Corollaries 4.3 and 4.4 provide bounds for the cases of leverage weighted and plain RFFs, respectively. The proofs are similar to the proofs of Corollaries 4.1 and 4.2.

Corollary 4.3. *If the probability density function from Theorem 4.5 is the empirical ridge leverage score distribution $q^{*}(v)$, then the upper bound on the risk from Eq. (4.5) holds for all $s \geq 5d_{\mathbf{K}}^{\lambda} \log \frac{16d_{\mathbf{K}}^{\lambda}}{\delta}$.*

Similar to Theorem 4.1, we consider four different cases for the effective dimension

of the problem $d_{\mathbf{K}}^{\lambda}$. Corollary 4.3 states that the statistical efficiency is preserved if the leverage weighted RFFs strategy is used with $s = \Omega(1)$, $s \geq \log n \log\log n$, $s \geq n^{1/(2t)} \log n$, and $s \geq n \log n$, respectively. Again, significant computational savings can be achieved if the kernel matrix $\mathbf{K}$ has a finite rank, as well as geometrically/exponentially or polynomially decaying eigenvalues.

Corollary 4.4. *If the probability density function from Theorem 4.5 is the spectral measure $p(v)$ from Eq. (2.9), then the upper bound on the risk from Eq. (4.5) holds for all $s \geq 5\frac{z_0^2}{\lambda} \log \frac{(16 d_{\mathbf{K}}^{\lambda})}{\delta}$.*

Corollary 4.4 states that $n \log n$ features are required to attain $O(n^{-1/2})$ convergence rate of the learning risk with plain RFFs, recovering results from Rahimi and Recht (2009). Similar to the analysis in the squared error loss case, Theorem 4.5 together with Corollaries 4.3 and 4.4 allows theoretically motivated trade-offs between the statistical and computational efficiency of the estimator g_{β}^{λ}. Table 4.6 summarizes the worst case trade-offs between the computational and statistical efficiency.

4.3.2.2 Refined Analysis

In general, it is hard for classification problems to achieve learning rates sharper/-faster than $O\left(1/\sqrt{n}\right)$. However, as pointed out by Bartlett et al. (2006) and Steinwart and Christmann (2008), under some benign conditions, it is possible to obtain $O(1/n)$ convergence rate. Hence, in this section, by adding an extra assumption, we derive a sharp learning rate for classification problems under RFFs setting. Similar to the squared error loss case, we rely on the notion of local Rademacher complexity to derive such a learning rate (details of the proof are presented in Section 4.6.5). Before we formally specify our result, we state the required additional assumption.

SAMPLING SCHEME	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
WEIGHTED RFFS	finite rank	$s \in \Omega(1)$	$O(1/\sqrt{n})$
	$\lambda_i \propto A^i$	$s \in \Omega(\log n \cdot \log\log n)$	
	$\lambda_i \propto i^{-2t}, t \geq 1$	$s \in \Omega(n^{1/2t} \cdot \log n)$	
	$\lambda_i \propto i^{-1}$	$s \in \Omega(n \cdot \log n)$	
PLAIN RFFS	finite rank	$s \in \Omega(n)$	$O(1/\sqrt{n})$
	$\lambda_i \propto A^i$	$s \in \Omega(n \cdot \log\log n)$	
	$\lambda_i \propto i^{-2t}, t \geq 1$	$s \in \Omega(n \cdot \log n)$	
	$\lambda_i \propto i^{-1}$		

Table 4.6: The worst case trade-offs between computational and statistical efficiency for Lipschitz continuous loss.

B.2 Recall that $g_{\mathcal{H}}$ is the estimator such that $g_{\mathcal{H}} = \arg\inf_{g \in \mathcal{H}} \mathbb{E}[l_g]$, where P is a probability distribution over $\mathcal{X} \times \mathcal{Y}$. We assume that there exists a constant B such that for all $g \in \mathcal{H}$

$$\mathbb{E}[(g - g_{\mathcal{H}})^2] \leq B\mathbb{E}[l_g - l_{g_{\mathcal{H}}}] \, .$$

Assumption B.2 is a condition for classification problems to obtain faster learning rates. It typically requires that the function space $\mathcal{H}$ is convex and uniformly bounded, as well as an additional uniform convexity condition on the loss function l. It can be shown that many loss functions satisfy this assumption, including squared loss (Bartlett et al., 2005) and hinge loss (Steinwart and Christmann, 2008, Chapter 8.5). Additional examples of such loss functions are discussed in Bartlett et al. (2006) and Mendelson (2002). As l is Lipschitz continuous, we have

$$\mathbb{E}[(l_g - l_{g_{\mathcal{H}}})^2] \leq L^2\mathbb{E}[(g - g_{\mathcal{H}})^2] \leq BL^2\mathbb{E}[l_g - l_{g_{\mathcal{H}}}] \, .$$

This is the variance condition described in Steinwart and Christmann (2008, Chapter 7.3), required to achieve faster convergence rates. The variance condition is also

linked to the Massart's low noise condition or more generally to the Tsybakov condition (Sun et al., 2018), which intuitively speaking, requires that $P(Y = 1 \mid X = x)$ is not close to $1/2$. For more details, please refer to Tsybakov et al. (2004) and Koltchinskii (2011).

Theorem 4.7. *Suppose that Assumptions B.1-2 hold and that the conditions on sampling measure $\tilde{l}$ from Theorem 4.1 apply to the setting with a Lipschitz continuous loss. If*

$$s \geq 5d_{\tilde{l}} \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta}$$

then for all $D > 1$ and $\delta \in (0, 1)$ with probability greater than $1 - \delta$, we have

$$\mathbb{E}[l_{g_\beta^\lambda}] \leq \frac{12D}{B} \hat{r}_{\mathcal{H}}^{*} + \frac{D}{D-1} \sqrt{2\lambda} + O(1/n) + \mathbb{E}[l_{g_{\mathcal{H}}}] \,. \tag{4.6}$$

Furthermore, denoting the eigenvalues of the normalized kernel matrix $(1/n)\mathbf{K}$ with $\{\hat{\lambda}_i\}_{i=1}^n$, we have that $\hat{r}_{\mathcal{H}}^{*}$ can be upper bounded by

$$\hat{r}_{\mathcal{H}}^{*} \leq \min_{0 \leq h \leq n} \left(b_0 \frac{h}{n} + \sqrt{\frac{1}{n} \sum_{i>h} \hat{\lambda}_i} \right) \,, \tag{4.7}$$

where B and b_0 are some constants.

Theorem 4.7 provides a sharper learning rate compared to Theorem 4.5. Similar to Theorem 4.2, $\hat{r}_{\mathcal{H}}^{*}$ can be upper bounded by $O(1/n)$ (Gram-matrix is of finite rank), $O(\log n/n)$ (eigenvalues decay exponentially), and $O(1/\sqrt{n})$ (eigenvalues decay proportional to $1/n$). This has various implications on the trade-offs between computational cost and statistical efficiency. Just as in previous sections, we split the discussion into two parts according to the two sampling strategies.

We first discuss the scenario with empirical leverage score sampling. In a finite rank setting, if we choose $\lambda \in O(1/n^2)$, we can see that the learning rate is of the

order $O(1/n)$. In addition, since we use the weighted sampling strategy and the Gram-matrix has finitely many eigenvalues, RFFs learning only requires a constant number of features to achieve $O(1/n)$ learning rate. To the best of our knowledge, this is the first result that achieves this.

In the case of exponential spectrum decay, the learning rate can be bounded with $\hat{r}^*_{\mathcal{H}} \leq \frac{\log n}{n}$ by setting $\lambda \in O(\frac{\log^2 n}{n^2})$. The number of required features is $s \geq \log n \log\log n$ because $d^{\lambda}_{\mathbf{K}} \leq \log(R^2/\lambda) \leq \log n$.

If the eigenvalues decay at the rate $\lambda_i \propto O(i^{-t})$ with $t > 1$, then the learning rate is $O(1/\sqrt{n})$ by setting $\lambda \in O(1/n)$, with the requirement on the number of features given by $d^{\lambda}_{\mathbf{K}} \leq (R^2/\lambda)^{1/t} \leq n^{1/t}$. Since $t > 1$, one can see that with fewer than n features, we could obtain fast $O(1/n)$ learning rate.

On the other hand, in the plain sampling scheme, if we would like to achieve the fast $O(1/n)$ learning rate, we need to set $\lambda \in O(1/n^2)$, implying that the required number of features has to be $s \geq n^2$. This is undesirable as it does not provide any computation savings. The bottleneck here is that in the Lipschitz continuous case, learning rate is upper bounded by $O(\sqrt{\lambda})$.

4.3.2.3 Comparison with the sharpest bounds from prior work

In the setting with Lipshcitz continuous loss, several previous results provide similar trade-offs between the computational cost and prediction accuracy (see, e.g., Bach, 2017b; Rahimi and Recht, 2009; Sun et al., 2018). We cover below the sharpest bounds from prior work and discuss how our results advance the understanding of learning with random features in this setting.

Worst Case Analysis. Rahimi and Recht (2009) were the first to consider the trade-offs between the computational cost and prediction accuracy for the plain RFFs sampling scheme. Building on that work, Bach (2017b) provide a characteri-

SAMPLING SCHEME	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
WEIGHTED RFFS	finite rank	$s \in \Omega(1)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega(\log n \cdot \log\log n)$	$O(\log n/n)$
	$\lambda_i \propto i^{-t}, t > 1$	$s \in \Omega(n^{1/2t} \cdot \log n)$	$O(1/\sqrt{n})$
PLAIN RFFS	finite rank	$s \in \Omega(n^2)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega(n^2)$	$O(\log n/n)$
	$\lambda_i \propto i^{-t}, t > 1$	$s \in \Omega(n \cdot \log n)$	$O(1/\sqrt{n})$

Table 4.7: The refined case trade-offs between computational and statistical efficiency for Lipschitz continuous loss.

zation for the setting with leverage weighted RFFs.

We start with a comparison to Rahimi and Recht (2009) in the setting with plain RFFs. The first block of rows in Table 4.8 illustrates the difference between our results and that work. We can see that under plain sampling, when the eigenspetrum has a finite rank or exhibits an exponential decay, the required number of features in our bounds is smaller than that required by Rahimi and Recht (2009), i.e., $s \in \Omega(n)$ versus $s \in \Omega(n \cdot \log n)$ and $s \in \Omega(n \cdot \log\log n)$ versus $s \in \Omega(n \cdot \log n)$, respectively. When the eigenspectum follows a polynomial decay, our results match those provided by Rahimi and Recht (2009).

For the worst case setting under the leverage weighted sampling scheme, Bach (2017b) provides a detailed theoretical analysis of learning with random features by establishing the equivalence between random features and the kernel quadrature rules. That work was also the first to demonstrate that it is possible to achieve computational savings while preserving the prediction accuracy. The second block of rows in Table 4.8 illustrates the difference between our results and Bach (2017b). We can see that apart from the finite rank case that was not covered by Bach (2017b), our results match the worst case bounds.

SAMPLING SCHEME	RESULTS	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
PLAIN RFFS	THIS WORK	finite rank	$s \in \Omega(n)$	$O(1/\sqrt{n})$
		$\lambda_i \propto A^i$	$s \in \Omega(n \cdot \log\log n)$	
		$\lambda_i \propto i^{-2t}, t \geq 1$	$s \in \Omega(n \cdot \log n)$	
		$\lambda_i \propto i^{-1}$		
	RAHIMI & RECHT (2009)	finite rank	$s \in \Omega(n \cdot \log n)$	
		$\lambda_i \propto A^i$		
		$\lambda_i \propto i^{-2t}, t \geq 1$		
		$\lambda_i \propto i^{-1}$		
WEIGHTED RFFS	THIS WORK	finite rank	$s \in \Omega(1)$	$O(1/\sqrt{n})$
		$\lambda_i \propto A^i$	$s \in \Omega(\log n \cdot \log\log n)$	
		$\lambda_i \propto i^{-2t}, t \geq 1$	$s \in \Omega(n^{1/2t} \cdot \log n)$	
		$\lambda_i \propto i^{-1}$	$s \in \Omega(n \cdot \log n)$	
	BACH (2017B)	finite rank	–	
		$\lambda_i \propto A^i$	$s \in \Omega(\log n \cdot \log\log n)$	
		$\lambda_i \propto i^{-2t}, t \geq 1$	$s \in \Omega(n^{1/2t} \cdot \log n)$	
		$\lambda_i \propto i^{-1}$	$s \in \Omega(n \cdot \log n)$	

Table 4.8: The comparison of our results to the sharpest learning rates from prior work in the worst case setting with a Lipschitz continuous loss and plain/weighted RFFs.

Refined Case Analysis. Having covered the worst case setting for Lipschitz continuous loss, we now proceed to the refined case where learning rates can be sharper than $O(1/\sqrt{n})$. We remark that while Rahimi and Recht (2009) and Bach (2017b) do not cover the refined case with fast learning rates, Sun et al. (2018) provide a refined analysis for the hinge loss and 0-1 loss. Similar to our work (i.e., Assumption B.2), Sun et al. (2018) assume a low noise condition and demonstrate that learning algorithms operating on features selected via an optimized sampling distribution can obtain sharper learning rates in classification problems than the standard $O(1/\sqrt{n})$ rate.

Table 4.9 illustrates the differences between our results and bounds from Sun et al. (2018). From the table, we can see that the guarantees match when the eigenspectrum has a finite rank. When the eigenspectrum exhibits a slower decay,

SAMPLING SCHEME	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
THIS WORK	finite rank	$s \in \Omega(1)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega(\log n \cdot \log\log n)$	$O(\log n/n)$
	$\lambda_i \propto i^{-t}, t > 1$	$s \in \Omega(n^{1/2t} \cdot \log n)$	$O(1/\sqrt{n})$
SUN ET AL. (2018)	finite rank	$s \in \Omega(1)$	$O(1/n)$
	$\lambda_i \propto A^i$	$s \in \Omega(\log^d n \cdot \log\log^d n)$	$O(\log^{d+2} n/n)$
	$\lambda_i \propto i^{-t}, t > 1$	$s \in \Omega(n^{\frac{2}{2+t}} \cdot \log n)$	$O(n^{\frac{t}{2+t}})$

Table 4.9: The refined case trade-offs between computational and statistical efficiency relative to Sun et al. (2018).

however, the results differ significantly. More specifically, the results from Sun et al. (2018) suffer from the curse of dimension when the decay is exponential. This is because both the number of features required and the learning rate obtained depend on the data dimension d, whereas our results do not have this dependency.

When the eigenspectrum exhibits a polynomial decay, we can see that our results achieve the minimax learning rate of $O(1/\sqrt{n})$ while Sun et al. (2018) obtain a more flexible rate that depends on the magnitude of the decay given by parameter t. When $t < 2$, our results have a better trade-off as both the number of features and learning rate are sharper than those from Sun et al. (2018). When $t > 2$, the learning rate obtained by Sun et al. (2018) is faster than ours. However, that comes at the cost of increasing the required number of features, i.e., $s \in \Omega(n^{\frac{2}{2+t}} \cdot \log n)$ versus $s \in \Omega(n^{\frac{1}{2t}} \cdot \log n)$. In particular, setting $t = 4$ we can see that Sun et al. (2018) obtain a fast $O(n^{2/3})$ learning rate, but at the cost of requiring $s \in \Omega(n^{1/3} \cdot \log n)$ random features. For the same setting, on the other hand, we obtain the minimax optimal $O(1/\sqrt{n})$ learning rate with only $s \in \Omega(n^{1/8} \cdot \log n)$ random features.

Algorithm 3 APPROXIMATE LEVERAGE WEIGHTED RFFS

Input: sample of examples $\{(x_i, y_i)\}_{i=1}^n$, shift-invariant kernel function k, and regularization parameter λ

Output: set of features $\{(v_1, p_1), \cdots, (v_m, p_m)\}$ with m and each p_i computed as in lines 3–4

1: sample s features $\{v_1, \ldots, v_s\}$ from $p(v)$

2: create a feature matrix $\mathbf{Z}_s$ such that the ith row of $\mathbf{Z}_s$ is

$$[z(v_1, x_i), \cdots, z(v_s, x_i)]^T$$

3: associate with each feature v_i a real number p_i such that p_i is equal to the ith diagonal element of the matrix

$$\mathbf{Z}_s^T \mathbf{Z}_s ((1/s)\mathbf{Z}_s^T \mathbf{Z}_s + n\lambda I)^{-1}$$

4: $m \leftarrow \sum_{i=1}^s p_i$ and $M \leftarrow \{(v_i, p_i/m)\}_{i=1}^s$

5: sample m features from M using the multinomial distribution given by the vector $(p_1/m, \cdots, p_s/m)$

4.4 A Fast Approximation of Leverage Weighted RFFs

As discussed in Sections 4.3, sampling according to the empirical ridge leverage score distribution (i.e., leverage weighted RFFs) could speed up kernel methods. However, computing ridge leverage scores is as costly as inverting the Gram-matrix. To address this computational shortcoming, we propose a simple algorithm to approximate the empirical ridge leverage score distribution and the leverage weights.

In particular, we propose to first sample a pool of s features from the spectral measure $p(\cdot)$ and form the feature matrix $\mathbf{Z}_s \in \mathbb{R}^{n \times s}$ (Algorithm 3, lines 1-2). Then, the algorithm associates an approximate empirical ridge leverage score to each feature (Algorithm 3, lines 3-4) and samples a set of $m \ll s$ features from the pool proportional to the computed scores (Algorithm 3, line 5). The output of the algorithm can be compactly represented via the feature matrix $\mathbf{Z}_m \in \mathbb{R}^{n \times m}$ such that the i-th row of $\mathbf{Z}_m$ is given by $\mathbf{z}_{x_i}(\mathbf{v}) = [\sqrt{\frac{m}{p_1}} z(v_1, x_i), \cdots, \sqrt{\frac{m}{p_m}} z(v_m, x_i)]^T$.

The computational cost of Algorithm 3 is dominated by the operations in step 3.

As $\mathbf{Z}_s \in \mathbb{R}^{n\times s}$, the multiplication of matrices $\mathbf{Z}_s^T\mathbf{Z}_s$ costs $O(ns^2)$ and inverting $\mathbf{Z}_s^T\mathbf{Z}_s + n\lambda I$ costs only $O(s^3)$. Hence, for $s \ll n$, the overall runtime is only $O(ns^2 + s^3)$. Moreover, $\mathbf{Z}_s^T\mathbf{Z}_s = \sum_{i=1}^n \mathbf{z}_{x_i}(\mathbf{v})\mathbf{z}_{x_i}(\mathbf{v})^T$ and it is possible to store only the rank-one matrix $\mathbf{z}_{x_i}(\mathbf{v})\mathbf{z}_{x_i}(\mathbf{v})^T$ into the memory. Thus, the algorithm only requires to store an $s \times s$ matrix and can avoid storing $\mathbf{Z}_s$, which would incur a storage cost of $O(n \times s)$.

The following theorem gives the convergence rate for the learning risk of Algorithm 3 in the kernel ridge regression setting.

Theorem 4.8. *Suppose that Assumption A.1 holds and consider the regression problem defined with a shift-invariant kernel k, a sample of examples $\{(x_i, y_i)\}_{i=1}^n$, and a regularization parameter λ. Let s be the number of RFFs in the pool of features from Algorithm 3, sampled using the spectral measure $p(\cdot)$ from Eq. (2.9) and the regularization parameter λ. Denote with $\tilde{f}_m^{\lambda^*}$ the ridge regression estimator obtained using a regularization parameter λ^* and a set of RFFs $\{v_i\}_{i=1}^m$ returned by Algorithm 3. If*

$$s \;\geq\; \frac{7z_0^2}{\lambda}\log\frac{(16d_{\mathbf{K}}^{\lambda})}{\delta} \quad \textit{and} \quad m \;\geq\; 5d_{\mathbf{K}}^{\lambda^*}\log\frac{(16d_{\mathbf{K}}^{\lambda^*})}{\delta}\,,$$

then for all $\delta \in (0,1)$, with probability $1-\delta$, the learning risk of $\tilde{f}_m^{\lambda^*}$ can be upper bounded by

$$\mathbb{E}[l_{\tilde{f}_m^{\lambda^*}}] \;\leq\; \mathbb{E}[l_{f_{\mathcal{H}}}] + 4\lambda + 4\lambda^* + O\left(\frac{1}{\sqrt{n}}\right).$$

Moreover, this upper bound holds for $m \in \Omega(\frac{s}{n\lambda})$.

Theorem 4.8 bounds the learning risk of the ridge regression estimator over random features generated by Algorithm 3. We can now observe that using the minimax choice of the regularization parameter for kernel ridge regression $\lambda, \lambda^* \propto n^{-1/2}$, the

number of features that Algorithm 3 needs to sample from the spectral measure of the kernel k is $s \in \Omega(\sqrt{n}\log n)$. Then, the ridge regression estimator $\tilde{f}_m^{\lambda^*}$ converges with the minimax rate to the hypothesis $f_{\mathcal{H}} \in \mathcal{H}$ for $m \in \Omega(\log n \cdot \log\log n)$.

This is a significant improvement compared to the spectral measure sampling (plain RFFs), which requires $\Omega(n^{3/2})$ features for in-sample training and $\Omega(\sqrt{n}\log n)$ for out-of-sample test predictions.

Theorem 4.9 provides a convergence bound for kernel support vector machines and logistic regression. Compared to the previous result, the convergence rate of the learning risk, however, is at a slower $O(\sqrt{\lambda} + \sqrt{\lambda^*})$ rate due to the difference in the employed loss function (see also Section 4.3.2).

Theorem 4.9. *Suppose that Assumption B.1 holds and consider the learning problem with a Lipschitz continuous loss function, a shift-invariant kernel k, a sample of examples $\{(x_i, y_i)\}_{i=1}^n$, and a regularization parameter λ. Let s be the number of RFFs in the pool of features from Algorithm 3, sampled using the spectral measure $p(\cdot)$ from Eq. (2.9) and the regularization parameter λ. Denote with $\tilde{g}_m^{\lambda^*}$ the estimator obtained using a regularization parameter λ^* and a set of RFFs $\{v_i\}_{i=1}^m$ returned by Algorithm 3. If*

$$s \geq \frac{5z_0^2}{\lambda}\log\frac{(16 d_{\mathbf{K}}^{\lambda})}{\delta} \quad \text{and} \quad m \geq 5 d_{\mathbf{K}}^{\lambda^*}\log\frac{(16 d_{\mathbf{K}}^{\lambda^*})}{\delta},$$

then for all $\delta \in (0,1)$, with probability $1-\delta$, the learning risk of $\tilde{g}_m^{\lambda^*}$ can be upper bounded by

$$\mathbb{E}[l_{\tilde{g}_m^{\lambda^*}}] \leq \mathbb{E}[l_{g_{\mathcal{H}}}] + \sqrt{2\lambda} + \sqrt{2\lambda^*} + O\left(\frac{1}{\sqrt{n}}\right).$$

We conclude by pointing out that the proposed algorithm provides an interesting new trade-off between the computational cost and prediction accuracy. In particular,

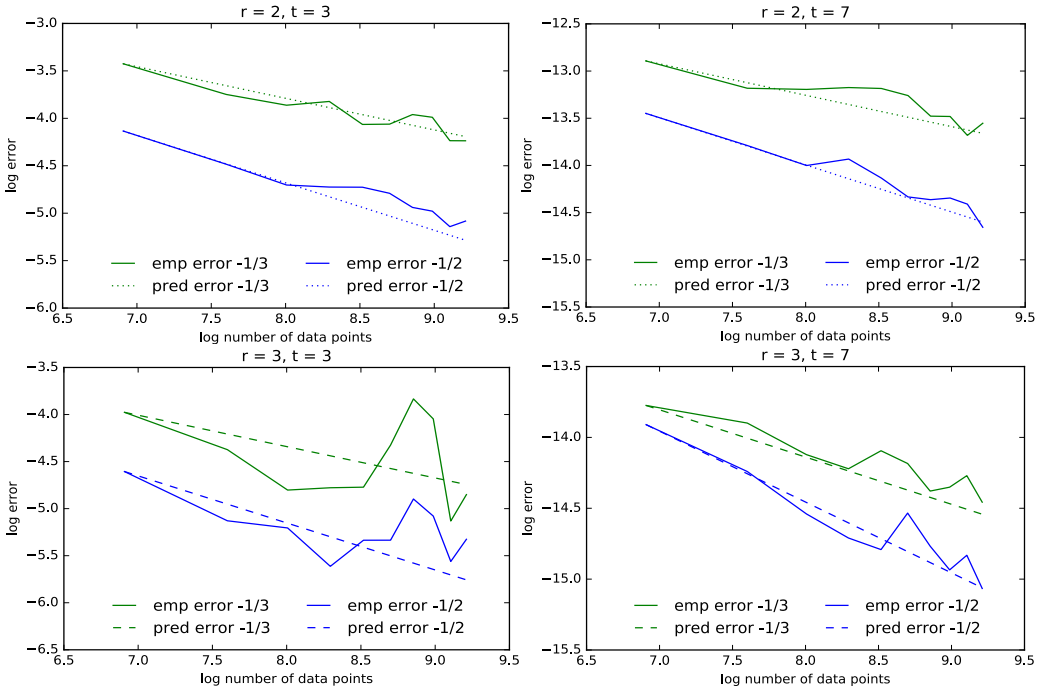

Figure 4.1: The log-log plot of the theoretical and simulated risk convergence rates, averaged over 100 repetitions.

one can pay an upfront cost (same as plain RFFs) to compute the leverage scores, re-sample significantly fewer features and employ them in the training, cross-validation, and prediction stages. This can reduce the computational cost for predictions at test points from $\Omega(\sqrt{n}\log n)$ to $\Omega(\log n \cdot \log\log n)$. Moreover, in the case where the amount of features with approximated leverage scores utilized is the same as in plain RFFs, the prediction accuracy would be significantly improved, as demonstrated in our experiments. We provide a comparison between our proposed algorithm and the original RFFs algorithm in Table 4.10.

ALGORITHMS	TRAINING TIME	MEMORY	TESTING TIME
OUR ALGORITHM	$O(n^2)$	$O(n)$	$O(\log\log n)$
PLAIN RFFS	$O(n^2)$	$O(n\sqrt{n})$	$O(\sqrt{n})$

Table 4.10: The computation time of our proposed algorithm and the plain RFFs.

4.5 Numerical Experiments

In this section, we report the results of our numerical experiments (on both simulated and real-world datasets) aimed at validating the theoretical results and demonstrating the utility of Algorithm 3. In the first experiment, the goal is to show that our bounds for the ridge regression setting are tight by demonstrating empirically that the observed error rates follow closely the provided learning risk bounds. The second experiment deals with the effectiveness of the proposed algorithm relative to the plain RFFs sampling scheme, evaluated on four benchmark datasets typically used for this type of problems. The third and final experiment aims at demonstrating the utility of the leverage weighted features in a simulated experiment designed such that an effective approximation of the target hypothesis requires a small number of random features which are located in the tails of the spectral measure corresponding to the selected shift-invariant kernel.

We use a simulated experiment to verify the sharpness of our theoretical results. More specifically, we consider a spline kernel of order r where $k_{2r}(x,y) = 1 + \sum_{m>0} \frac{1}{m^{2r}} \cos 2\pi m(x-y)$ (also considered by Bach, 2017b; Rudi and Rosasco, 2017). If the marginal distribution of X is uniform on $[0,1]$, one can show that $k_{2r}(x,y) = \int_0^1 z(v,x)z(v,y)p(v)dv$, where $z(v,x) = k_r(v,x)$ and $p(v)$ is uniform on $[0,1]$. Moreover, one can show that the optimal weighted sampling distribution $q^*(v)$ is the same as $p(v)$, which allows us to use weighted RFFs sampling strategy. We let y be a Gaussian random variable with mean $f(x) = k_t(x, x_0)$ (for some $x_0 \in [0,1]$) and variance σ^2. We sample features according to $q^*(v)$ to estimate f and compute the excess risk. By Theorem 4.1 and Corollary 4.1, if the number of features is proportional to $d_{\mathbf{K}}^{\lambda}$ and $\lambda \propto n^{-1/2}$, we should expect the excess risk converging at $O(n^{-1/2})$, or at $O(n^{-1/3})$ if $\lambda \propto n^{-1/3}$. Figure 4.1 demonstrates that this is indeed the case for different values of r and t.

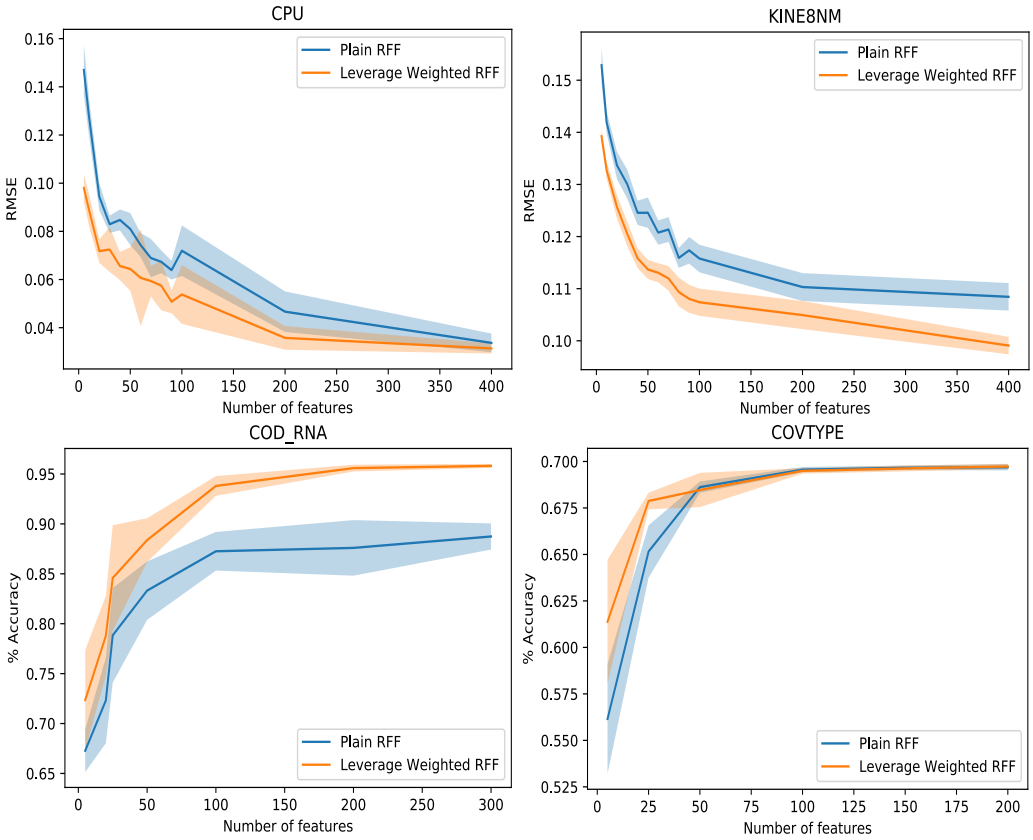

Figure 4.2: Comparison of leverage weighted and plain RFFs, with weights computed according to Algorithm 3.

Next, we make a comparison between the performances of leverage weighted (computed according to Algorithm 3) and plain RFFs on real-world datasets. In particular, we use four datasets from Chang and Lin (2011) and Dheeru and Karra Taniskidou (2017) for this purpose, including two for regression and two for classification: `CPU`, `KINEMATICS`, `COD-RNA` and `COVTYPE`. Apart from `KINEMATICS`, the other three datasets were used in Yang et al. (2012) to investigate the difference between the Nyström method and plain RFFs. We use the ridge regression and SVM package from Pedregosa et al. (2011) as a solver to perform our experiments. We evaluate the regression tasks using the root mean squared error and the classification ones using the average percentage of misclassified examples. The Gaussian/RBF kernel is used for all the datasets with hyper-parameter

# OF FEATURES	PLAIN RFFS	LEVERAGE WEIGHTED RFFS
$1,000$	0.13 ± 0.06	0.04 ± 0.01
$50,000$	0.04 ± 0.02	–

Table 4.11: The table summarizes our experiment on the synthetic dataset and illustrates the effectiveness of the proposed algorithm (i.e., leverage weighted RFFs) relative to plain RFFs sampling. The reported numbers are the root mean squared error (RMSE) along with a corresponding confidence interval.

tuning via 5-fold inner cross validation. We have repeated all the experiments 10 times and reported the average test error for each dataset. Figure 4.2 compares the performances of leverage weighted and plain RFFs. In regression tasks, we observe that the upper bound of the confidence interval for the root mean squared error corresponding to leverage weighted RFF is below the lower bound of the confidence interval for the error corresponding to plain RFFs. Similarly, the lower bound of the confidence interval for the classification accuracy of leverage weighted RFFs is (most of the time) higher than the upper bound on the confidence interval for plain RFFs. This indicates that leverage weighted RFFs perform statistically significantly better than plain RFFs in terms of the learning accuracy and prediction error.

In the final experiment, we would like to show that the proposed algorithm can significantly reduce the number of required features without loss of statistical efficiency. As it is challenging to find an appropriate real-world dataset for this illustration, we design a synthetic regression problem on our own. The main idea is to define a target regression function as a linear model directly in the space of RFFs. We select the target features such that they are in the tail of the spectral measure of the kernel that defines our hypothesis space. This ensures that plain RFFs strategy will require a large number of features to describe the target hypothesis. We evaluate the effectiveness of our algorithm relative to plain RFFs and construct the described synthetic dataset as follows: we first generate

samples w^* from a multimodal Gaussian distribution where the modes are at $(-2,-2),(-2,2),(2,-2)$, and $(2,2)$. Moreover, each of the modes has a diagonal covariance matrix of 0.5. These samples are going to be our frequencies for a RFFs mapping. Next, we sample our covariates x from $\mathcal{N}(0, 5 * I)$. In order to generate our response variables, we map the covariates x through a RFFs map where the frequencies are given by samples w^*. We then randomly sample regression weights α_r from $\mathcal{N}(0,1)$. Hence, the data generating process can be specified by:

$$y = \alpha_r^T \phi_{w^*}(x) + \epsilon \,,$$

where $\epsilon \sim \mathcal{N}(0,\sigma)$ and ϕ_{w^*} is a RFFs map with w^* as the frequencies. By setting up our data generating process in this way, we are able to systematically investigate how well the proposed algorithm works. In particular, we consider learning the above described hypothesis using RFFs that correspond to a Gaussian kernel. Such a kernel corresponds to a uni-modal Gaussian distribution in the frequency domain and we will show that leverage weighted sampling is capable of selecting a subset of plain RFFs sampled from that distribution, which covers the modes of the multimodal distribution that characterizes the data generating process.

In our simulations, we have opted for the following setting: $50,000$ data points denoted with x in the data-generating process, 400 features/frequencies w^* that define the target hypothesis y, and the additive noise variance parameter $\sigma = 0.1$. We then run plain RFFs as well as our leverage weighted RFFs on this dataset. For plain RFFs, we run the experiments with $1,000$ and $50,000$ frequencies/features, while with our leverage weighted RFFs we only use $1,000$ features which have been selected from a pool consisting of $10,000$ plain random RFFs (i.e., a sub-sample from the original $50,000$ features). We carefully cross-validate both methods across a grid of hyper-parameters and report the results in Table 4.11. The results

confirm our theoretical findings and illustrate that learning with $1,000$ leverage weighted RFFs is as effective as learning with a complete pool of $50,000$ plain RFFs.

4.6 Proofs

4.6.1 Proof of Proposition 4.1

Proposition 4.1. *Assume that the RKHS $\mathcal{H}$ with kernel k admits a decomposition as in Eq. (2.9) and denote by $\tilde{\mathcal{H}} \coloneqq \{\tilde{f} \mid \tilde{f} = \sum_{i=1}^{s} \alpha_i z(v_i, \cdot), \alpha_i \in \mathbb{R}\}$ the RKHS with kernel $\tilde{k}$ (see Eq. 2.10). Then, for all $\tilde{f} \in \tilde{\mathcal{H}}$ it holds that $\|\tilde{f}\|_{\tilde{\mathcal{H}}}^2 \leq s\|\alpha\|_2^2$.*

Proof. Let us define a space of functions as

$$\mathcal{H}_1 \coloneqq \{f \mid f(x) = \alpha z(v, x), \alpha \in \mathbb{R}\} .$$

We now show that $\mathcal{H}_1$ is a RKHS with kernel defined as $k_1(x, y) = (1/s)z(v, x)z(v, y)$, where s is a constant. Define a map $M : \mathbb{R} \to \mathcal{H}_1$ such that $M\alpha = \alpha z(v, \cdot), \forall \alpha \in \mathbb{R}$. The map M is a bijection, i.e. for any $f \in \mathcal{H}_1$ there exists a unique $\alpha_f \in \mathbb{R}$ such that $M^{-1}f = \alpha_f$. Now, we define an inner product on $\mathcal{H}_1$ as

$$\langle f, g\rangle_{\mathcal{H}_1} = \langle \sqrt{s}M^{-1}f, \sqrt{s}M^{-1}g\rangle_{\mathbb{R}} = s\alpha_f\alpha_g .$$

It is easy to show that this is a well defined inner product and, thus, $\mathcal{H}_1$ is a Hilbert space.

For any instance y, $k_1(\cdot, y) = (1/s)z(v, \cdot)z(v, y) \in \mathcal{H}_1$, since $(1/s)z(v, y) \in \mathbb{R}$

by definition. Take any $f \in \mathcal{H}_1$ and observe that

$$\begin{aligned}\langle f, k_1(\cdot, y)\rangle_{\mathcal{H}_1} &= \langle \sqrt{s}M^{-1}f, \sqrt{s}M^{-1}k_1(\cdot, y)\rangle_{\mathbb{R}} \\ &= s\langle \alpha_f, 1/sz(v, y)\rangle_{\mathbb{R}} \\ &= \alpha_f z(v, y) = f(y)\ .\end{aligned}$$

Hence, we have demonstrated the reproducing property for $\mathcal{H}_1$ and $\|f\|^2_{\mathcal{H}_1} = s\alpha_f^2$.

Now, suppose we have a sample of features $\{v_i\}_{i=1}^s$. For each v_i, we define the RKHS

$$\mathcal{H}_i \coloneqq \{f \mid f(x) = \alpha z(v_i, x), \alpha \in \mathbb{R}\}$$

with the kernel $k_i(x, y) = (1/s)z(v_i, x)z(v_i, y)$. Denoting with

$$\tilde{\mathcal{H}} = \oplus_{i=1}^s \mathcal{H}_i = \{\tilde{f} : \tilde{f} = \sum_{i=1}^s f_i, f_i \in \mathcal{H}_i\}$$

and using the fact that the direct sum of RKHSs is another RKHS (Berlinet and Thomas-Agnan, 2011), we have that $\tilde{k}(x, y) = \sum_{i=1}^s k_i(x, y) = (1/s)\sum_{i=1}^s z(v_i, x)z(v_i, y)$ is the kernel of $\tilde{\mathcal{H}}$ and that the squared norm of $\tilde{f} \in \tilde{\mathcal{H}}$ is defined as

$$\min_{f_i \in \mathcal{H}_i \mid \tilde{f} = \sum_{i=1}^s f_i} \sum_{i=1}^s \|f_i\|^2_{\mathcal{H}_i} =$$

$$\min_{\alpha_i \in \mathbb{R} \mid \tilde{f} = \sum_{i=1}^s \alpha_i z(v_i, \cdot)} \sum_{i=1}^s s\alpha_i^2 = \min_{\alpha_i \in \mathbb{R} \mid \tilde{f} = \sum_{i=1}^s \alpha_i z(v_i, \cdot)} s\|\alpha\|_2^2\ .$$

Hence, we have that $\|\tilde{f}\|^2_{\tilde{\mathcal{H}}} \leq s\|\alpha\|_2^2$. □

4.6.2 Proof of Theorem 4.1

To prove Theorem 4.1, we need two auxiliary results formulated in Lemmas 4.6 and 4.7 (the proofs are provided in Appendices 4.8.2 and 4.8.3, respectively). More specifically, Lemma 4.6 is a general result that gives an upper bound on the approximation error between any function $f \in \mathcal{H}$ and its estimator based on RFFs. As discussed in Section 4.2, we would like to approximate a function $f \in \mathcal{H}$ at observation points using $\tilde{f} \in \tilde{\mathcal{H}}$, with preferably as small function norm $\|\tilde{f}\|_{\tilde{\mathcal{H}}}$ as possible. As such, the estimation of $\mathbf{f}_x = [f(x_1), \ldots, f(x_n)]^T$ with $\tilde{\mathbf{f}}_x = [\tilde{f}(x_1), \ldots, \tilde{f}(x_n)]^T$ can be formulated via the following optimization problem:

$$\min_{\beta} \frac{1}{n}\|\mathbf{f}_x - \mathbf{Z}_q\beta\|_2^2 + \lambda s\|\beta\|_2^2 \,.$$

Lemma 4.6. *Suppose that Assumption A.1 holds and that the conditions on sampling measure $\tilde{l}$ from Theorem 4.1 apply as well. If*

$$s \geq 5d_{\tilde{l}} \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta} \,,$$

then for all $\delta \in (0,1)$ and any $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}} \leq 1$, with probability greater than $1-\delta$, the following holds

$$\min_{\beta} \left\{ \frac{1}{n}\|\mathbf{f}_x - \mathbf{Z}_q\beta\|_2^2 + \lambda s\|\beta\|_2^2 \right\} \leq 2\lambda \,.$$

For the sake of brevity, we will henceforth use $\tilde{\mathcal{H}}$ to denote the hypothesis space corresponding to this optimization problem. Then, the latter bound can be written as:

$$\sup_{\|f\|_{\mathcal{H}} \leq 1} \inf_{\|\tilde{f}\|_{\tilde{\mathcal{H}}} \leq \sqrt{2}} \frac{1}{n}\|\mathbf{f}_x - \tilde{\mathbf{f}}_x\|_2^2 \leq 2\lambda \,.$$

Remark 4.10. *We note here that in the strict sense the hypothesis space $\tilde{\mathcal{H}}$ is contained in the interior of the ball of radius $\sqrt{2}$, centered at the origin (see also Proposition 4.1). To simplify our presentation and avoid carrying the cumbersome optimization problem for learning with RFFs (e.g., formally given in Lemma 4.6), we make a minor adjustment/violation in notation and refer to the whole ball of radius $\sqrt{2}$ as the hypothesis space of RFFs.*

The following lemma is important for demonstrating the risk convergence rate and its proof is given in Appendix 4.8.3. Recall that we have defined $\hat{f}^\lambda$ as the empirical estimator for the kernel ridge regression problem in Eq. (2.3).

Lemma 4.7. *Denote the in-sample prediction of $\hat{f}^\lambda$ with*

$$\hat{\mathbf{f}}_x^\lambda = [\hat{f}^\lambda(x_1), \ldots, \hat{f}^\lambda(x_n)]^T \tag{4.8}$$

and let $\{v_i\}_{i=1}^s$ be independent samples selected according to a probability density function $q(v)$, which define the feature matrix $\mathbf{Z}_q$ and the corresponding RKHS $\tilde{\mathcal{H}}$. Let $\tilde{\beta}^\lambda$ be the solution to the following optimization problem

$$\tilde{\beta}^\lambda := \min_\beta \frac{1}{n}\|\hat{\mathbf{f}}_x^\lambda - \mathbf{Z}_q\beta\|_2^2 + \lambda s\|\beta\|_2^2,$$

and denote the in-sample prediction of the resulting hypothesis with $\tilde{\mathbf{f}}_x^\lambda = \mathbf{Z}_q\tilde{\beta}^\lambda$. Then, we have

$$\frac{1}{n}\langle Y - \hat{\mathbf{f}}_x^\lambda, \hat{\mathbf{f}}_x^\lambda - \tilde{\mathbf{f}}_x^\lambda\rangle \leq \lambda\,.$$

Equipped with Lemma 4.6 and Lemma 4.7, we now prove Theorem 4.1.

Theorem 4.1. *Suppose that Assumption A.1 holds and let $\tilde{l} : \mathcal{V} \to \mathbb{R}$ be a measurable function such that $\tilde{l}(v) \geq l_\lambda(v)$ $(\forall v \in \mathcal{V})$ with $d_{\tilde{l}} = \int_{\mathcal{V}} \tilde{l}(v)dv < \infty$. Suppose also that $\{v_i\}_{i=1}^s$ are sampled independently from the probability density function $q(v) = \tilde{l}(v)/d_{\tilde{l}}$. If*

$$s \;\geq\; 5d_{\tilde{l}} \log \frac{16 d_{\mathbf{K}}^\lambda}{\delta}\,,$$

then for all $\delta \in (0,1)$*, with probability* $1-\delta$*, the excess risk of* $\tilde{f}_\beta^\lambda$ *can be upper bounded by*

$$\mathbb{E}[l_{\tilde{f}_\beta^\lambda}] - \mathbb{E}[l_{f_\mathcal{H}}] \;\leq\; 4\lambda + O\left(\frac{1}{\sqrt{n}}\right) + \mathbb{E}[l_{\hat{f}^\lambda}] - \mathbb{E}[l_{f_\mathcal{H}}]\,. \tag{4.2}$$

Proof. The proof relies on the decomposition of the learning risk of $\mathbb{E}[l_{\tilde{f}_\beta^\lambda}]$ as follows

$$\begin{aligned}
\mathbb{E}[l_{\tilde{f}_\beta^\lambda}] = \mathbb{E}[l_{\tilde{f}_\beta^\lambda}] - \mathbb{E}_n[l_{\tilde{f}_\beta^\lambda}] &\tag{4.9}\\
+ \mathbb{E}_n[l_{\tilde{f}_\beta^\lambda}] - \mathbb{E}_n[l_{\hat{f}^\lambda}] &\tag{4.10}\\
+ \mathbb{E}_n[l_{\hat{f}^\lambda}] - \mathbb{E}[l_{\hat{f}^\lambda}] &\tag{4.11}\\
+ \mathbb{E}[l_{\hat{f}^\lambda}] - \mathbb{E}[l_{f_\mathcal{H}}] &\tag{4.12}\\
+ \mathbb{E}[l_{f_\mathcal{H}}]\,. &
\end{aligned}$$

For (4.9), the bound is based on the Rademacher complexity of the RKHS $\tilde{\mathcal{H}}$, where $\tilde{\mathcal{H}}$ corresponds to the approximated kernel $\tilde{k}$. We can upper bound the Rademacher complexity of this hypothesis space using Lemma 4.1. More specifically, as $l(y, f(x))$ is the squared error loss function with y and $f(x)$ both bounded, we have that l is a Lipschitz continuous function with some constant $L > 0$. Hence,

$$\begin{aligned}
(4.9) &\leq R_n(l \circ \tilde{\mathcal{H}}) + \sqrt{\frac{8\log(2/\delta)}{n}}\\
&\leq \sqrt{2}L\frac{1}{n}\mathbb{E}_X[\sqrt{\mathrm{Tr}(\tilde{\mathbf{K}})}] + \sqrt{\frac{8\log(2/\delta)}{n}}\\
&\leq \sqrt{2}L\frac{1}{n}\sqrt{\mathbb{E}_X[\mathrm{Tr}(\tilde{\mathbf{K}})]} + \sqrt{\frac{8\log(2/\delta)}{n}}\\
&\leq \sqrt{2}L\frac{1}{n}\sqrt{nz_0^2} + \sqrt{\frac{8\log(2/\delta)}{n}}\\
&= \frac{\sqrt{2}Lz_0}{\sqrt{n}} + \sqrt{\frac{8\log(2/\delta)}{n}} \in O\left(\frac{1}{\sqrt{n}}\right),
\end{aligned} \tag{4.13}$$

where in the first inequality we applied Lemma 4.2 to $\tilde{\mathcal{H}}$, which is a RKHS contained in the ball of radius $\sqrt{2}$ centered at the origin. Moreover, the bound on $R_n(l \circ \tilde{\mathcal{H}})$ relies on the Lipschitz composition property of Rademacher complexity (Bartlett and Mendelson, 2002). For (4.11), a similar reasoning can be applied to the unit ball in the RKHS $\mathcal{H}$.

For (4.10), we recall that $\tilde{\mathbf{f}}_\beta^\lambda = \mathbf{Z}\beta_\lambda$ where β_λ is the solution of the following optimization problem

$$\beta_\lambda := \min_\beta \frac{1}{n}\|Y - \mathbf{Z}\beta\|_2^2 + \lambda s\|\beta\|_2^2 \,.$$

We now observe that

$$\begin{aligned}
\mathbb{E}_n[l_{\tilde{f}_\beta^\lambda}] - \mathbb{E}_n[l_{\hat{f}^\lambda}] &= \frac{1}{n}\|Y - \tilde{\mathbf{f}}_\beta^\lambda\|_2^2 - \frac{1}{n}\|Y - \hat{\mathbf{f}}_x^\lambda\|_2^2 \\
&= \frac{1}{n}\|Y - \mathbf{Z}\beta_\lambda\|_2^2 - \frac{1}{n}\|Y - \hat{\mathbf{f}}_x^\lambda\|_2^2 \\
&\leq \min_\beta \left\{ \frac{1}{n}\|Y - \mathbf{Z}\beta\|_2^2 + \lambda s\|\beta\|_2^2 \right\} - \frac{1}{n}\|Y - \hat{\mathbf{f}}_x^\lambda\|_2^2 \\
&= \min_\beta \left\{ \frac{1}{n}\|Y - \hat{\mathbf{f}}_x^\lambda\|_2^2 + \frac{1}{n}\|\hat{\mathbf{f}}_x^\lambda - \mathbf{Z}\beta\|_2^2 \right. \\
&\quad \left. + \frac{2}{n}\langle Y - \hat{\mathbf{f}}_x^\lambda, \hat{\mathbf{f}}_x^\lambda - \mathbf{Z}\beta\rangle + \lambda s\|\beta\|_2^2 \right\} - \frac{1}{n}\|Y - \hat{\mathbf{f}}_x^\lambda\|_2^2 \\
&= \frac{1}{n}\min_\beta \left\{ \|\hat{\mathbf{f}}_x^\lambda - \mathbf{Z}\beta\|_2^2 + 2\langle Y - \hat{\mathbf{f}}_x^\lambda, \hat{\mathbf{f}}_x^\lambda - \mathbf{Z}\beta\rangle + \lambda sn\|\beta\|_2^2 \right\} \\
&\leq \frac{1}{n}\left\{ \|\hat{\mathbf{f}}_x^\lambda - \mathbf{Z}\tilde{\beta}^\lambda\|_2^2 + \lambda sn\|\tilde{\beta}^\lambda\|_2^2 + 2\langle Y - \hat{\mathbf{f}}_x^\lambda, \hat{\mathbf{f}}_x^\lambda - \mathbf{Z}\tilde{\beta}^\lambda\rangle \right\} \\
&\leq \frac{1}{n}\|\hat{\mathbf{f}}_x^\lambda - \mathbf{Z}\tilde{\beta}^\lambda\|_2^2 + \lambda s\|\tilde{\beta}^\lambda\|_2^2 + 2\lambda \quad \text{(by Lemma 4.7)} \\
&\leq 4\lambda \,. \quad \text{(by Lemma 4.6)}
\end{aligned}$$

For the last inequality, we observe that $\tilde{\beta}^\lambda$ is the solution of the following optimiza-

tion problem

$$\tilde{\beta}^{\lambda} = \min_{\beta} \frac{1}{n} \|\hat{\mathbf{f}}_x^{\lambda} - \mathbf{Z}\beta\|_2^2 + \lambda s \|\beta\|_2^2 \,.$$

Recall also that we have defined $\hat{f}^{\lambda}$ and $\hat{\mathbf{f}}_x^{\lambda}$ in Eq. (2.3) and Eq. (4.8), respectively. Now, observe that $\hat{f}^{\lambda} \in \mathcal{H}$ and in combination with Lemma 4.6 one obtains the upper bound on Eq. (4.10).

Finally, we combine the three results and derive

$$\mathbb{E}[l_{\tilde{f}_{\tilde{\beta}}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \leq 4\lambda + O\left(\frac{1}{\sqrt{n}}\right) + \mathbb{E}[l_{\hat{f}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \,. \tag{4.14}$$

□

4.6.3 Proof of Theorem 4.2

To prove Theorem 4.2, we rely on the notion of local Rademacher complexity instead of the global one. More specifically, the bound in Theorem 4.1 is not sharp because when analysing Eqs. (4.9) and (4.11), we used the global Rademacher complexity that accounts for the whole RKHS. As empirically optimal hypotheses are typically concentrated around $f_{\mathcal{H}}$, we could just analyze the local space around it. In particular, we can apply Lemma 4.4 to Eqs. (4.9) and (4.11).

To this end, we define the transformed function class as $l_{\mathcal{H}} \coloneqq \{(x, y) \mapsto l(f(x), y) \mid f \in \mathcal{H}\}$, for any RKHS $\mathcal{H}$ and a loss function l. We now would like to apply Lemma 4.4 to the function class $l_{\mathcal{H}}$. First, it is easy to see that $\mathbb{E}[l_f^2] \leq B\mathbb{E}[l_f]$ for some constant B since l_f is bounded. Now if we assume that there exists a sub-root function $\hat{\psi}_n(r)$ such that it satisfies:

$$\hat{\psi}_n(r) \geq c_1 \hat{R}_n\{l_f \in \text{star}(l_{\mathcal{H}}, 0) \mid \mathbb{E}_n[l_f^2] \leq r\} + \frac{c_2}{n} \log \frac{1}{\delta} \,,$$

then with high probability, we have

$$\mathbb{E}[l_f] \leq \frac{D}{D-1}\mathbb{E}_n[l_f] + \frac{6D}{B}\hat{r}^* + \frac{c_3}{n}\log\frac{1}{\delta}\,,$$

where r^* is the fixed point of $\hat{\psi}_n(r)$.

Hence, our job now is to find a proper $\hat{\psi}_n(r)$ such that we can compute its fixed point r^*. To this end, we define $\hat{f} = \inf_{f\in\mathcal{H}} \mathbb{E}_n[l_f] = \inf_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n (f(x_i)-y_i)^2$ for a given training sample $\{(x_i,y_i)\}_{i=1}^n$. Observe that for all $l_f \in l_{\mathcal{H}}$

$$\begin{aligned}
\mathbb{E}_n[l_f^2] &\geq (\mathbb{E}_n[l_f])^2 \quad (x^2 \text{ is convex}) \\
&\geq (\mathbb{E}_n[l_f])^2 - (\mathbb{E}_n[l_{\hat{f}}])^2 \\
&\geq 2\mathbb{E}_n[l_{\hat{f}}]\,\mathbb{E}_n[l_f - l_{\hat{f}}] \quad (\text{since } a^2-b^2 \geq 2b(a-b), \forall a,b\geq 0) \\
&\geq \frac{2}{B}\mathbb{E}_n[l_{\hat{f}}]\,\mathbb{E}_n[(f-\hat{f})^2]\,. \quad (\text{Lemma 4.12 in Appendix 4.8.4}) \quad (4.15)
\end{aligned}$$

Hence, to obtain a lower bound on $\mathbb{E}_n[l_f^2]$ expressed solely in terms of $\mathbb{E}_n[(f-\hat{f})^2]$, we need to find a lower bound of $\mathbb{E}_n[l_{\hat{f}}]$. Since $\mathbb{E}_n[l_{\hat{f}}] = \frac{1}{n}\sum_{i=1}^n (\hat{f}(x_i)-y_i)^2$, we have $\mathbb{E}_P\left[\mathbb{E}_n[l_{\hat{f}}]\right] \geq \mathbb{E}[l_{f_{\mathcal{H}}}] \geq \sigma^2$, where we recall σ^2 is the variance of ϵ defined in Assumption A.1. In addition, for each labeled example (x_i,y_i), $l(\hat{f}(x_i),y_i)$ is bounded and i.i.d. Applying Hoeffding lemma, we have for all $\delta\in(0,1)$, with probability greater than $1-\delta$, $\mathbb{E}_n[l_{\hat{f}}]$ is lower bounded by some constant e_0. Hence, with probability greater than $1-\delta$, Eq. (4.15) becomes

$$\mathbb{E}_n[l_f^2] \geq \frac{e_1}{B}\mathbb{E}_n[(f-\hat{f})^2] =: e_2\mathbb{E}_n[(f-\hat{f})^2]\,.$$

As a result of this, we have the following inequality for the two function classes

$$\{l_f \in \text{star}(l_{\mathcal{H}},0) \mid \mathbb{E}_n[l_f^2] \leq r\} \subseteq \{l_f \in \text{star}(l_{\mathcal{H}},0) \mid \mathbb{E}_n[(f-\hat{f})^2] \leq \frac{r}{e_2}\}\,.$$

Recall that for a function class $\mathcal{H}$, we denote its empirical Rademacher complexity by $\hat{R}_n(\mathcal{H})$. Then, we have the following inequality

$$
\begin{aligned}
&\hat{R}_n\{l_f \in \mathrm{star}(l_{\mathcal{H}}, 0) \mid \mathbb{E}_n[l_f^2] \leq r\} \leq \\
&\hat{R}_n\{l_f \in \mathrm{star}(l_{\mathcal{H}}, 0) \mid \mathbb{E}_n[(f - \hat{f})^2] \leq \frac{r}{e_2}\} = \\
&\hat{R}_n\{l_f - l_{\hat{f}} \mid \mathbb{E}_n[(f - \hat{f})^2] \leq \frac{r}{e_2} \ \wedge \ l_f \in \mathrm{star}(l_{\mathcal{H}}, 0)\} \leq \\
&L\hat{R}_n\{f - \hat{f} \mid \mathbb{E}_n[(f - \hat{f})^2] \leq \frac{r}{e_2} \ \wedge \ f \in \mathcal{H}\} \leq \\
&L\hat{R}_n\{f - g \mid \mathbb{E}_n[(f - g)^2] \leq \frac{r}{e_2} \ \wedge \ f, g \in \mathcal{H}\} \leq \\
&2L\hat{R}_n\{f \in \mathcal{H} \mid \mathbb{E}_n[f^2] \leq \frac{1}{4}\frac{r}{e_2}\} = \\
&2L\hat{R}_n\{f \in \mathcal{H} \mid \mathbb{E}_n[f^2] \leq e_3 r\} \ ,
\end{aligned}
\tag{4.16}
$$

where the last inequality was proved in Bartlett et al. (2005, Corollary 6.7). Now, since $\mathcal{H}$ is a RKHS with kernel k, applying Lemma 4.5 gives an upper bound of Eq. (4.16). We can then derive the following lemma which gives us the proper sub-root function $\hat{\psi}_n$. The lemma is proved in Appendix 4.8.5.

Lemma 4.8. *Assume $\{x_i, y_i\}_{i=1}^n$ is an independent sample from a probability measure P defined on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y}$ has a bounded range. Let k be a positive definite kernel with the RKHS $\mathcal{H}$ and let $\hat{\lambda}_1 \geq \cdots, \geq \hat{\lambda}_n$ be the eigenvalues of the normalized kernel Gram-matrix. Denote the squared error loss function by $l(f(x), y) = (f(x) - y)^2$ and fix $\delta \in (0, 1)$. If*

$$
\hat{\psi}_n(r) = 2Lc_1 \left(\frac{2}{n} \sum_{i=1}^{n} \min\{r, \hat{\lambda}_i\} \right)^{1/2} + \frac{c_2}{n} \log(\frac{1}{\delta}) \ ,
$$

then for all $l_f \in l_{\mathcal{H}}$ and $D > 1$, with probability $1 - \delta$,

$$
\mathbb{E}[l_f] \leq \frac{D}{D-1} \mathbb{E}_n[l_f] + \frac{6D}{B} \hat{r}^* + \frac{c_3}{n} \log(\frac{1}{\delta}) \ .
$$

Moreover, the fixed point $\hat{r}^*$ defined with $\hat{r}^* = \hat{\psi}_n(\hat{r}^*)$ can be upper bounded by

$$\hat{r}^* \leq \min_{0\leq h\leq n} \left(e_0 \frac{h}{n} + \sqrt{\frac{1}{n}\sum_{i>h} \hat{\lambda}_i} \right) ,$$

where e_0 is a constant.

We are now ready to deliver the proof of Theorem 4.2.

Theorem 4.2. *Suppose that Assumption A.1 holds and that the conditions on sampling measure $\tilde{l}$ from Theorem 4.1 apply to this setting. If*

$$s \;\geq\; 5 d_{\tilde{l}} \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta}$$

then for all $D > 1$ and $\delta \in (0,1)$, with probability $1-\delta$, the excess risk of $\tilde{f}_{\beta}^{\lambda}$ can be bounded by

$$\mathbb{E}[l_{\tilde{f}_{\beta}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \;\leq\; \frac{12D}{B}\hat{r}_{\mathcal{H}}^{*} + 4\frac{D}{D-1}\lambda + O\left(\frac{1}{n}\right) + \mathbb{E}[l_{\hat{f}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \,. \quad (4.3)$$

Furthermore, denoting the eigenvalues of the normalized kernel matrix $(1/n)\mathbf{K}$ with $\{\hat{\lambda}_i\}_{i=1}^{n}$, we have that

$$\hat{r}_{\mathcal{H}}^{*} \leq \min_{0\leq h\leq n} \left(e_0 \frac{h}{n} + \sqrt{\frac{1}{n}\sum_{i>h} \hat{\lambda}_i} \right) , \quad (4.4)$$

where $B, e_0 > 0$ are some constant and $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_n$.

Proof. We decompose $\mathbb{E}[l_{\tilde{f}_{\beta}^{\lambda}}]$ with $D > 1$ as follows:

$$\begin{aligned}
\mathbb{E}[l_{\tilde{f}_{\beta}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \leq \mathbb{E}[l_{\tilde{f}_{\beta}^{\lambda}}] - \frac{D}{D-1}\mathbb{E}_n[l_{\tilde{f}_{\beta}^{\lambda}}] & \quad (4.17)\\
+\frac{D}{D-1}(\mathbb{E}_n[l_{\tilde{f}_{\beta}^{\lambda}}] - \mathbb{E}_n[l_{\hat{f}^{\lambda}}]) & \quad (4.18)\\
+\frac{D}{D-1}\mathbb{E}_n[l_{\hat{f}^{\lambda}}] - \mathbb{E}[l_{\hat{f}^{\lambda}}] & \quad (4.19)\\
+\mathbb{E}[l_{\hat{f}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \,. & \quad (4.20)
\end{aligned}$$

We have already demonstrated that

$$\text{Eq. } (4.18) \leq 4\frac{D}{D-1}\lambda \, .$$

For Eq. (4.17), we can derive an upper bound using Lemma 4.8. Similarly, we can upper bound Eq. (4.19) by interchaning the positions of empirical and expected risk functions in Lemma 4.8. The proof of the latter result resembles that of Lemma 4.8 and is a direct consequence of the second part of Theorem 4.1 in Bartlett et al. (2005). However, note that $\tilde{f}_{\beta}^{\lambda}$ and $\hat{f}^{\lambda}$ belong to different RKHSs. As a result, we have

$$\begin{aligned}
\text{Eq. } (4.17) &\leq \frac{6D}{B}\hat{r}^{*}_{\tilde{\mathcal{H}}} + O(1/n) \, , \\
\text{Eq. } (4.19) &\leq \frac{6D}{B}\hat{r}^{*}_{\mathcal{H}} + O(1/n) \, .
\end{aligned}$$

Now, combining these inequalities together we deduce

$$\begin{aligned}
\mathbb{E}[l_{\tilde{f}_{\beta}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] &\leq \frac{6D}{B}\hat{r}^{*}_{\tilde{\mathcal{H}}} + \frac{6D}{B}\hat{r}^{*}_{\mathcal{H}} + 4\frac{D}{D-1}\lambda + O(1/n) \\
&\quad + \mathbb{E}[l_{\hat{f}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \\
&\leq \frac{12D}{B}\hat{r}^{*}_{\mathcal{H}} + 4\frac{D}{D-1}\lambda + O(1/n) \\
&\quad + \mathbb{E}[l_{\hat{f}^{\lambda}}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \, .
\end{aligned}$$

The last inequality holds because the eigenvalues of the Gram-matrix for the RKHS $\tilde{\mathcal{H}}$ decay faster than the eigenvalues of $\mathcal{H}$. Consequently, we have that $\hat{r}^{*}_{\tilde{\mathcal{H}}} \leq \hat{r}^{*}_{\mathcal{H}}$.

Now, Lemma 4.8 implies that

$$\hat{r}^{*}_{\mathcal{H}} \leq \min_{0\leq h\leq n}\left(e_0\frac{h}{n} + \sqrt{\frac{1}{n}\sum_{i>h}\hat{\lambda}_i}\right) . \tag{4.21}$$

There are two cases that merit a discussion here. First, if the eigenvalues of $\mathbf{K}$ decay exponentially then setting $h = \lceil \log n \rceil$ implies that

$$\hat{r}^*_{\mathcal{H}} \leq O\left(\frac{\log n}{n}\right).$$

Now, according to Caponnetto and De Vito (2007)

$$\mathbb{E}[l_{\hat{f}^\lambda}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \in O\left(\frac{\log n}{n}\right),$$

and, thus, if we set $\lambda \propto \frac{\log n}{n}$ then the learning risk rate can be upper bounded by

$$\mathbb{E}[l_{\tilde{f}^\lambda_\beta}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \in O\left(\frac{\log n}{n}\right).$$

On the other hand, if $\mathbf{K}$ has finitely many non-zero eigenvalues (t), then setting $h \geq t$ implies that

$$\hat{r}^*_{\mathcal{H}} \in O\left(\frac{1}{n}\right).$$

Moreover, in this case, $\mathbb{E}[l_{\hat{f}^\lambda}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \in O(\frac{1}{n})$ and setting $\lambda \propto \frac{1}{n}$, we deduce that

$$\mathbb{E}[l_{\tilde{f}^\lambda_\beta}] - \mathbb{E}[l_{f_{\mathcal{H}}}] \leq O\left(\frac{1}{n}\right).$$

□

4.6.4 Proof of Theorem 4.5

Theorem 4.5. *Suppose that Assumption B.1 holds and that the conditions on sampling measure $\tilde{l}$ from Theorem 4.1 apply to the setting with a Lipschitz continuous loss. If*

$$s \geq 5 d_{\tilde{l}} \log \frac{16 d^\lambda_{\mathbf{K}}}{\delta}$$

then for all $\delta \in (0,1)$, with probability $1-\delta$, the learning risk of g_β^λ can be upper bounded by

$$\mathbb{E}[l_{g_\beta^\lambda}] \leq \mathbb{E}[l_{g_\mathcal{H}}] + \sqrt{2\lambda} + O\left(\frac{1}{\sqrt{n}}\right). \tag{4.5}$$

Proof. The proof is similar to Theorem 4.1. In particular, we decompose the expected learning risk as

$$\begin{aligned}
\mathbb{E}[l_{g_\beta^\lambda}] = \mathbb{E}[l_{g_\beta^\lambda}] - \mathbb{E}_n[l_{g_\beta^\lambda}] & \quad (4.22)\\
+\mathbb{E}_n[l_{g_\beta^\lambda}] - \mathbb{E}_n[l_{g_\mathcal{H}}] & \quad (4.23)\\
+\mathbb{E}_n[l_{g_\mathcal{H}}] - \mathbb{E}[l_{g_\mathcal{H}}] & \quad (4.24)\\
+\mathbb{E}[l_{g_\mathcal{H}}]\,. &
\end{aligned}$$

Now, (4.22) and (4.24) can be upper bounded similar to Theorem 4.1, through the Rademacher complexity bound from Lemma 4.2. For (4.23), we have

$$\begin{aligned}
&\mathbb{E}_n[l_{g_\beta^\lambda}] - \mathbb{E}_n[l_{g_\mathcal{H}}] = \\
&\frac{1}{n}\sum_{i=1}^n l(y_i, g_\beta^\lambda(x_i)) - \frac{1}{n}\sum_{i=1}^n l(y_i, g_\mathcal{H}(x_i)) = \\
&\frac{1}{n}\inf_{\|g_\beta\|}\sum_{i=1}^n l(y_i, g_\beta(x_i)) - \frac{1}{n}\sum_{i=1}^n l(y_i, g_\mathcal{H}(x_i)) \\
&\leq \inf_{\|g_\beta\|}\frac{1}{n}\sum_{i=1}^n |g_\beta(x_i) - g_\mathcal{H}(x_i)| \\
&\leq \inf_{\|g_\beta\|}\sqrt{\frac{1}{n}\sum_{i=1}^n |g_\beta(x_i) - g_\mathcal{H}(x_i)|^2} \\
&\leq \sup_{\|g\|}\inf_{\|g_\beta\|}\sqrt{\frac{1}{n}\|g - g_\beta\|_2^2} \leq \sqrt{2\lambda}\,.
\end{aligned}$$

□

4.6.5 Proof of Theorem 4.7

To prove Theorem 4.7, we adopt a similar strategy to the proof of Theorem 4.2. In particular, we utilize the properties of the local Rademacher complexity by applying Lemma 4.4 to the decomposition of the learning risk in the Lipschitiz continuous loss case, namely Eqs. (4.22) and (4.24). In order to do that, we need two steps. The first step is to find a proper sub-root function $\hat{\psi}_n(r)$. The second step is to find the fixed point of $\hat{\psi}_n(r)$. Hence, the following is devoted to solving these two problems.

Theorem 4.7. *Suppose that Assumptions B.1-2 hold and that the conditions on sampling measure $\tilde{l}$ from Theorem 4.1 apply to the setting with a Lipschitz continuous loss. If*

$$s \geq 5d_{\tilde{l}} \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta}$$

then for all $D > 1$ and $\delta \in (0,1)$ with probability greater than $1-\delta$, we have

$$\mathbb{E}[l_{g_{\hat{\beta}}^{\lambda}}] \leq \frac{12D}{B} \hat{r}_{\mathcal{H}}^{*} + \frac{D}{D-1}\sqrt{2\lambda} + O(1/n) + \mathbb{E}[l_{g_{\mathcal{H}}}] \,. \tag{4.6}$$

Furthermore, denoting the eigenvalues of the normalized kernel matrix $(1/n)\mathbf{K}$ with $\{\hat{\lambda}_i\}_{i=1}^{n}$, we have that $\hat{r}_{\mathcal{H}}^{*}$ can be upper bounded by

$$\hat{r}_{\mathcal{H}}^{*} \leq \min_{0 \leq h \leq n} \left(b_0 \frac{h}{n} + \sqrt{\frac{1}{n} \sum_{i>h} \hat{\lambda}_i} \right) \,, \tag{4.7}$$

where B and b_0 are some constants.

Proof. First, recall that we have defined the transformed function class as $g_{\mathcal{H}} \coloneqq \{(x,y) \mapsto g(f(x),y) \mid g \in \mathcal{H}\}$ for a Lipschitz continuous loss function l and

$\hat{g} = \inf_{g\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} l(g(x_i), y_i)$. We observe that for any $l_g \in l_{\mathcal{H}}$, we have

$$\begin{aligned}
\mathbb{E}_n[l_g^2] &\geq (\mathbb{E}_n[l_g])^2 \ \ (x^2 \text{ is convex}) \\
&\geq (\mathbb{E}_n[l_g])^2 - (\mathbb{E}_n[l_{\hat{g}}])^2 \\
&\geq 2\mathbb{E}_n[l_{\hat{g}}]\ \mathbb{E}_n[l_g - l_{\hat{g}}] \ \ (\text{since } a^2 - b^2 \geq 2b(a-b), \forall a, b \geq 0) \\
&\geq \frac{2}{B}\mathbb{E}_n[l_{\hat{g}}]\mathbb{E}_n[(g-\hat{g})^2]\ . \ \ (\text{Assumption B.4}) \qquad (4.25)
\end{aligned}$$

Following the reasoning for Eq. (4.15) in Section 4.6.3, we can see that for all $\delta \in (0,1)$, with probability greater than $1-\delta$, the term $\mathbb{E}_n[l_{\hat{g}}]$ can be lower bounded by a constant, denoted with b_0. Hence, Eq. (4.25) becomes

$$\mathbb{E}_n[l_g^2] \geq b_1 \mathbb{E}_n[(g-\hat{g})^2]\ .$$

Similar to Section 4.6.3, we have

$$\{l_g \in l_{\mathcal{H}} \mid \mathbb{E}_n[l_g^2] \leq r\} \subseteq \{l_g \in l_{\mathcal{H}} \mid \mathbb{E}_n[(g-\hat{g})^2] \leq \frac{r}{b_1}\}\ .$$

This further implies that

$$\hat{R}_n\{l_g \in l_{\mathcal{H}} \mid \mathbb{E}_n[l_g^2] \leq r\} \leq 2L\hat{R}_n\{g \in \mathcal{H} \mid \mathbb{E}_n[g^2] \leq b_2 r\}\ ,$$

where we recall L is the Lipschitz constant of the loss function l. By appealing to Lemma 4.5, we obtain an upper bound on $\hat{R}_n\{g \in \mathcal{H} \mid \mathbb{E}_n[g^2] \leq b_2 r\}$. Applying Lemma 4.8 to the function class $l_{\mathcal{H}}$, we have that for all $l_g \in l_{\mathcal{H}}$ and $D > 1$, with probability greater than $1-\delta$,

$$\mathbb{E}[l_g] \leq \frac{D}{D-1}\mathbb{E}_n[l_g] + \frac{12D}{B}\hat{r}^* + \frac{c_1}{n}\log(\frac{1}{\delta})\ . \qquad (4.26)$$

Moreover, the fixed point $\hat{r}^*$ can be upper bounded by

$$\hat{r}^* \leq \min_{0\leq h\leq n} \left(b_0 \frac{h}{n} + \sqrt{\frac{1}{n}\sum_{i>h} \hat{\lambda}_i} \right) ,$$

where b_0 is a constant.

Having this in mind, we turn to the risk decomposition:

$$\begin{aligned}
\mathbb{E}[l_{g_\beta^\lambda}] = \mathbb{E}[l_{g_\beta^\lambda}] - \frac{D}{D-1}\mathbb{E}_n[l_{g_\beta^\lambda}] & \quad (4.27)\\
+\frac{D}{D-1}\mathbb{E}_n[l_{g_\beta^\lambda}] - \frac{D}{D-1}\mathbb{E}_n[l_{g_\mathcal{H}}] & \quad (4.28)\\
+\frac{D}{D-1}\mathbb{E}_n[l_{g_\mathcal{H}}] - \mathbb{E}[l_{g_\mathcal{H}}] & \quad (4.29)\\
+\mathbb{E}[l_{g_\mathcal{H}}] \, . &
\end{aligned}$$

The term in Eq. (4.28) can be upper bounded by $\frac{D}{D-1}\sqrt{2\lambda}$ using the results from Section 4.6.4. As $g_\beta^\lambda \in \tilde{\mathcal{H}}$, we can upper bound Eq. (4.27) using the result from Eq. (4.26) adjusted to the RKHS $\tilde{\mathcal{H}}$. We repeat the same procedure for Eq. (4.29) using Eq. (4.26) applied to $\mathcal{H}$. Now, combing these three auxiliary results, we obtain that with probability greater than $1-\delta$,

$$\mathbb{E}[l_{g_\beta^\lambda}] \leq \frac{12D}{B_0}\hat{r}^*_\mathcal{H} + \frac{D}{D-1}\sqrt{2\lambda} + O(1/n) + \mathbb{E}[l_{g_\mathcal{H}}] \, , \quad (4.30)$$

where $\hat{r}^*_\mathcal{H}$ can be upper bounded by

$$\hat{r}^*_\mathcal{H} \leq \min_{0\leq h\leq n} \left(b_0 \frac{h}{n} + \sqrt{\frac{1}{n}\sum_{i>h} \hat{\lambda}_i} \right) \, . \quad (4.31)$$

□

4.6.6 Proof of Theorem 4.8

Theorem 4.8. *Suppose that Assumption A.1 holds and consider the regression problem defined with a shift-invariant kernel k, a sample of examples $\{(x_i, y_i)\}_{i=1}^n$, and a regularization parameter λ. Let s be the number of RFFs in the pool of features from Algorithm 3, sampled using the spectral measure $p(\cdot)$ from Eq. (2.9) and the regularization parameter λ. Denote with $\tilde{f}_m^{\lambda^*}$ the ridge regression estimator obtained using a regularization parameter λ^* and a set of RFFs $\{v_i\}_{i=1}^m$ returned by Algorithm 3. If*

$$s \geq \frac{7z_0^2}{\lambda} \log \frac{(16 d_{\mathbf{K}}^{\lambda})}{\delta} \quad \text{and} \quad m \geq 5 d_{\mathbf{K}}^{\lambda^*} \log \frac{(16 d_{\mathbf{K}}^{\lambda^*})}{\delta} ,$$

then for all $\delta \in (0,1)$, with probability $1-\delta$, the learning risk of $\tilde{f}_m^{\lambda^*}$ can be upper bounded by

$$\mathbb{E}[l_{\tilde{f}_m^{\lambda^*}}] \leq \mathbb{E}[l_{f_{\mathcal{H}}}] + 4\lambda + 4\lambda^* + O\left(\frac{1}{\sqrt{n}}\right).$$

Moreover, this upper bound holds for $m \in \Omega(\frac{s}{n\lambda})$.

Proof. Suppose the examples $\{x_i, y_i\}_{i=1}^n$ are independent and identically distributed and that the kernel k can be decomposed as in Eq. (2.9). Let $\{v_i\}_{i=1}^s$ be an independent sample selected according to $p(v)$. Then, using these s features we can approximate the kernel as

$$\begin{aligned} \tilde{k}(x,y) &= \frac{1}{s} \sum_{i=1}^{s} z(v_i, x) z(v_i, y) \\ &= \int_V z(v,x) z(v,y) d\hat{P}(v) , \end{aligned} \tag{4.32}$$

where $\hat{P}$ is the empirical measure on $\{v_i\}_{i=1}^s$. Denote the RKHS associated with

kernel $\tilde{k}$ by $\tilde{\mathcal{H}}$ and suppose that kernel ridge regression was performed with the approximate kernel $\tilde{k}$. From Theorem 4.1 and Corollary 4.2, it follows that if

$$s \geq \frac{7z_0^2}{\lambda} \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta} \ ,$$

then for all $\delta \in (0,1)$, with probability $1-\delta$, the risk convergence rate of the kernel ridge regression estimator based on RFFs can be upper bounded by

$$\mathbb{E}[l_{f_\alpha^\lambda}] \leq 4\lambda + O\left(\frac{1}{\sqrt{n}}\right) + \mathbb{E}[l_{f_{\mathcal{H}}}] \ . \tag{4.33}$$

Note that in Eq. (4.33) we have used the fact that $\mathbb{E}[l_{f_{\mathcal{H}}}]$ differs with $\mathbb{E}[l_{\hat{f}^\lambda}]$ by at most $O(1/\sqrt{n})$. Let $f_{\tilde{\mathcal{H}}}$ be the function in the RKHS $\tilde{\mathcal{H}}$ achieving the minimal risk, i.e., $\mathbb{E}[l_{f_{\tilde{\mathcal{H}}}}] = \inf_{f \in \tilde{\mathcal{H}}} \mathbb{E}[l_f]$. We now treat $\tilde{k}$ as the actual kernel that can be decomposed via the expectation with respect to the empirical measure in Eq. (4.32) and re-sample features from the set $\{v_i\}_{i=1}^s$, but this time the sampling is performed using the optimal ridge leverage scores. As $\tilde{k}$ is the actual kernel, it follows from Eq. (2.14) that the leverage function in this case can be defined by

$$l_\lambda(v) = p(v)\mathbf{z}_v(\mathbf{x})^T(\tilde{\mathbf{K}} + n\lambda I)^{-1}\mathbf{z}_v(\mathbf{x}) \ .$$

Now, observe that

$$l_\lambda(v_i) = p(v_i)[\mathbf{Z}_s^T(\tilde{\mathbf{K}} + n\lambda I)^{-1}\mathbf{Z}_s]_{ii}$$

where $[A]_{ii}$ denotes the i-th diagonal element of matrix A. As $\tilde{\mathbf{K}} = (1/s)\mathbf{Z}_s\mathbf{Z}_s^T$, then the Woodbury inversion lemma implies that

$$l_\lambda(v_i) = p(v_i)[\mathbf{Z}_s^T\mathbf{Z}_s(\frac{1}{s}\mathbf{Z}_s^T\mathbf{Z}_s + n\lambda I)^{-1}]_{ii} \ .$$

If we let $l_\lambda(v_i) = p_i$, then the optimal distribution for $\{v_i\}_{i=1}^s$ is multinomial with individual probabilities $q(v_i) = p_i/(\sum_{j=1}^s p_j)$. Hence, we can re-sample m features according to $q(v)$ and perform linear ridge regression using the sampled leverage weighted features. Denoting this estimator with $\tilde{f}_m^{\lambda^*}$ and the corresponding number of degrees of freedom with $d_{\tilde{\mathbf{K}}}^\lambda = \mathrm{Tr}\tilde{\mathbf{K}}(\tilde{\mathbf{K}} + n\lambda)^{-1}$, we deduce (using Theorem 4.1 and Corollary 4.1)

$$\mathbb{E}[l_{\tilde{f}_m^{\lambda^*}}] \leq 4\lambda^* + O\left(\frac{1}{\sqrt{n}}\right) + \mathbb{E}[l_{f_{\tilde{\mathcal{H}}}}]\,, \tag{4.34}$$

with the number of features $l \propto d_{\tilde{\mathbf{K}}}^\lambda$, and we again used the fact that $\mathbb{E}[l_{f_{\tilde{\mathcal{H}}}}]$ differs from $\mathbb{E}[l_{\tilde{f}_m^{\lambda^*}}]$ by at most $O(1/\sqrt{n})$.

As $f_{\tilde{\mathcal{H}}}$ is the function achieving the minimal risk over $\tilde{\mathcal{H}}$, we can conclude that $\mathbb{E}[l_{f_{\tilde{\mathcal{H}}}}] \leq \mathbb{E}[l_{f_\alpha^\lambda}]$. Now, combining Eqs. (4.33) and (4.34), we obtain the final bound on $\mathbb{E}[l_{\tilde{f}_m^{\lambda^*}}]$. □

4.7 Discussion

We have investigated the generalization properties of learning with RFFs in the context of different kernel methods: kernel ridge regression, support vector machines, and kernel logistic regression. In particular, we have given generic bounds on the number of features required for consistency of learning with two sampling strategies: *leverage weighted* and *plain RFFs*. The derived convergence rates account for the complexity of the target hypothesis and the structure of the RKHS with respect to the marginal distribution of a data-generating process. In addition to this, we have also proposed an algorithm for fast approximation of empirical leverage scores and demonstrated its superiority in both theoretical and empirical analyses.

For kernel ridge regression, Avron et al. (2017) and Rudi and Rosasco (2017) extensively analyze the performance of learning with RFFs. In particular, Avron et al. (2017) show that $o(n)$ features are enough to guarantee a good estimator in terms of its *empirical risk*. The authors of that work also propose a modified data-dependent sampling distribution and demonstrate that a further reduction in the number of RFFs is possible for leverage weighted sampling. However, their results do not provide a convergence rate for the *learning risk* of the estimator which could still potentially imply that computational savings come at the expense of statistical efficiency. Furthermore, the modified sampling distribution can only be used in the 1D Gaussian kernel case. While Avron et al. (2017) focus on bounding the empirical risk of an estimator, Rudi and Rosasco (2017) give a comprehensive study of the generalization properties of RFFs for kernel ridge regression by bounding the learning risk of an estimator. The latter work for the first time shows that $\Omega(\sqrt{n}\log n)$ features are sufficient to guarantee the (kernel ridge regression) minimax rate and observes that further improvements to this result are possible by relying on a data-dependent sampling strategy. However, such a distribution is defined in a complicated way and it is not clear how one could devise a practical algorithm by sampling from it. While in our analysis of learning with RFFs we also bound the learning risk of an estimator, the analysis is not restricted to kernel ridge regression and covers other kernel methods such as support vector machines and kernel logistic regression. In addition to this, our derivations are much simpler compared to Rudi and Rosasco (2017) and provide sharper bounds in some cases. More specifically, we demonstrate that $\Omega(\sqrt{n}\log\log n)$ features are sufficient to attain the minimax rate in the case where eigenvalues of the Gram-matrix have a geometric/exponential decay. In other cases, we recover the results from Rudi and Rosasco (2017). Another important difference with respect to this work is that we consider a data-dependent sampling distribution based on empirical ridge leverage

scores, showing that it can further reduce the number of features and in this way provide a more effective estimator.

In addition to the squared error loss, we also investigate the properties of learning with RFFs using the Lipschitz continuous loss functions and the 0-1 loss function. Both Rahimi and Recht (2009) and Bach (2017b) study this problem setting and obtain that $\Omega(n)$ features are needed to ensure $O(1/\sqrt{n})$ learning risk convergence rate. Moreover, Bach (2017b) define an optimal sampling distribution by referring to the leverage score function based on the integral operator and show that the number of features can be significantly reduced when the eigenvalues of a Gram-matrix exhibit a fast decay. The $\Omega(n)$ requirement on the number of features is too restrictive and precludes any computational savings. Also, the optimal sampling distribution is typically intractable. In our analysis, we demonstrate that a sharper bound on the requirede number of features than previous work. Moreover, we provide a much simpler form of the empirical leverage score distribution and demonstrate that the number of features can be significantly smaller than n, without incurring any loss of statistical efficiency.

Having given risk convergence rates for learning with RFFs, we provide a fast and practical algorithm for sampling them in a data-dependent way, such that they approximate the ridge leverage score distribution. In the kernel ridge regression setting, our theoretical analysis demonstrates that, compared to spectral measure sampling, significant computational savings can be achieved while preserving the statistical properties of the estimators. Furthermore, we verify our findings empirically on simulated and real-world datasets. An interesting extension of our empirical analysis would be a thorough and comprehensive comparison of the proposed leverage weighted sampling scheme to other recently proposed data-dependent strategies for selecting good features (e.g., Rudi et al., 2018), as well as a comparison to the Nyström method.

4.8 Auxilliary Results

4.8.1 Bernstein Inequality

The next lemma is the matrix Bernstein inequality, cited from (Avron et al., 2017, Lemma 27) which is a restatement of Corollary 7.3.3 in Tropp (2015) with some fix in the typos.

Lemma 4.9. *(Bernstein inequality, Tropp, 2015, Corollary 7.3.3) Let* $\mathbf{R}$ *be a fixed* $d_1 \times d_2$ *matrix over the set of complex/real numbers. Suppose that* $\{\mathbf{R}_1, \cdots, \mathbf{R}_n\}$ *is an independent and identically distributed sample of* $d_1 \times d_2$ *matrices such that*

$$\mathbb{E}[\mathbf{R}_i] = \mathbf{R} \qquad \textit{and} \qquad \|\mathbf{R}_i\|_2 \leq L \,,$$

where $L > 0$ *is a constant independent of the sample. Furthermore, let* $\mathbf{M}_1, \mathbf{M}_2$ *be semidefinite upper bounds for the matrix-valued variances*

$$\mathrm{Var}_1[\mathbf{R}_i] \preceq \mathbb{E}[\mathbf{R}_i \mathbf{R}_i^T] \preceq \mathbf{M}_1$$
$$\mathrm{Var}_2[\mathbf{R}_i] \preceq \mathbb{E}[\mathbf{R}_i^T \mathbf{R}_i] \preceq \mathbf{M}_2 \,.$$

Let $m = \max(\|\mathbf{M}_1\|_2, \|\mathbf{M}_2\|_2)$ *and* $d = \frac{Tr(\mathbf{M}_1)+Tr(\mathbf{M}_2)}{m}$. *Then, for* $\epsilon \geq \sqrt{m/n} + 2L/3n$, *we can bound*

$$\bar{\mathbf{R}}_n = \frac{1}{n}\sum_{i=1}^{n} \mathbf{R}_i$$

around its mean using the concentration inequality

$$P(\|\bar{\mathbf{R}}_n - \mathbf{R}\|_2 \geq \epsilon) \leq 4d \exp\left(\frac{-n\epsilon^2/2}{m + 2L\epsilon/3}\right) .$$

4.8.2 Proof of Lemma 4.6

The following two lemmas are required for our proof of Lemma 4.6.

Lemma 4.10. *Suppose that the assumptions from Lemma 4.6 hold and let* $\epsilon \geq \sqrt{\frac{m}{s}} + \frac{2L}{3s}$ *with constants* m *and* L *(see the proof for explicit definition). If the number of features*

$$s \geq d_{\tilde{l}}(\frac{1}{\epsilon^2} + \frac{2}{3\epsilon}) \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta} \, ,$$

then for all $\delta \in (0, 1)$*, with probability greater than* $1 - \delta$*,*

$$-\epsilon \mathbf{I} \preceq (\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} (\tilde{\mathbf{K}} - \mathbf{K})(\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} \preceq \epsilon \mathbf{I} \, .$$

Proof. Following the derivations in Avron et al. (2017), we utilize the matrix Bernstein concentration inequality to prove the result. More specifically, we observe that

$$\begin{aligned}
&(\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} \tilde{\mathbf{K}} (\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} = \\
&\frac{1}{s} \sum_{i=1}^{s} (\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} \mathbf{z}_{q,v_i}(\mathbf{x}) \mathbf{z}_{q,v_i}(\mathbf{x})^T (\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} = \\
&\frac{1}{s} \sum_{i=1}^{s} \mathbf{R}_i =: \bar{\mathbf{R}}_s \, ,
\end{aligned}$$

with

$$\mathbf{R}_i = (\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} \mathbf{z}_{q,v_i}(\mathbf{x}) \mathbf{z}_{q,v_i}(\mathbf{x})^T (\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} \, .$$

Now, observe that

$$\mathbf{R} = \mathbb{E}[\mathbf{R}_i] = (\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} \mathbf{K} (\mathbf{K} + n\lambda \mathbf{I})^{-\frac{1}{2}} \, .$$

The operator norm of $\mathbf{R}_i$ is equal to

$$\|(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{z}_{q,v_i}(\mathbf{x})\mathbf{z}_{q,v_i}(\mathbf{x})^T(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}\|_2 \ .$$

As $\mathbf{z}_{q,v_i}(\mathbf{x})\mathbf{z}_{q,v_i}(\mathbf{x})^T$ is a rank one matrix, we have that the operator norm of this matrix is equal to its trace, i.e.,

$$\begin{aligned}
&\|\mathbf{R}_i\|_2 = \\
&\text{Tr}((\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{z}_{q,v_i}(\mathbf{x})\mathbf{z}_{q,v_i}(\mathbf{x})^T(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}) = \\
&\frac{p(v_i)}{q(v_i)}\text{Tr}((\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{z}_{v_i}(\mathbf{x})\mathbf{z}_{v_i}(\mathbf{x})^T(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}) = \\
&\frac{p(v_i)}{q(v_i)}\text{Tr}(\mathbf{z}_{v_i}(\mathbf{x})^T(\mathbf{K}+n\lambda\mathbf{I})^{-1}\mathbf{z}_{v_i}(\mathbf{x})) = \\
&\frac{l_\lambda(v_i)}{q(v_i)} =: L_i \quad \text{and} \quad L_q \coloneqq \sup_i \ L_i \ .
\end{aligned}$$

Observe that $L_q = \sup_i L_i = \sup_i \frac{l_\lambda(v_i)}{q(v_i)} \leq \sup_i \frac{\tilde{l}(v_i)}{q(v_i)} = d_{\tilde{l}}$. On the other hand,

$$\begin{aligned}
&\mathbf{R}_i\mathbf{R}_i^T = \\
&(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{z}_{q,v_i}(\mathbf{x})\mathbf{z}_{q,v_i}(\mathbf{x})^T(\mathbf{K}+n\lambda\mathbf{I})^{-1}\mathbf{z}_{q,v_i}(\mathbf{x}) \\
&\cdot \mathbf{z}_{q,v_i}(\mathbf{x})^T(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}} = \\
&\frac{p(v_i)l_\lambda(v_i)}{q^2(v_i)}(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{z}_{v_i}(\mathbf{x})\mathbf{z}_{v_i}(\mathbf{x})^T(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}} \preceq \\
&\frac{\tilde{l}(v_i)}{q(v_i)}\frac{p(v_i)}{q(v_i)}(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{z}_{v_i}(\mathbf{x})\mathbf{z}_{v_i}(\mathbf{x})^T(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}} = \\
&d_{\tilde{l}}\frac{p(v_i)}{q(v_i)}(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{z}_{v_i}(\mathbf{x})\mathbf{z}_{v_i}(\mathbf{x})^T(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}} \ .
\end{aligned}$$

From the latter inequality, we obtain that

$$\mathbb{E}[\mathbf{R}_i\mathbf{R}_i^T] \preceq d_{\tilde{l}}(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{K}(\mathbf{K}+n\lambda\mathbf{I})^{-\frac{1}{2}} =: \mathbf{M}_1 \ .$$

We also have the following two equalities

$$m = \|\mathbf{M}_1\|_2 = d_{\tilde{l}} \frac{\lambda_1}{\lambda_1 + n\lambda} =: d_{\tilde{l}} d_1$$
$$d = \frac{2\,\mathrm{Tr}(\mathbf{M}_1)}{m} = 2\frac{\lambda_1 + n\lambda}{\lambda_1} d_{\mathbf{K}}^{\lambda} = 2 d_1^{-1} d_{\mathbf{K}}^{\lambda} .$$

We are now ready to apply the matrix Bernstein concentration inequality (Tropp, 2015, Corollary 7.3.3). More specifically, for $\epsilon \geq \sqrt{m/s} + 2L/3s$ and for all $\delta \in (0, 1)$, with probability $1 - \delta$, we have that

$$\begin{aligned} \mathrm{P}(\|\bar{\mathbf{R}}_s - \mathbf{R}\|_2 \geq \epsilon) &\leq 4d \exp\left(\frac{-s\epsilon^2/2}{m + 2L\epsilon/3}\right) \\ &\leq 8 d_1^{-1} d_{\mathbf{K}}^{\lambda} \exp\left(\frac{-s\epsilon^2/2}{d_{\tilde{l}} d_1 + d_{\tilde{l}} 2\epsilon/3}\right) \\ &\leq 16 d_{\mathbf{K}}^{\lambda} \exp\left(\frac{-s\epsilon^2}{d_{\tilde{l}}(1 + 2\epsilon/3)}\right) \leq \delta . \end{aligned}$$

In the third line, we have used the assumption that $n\lambda \leq \lambda_1$ and, consequently, $d_1 \in [1/2, 1)$. □

Remark 4.11. *We note here that the two considered sampling strategies lead to two different results. In particular, if we let $\tilde{l}(v) = l_\lambda(v)$ then $q(v) = l_\lambda(v)/d_{\mathbf{K}}^{\lambda}$, i.e., we are sampling proportional to the ridge leverage scores. Thus, the leverage weighted RFFs sampler requires*

$$s \geq d_{\mathbf{K}}^{\lambda} \left(\frac{1}{\epsilon^2} + \frac{2}{3\epsilon}\right) \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta} . \tag{4.35}$$

Alternatively, we can opt for the plain RFF sampling strategy by taking $\tilde{l}(v) = z_0^2 p(v)/\lambda$, with $l_\lambda(v) \leq z_0^2 p(v)/\lambda$. Then, plain RFFs sampling requires

$$s \geq \frac{z_0^2}{\lambda} \left(\frac{1}{\epsilon^2} + \frac{2}{3\epsilon}\right) \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta} . \tag{4.36}$$

Thus, the leverage weighted RFFs sampling scheme can dramatically change the number of features required to achieve a predefined approximation error in the operator norm.

Lemma 4.11. *Let $f \in \mathcal{H}$, where $\mathcal{H}$ is the RKHS associated with a kernel k. Recall we have assumed that $\|f\|_{\mathcal{H}} \leq 1, \forall f$ and $\mathbf{f}_x = [f(x_1), \cdots, f(x_n)]^T$. Let $\mathbf{K}$ be the Gram-matrix of the kernel k given by the provided set of instances. Then,*

$$\mathbf{f}_x^T \mathbf{K}^{-1} \mathbf{f}_x \leq 1 \, .$$

Proof. Recall that a function $f \in \mathcal{H}$ can be expressed as:

$$f(x) = \int_{\mathcal{V}} g(v) z(v, x) p(v) dv \qquad (\forall x \in \mathcal{X}) \, , \tag{4.37}$$

where $g \in L_2(d\tau)$ is a real-valued function with $\|f\|_{\mathcal{H}}$ equal to the minimum of $\|g\|_{L_2(d\tau)}$, over all possible decompositions of f. For a vector $\mathbf{a} \in \mathbb{R}^n$, we have that

$$\begin{aligned}
\mathbf{a}^T \mathbf{f}_x \mathbf{f}_x^T \mathbf{a} = \Big(\mathbf{f}_x^T \mathbf{a} \Big)^2 &= \Big(\sum_{i=1}^n a_i f(x_i) \Big)^2 \\
&= \Big(\sum_{i=1}^n a_i \int_{\mathcal{V}} g(v) z(v, x_i) d\tau(v) \Big)^2 \\
&= \Big(\int_{\mathcal{V}} g(v) \mathbf{z}_v(\mathbf{x})^T \mathbf{a} \, d\tau(v) \Big)^2 \\
&\leq \int_{\mathcal{V}} g(v)^2 d\tau(v) \int_{\mathcal{V}} (\mathbf{z}_v(\mathbf{x})^T \mathbf{a})^2 \, d\tau(v) \\
&= \int_{\mathcal{V}} \mathbf{a}^T \mathbf{z}_v(\mathbf{x}) \mathbf{z}_v(\mathbf{x})^T \mathbf{a} \, d\tau(v) \\
&= \mathbf{a}^T \int_{\mathcal{V}} \mathbf{z}_v(\mathbf{x}) \mathbf{z}_v(\mathbf{x})^T \, d\tau(v) \, \mathbf{a} \\
&= \mathbf{a}^T \mathbf{K} \mathbf{a} \, .
\end{aligned}$$

The third equality is due to the fact that, for all $f \in \mathcal{H}$, we have that $f(x) =$

$\int_{\mathcal{V}} g(v)z(v,x)p(v)dv$ $(\forall x \in \mathcal{X})$ and

$$\|f\|_{\mathcal{H}} = \min_{\left\{g \mid f(x)=\int_{\mathcal{V}} g(v)z(v,x)p(v)dv\right\}} \|g\|_{L_2(d\tau)} .$$

The first inequality, on the other hand, follows from the Cauchy-Schwarz inequality. The bound implies that $\mathbf{f}_x\mathbf{f}_x^T \preceq \mathbf{K}$ and, consequently, we derive $\mathbf{f}_x^T\mathbf{K}^{-1}\mathbf{f}_x \leq 1$. □

Now we are ready to prove Lemma 4.6.

Lemma 4.6. *Suppose that Assumption A.1 holds and that the conditions on sampling measure $\tilde{l}$ from Theorem 4.1 apply as well. If*

$$s \geq 5d_{\tilde{l}} \log \frac{16d_{\mathbf{K}}^{\lambda}}{\delta} ,$$

then for all $\delta \in (0,1)$ and any $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}} \leq 1$, with probability greater than $1-\delta$, the following holds

$$\min_{\beta} \left\{ \frac{1}{n}\|\mathbf{f}_x - \mathbf{Z}_q\beta\|_2^2 + \lambda s\|\beta\|_2^2 \right\} \leq 2\lambda .$$

For the sake of brevity, we will henceforth use $\tilde{\mathcal{H}}$ to denote the hypothesis space corresponding to this optimization problem. Then, the latter bound can be written as:

$$\sup_{\|f\|_{\mathcal{H}}\leq 1} \inf_{\|\tilde{f}\|_{\tilde{\mathcal{H}}}\leq\sqrt{2}} \frac{1}{n}\|\mathbf{f}_x - \tilde{\mathbf{f}}_x\|_2^2 \leq 2\lambda .$$

Proof. For any $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}} \leq 1$, we write the following optimization problem:

$$\frac{1}{n}\|\mathbf{f}_x - \mathbf{Z}_q\beta\|_2^2 + s\lambda\|\beta\|_2^2 . \tag{4.38}$$

The minimizer can be computed as:

$$\begin{aligned}\beta &= \frac{1}{s}(\frac{1}{s}\mathbf{Z}_q^T\mathbf{Z}_q + n\lambda\mathbf{I})^{-1}\mathbf{Z}_q^T\mathbf{f}_x \\ &= \frac{1}{s}\mathbf{Z}_q^T(\frac{1}{s}\mathbf{Z}_q\mathbf{Z}_q^T + n\lambda\mathbf{I})^{-1}\mathbf{f}_x \\ &= \frac{1}{s}\mathbf{Z}_q^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x \ ,\end{aligned}$$

where the second equality follows from the Woodbury inversion lemma.

Substituting β into Eq. (4.38), we transform the first part as

$$\begin{aligned}\frac{1}{n}\|\mathbf{f}_x - \mathbf{Z}_q\beta\|_2^2 &= \frac{1}{n}\|\mathbf{f}_x - \frac{1}{s}\mathbf{Z}_q\mathbf{Z}_q^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x\|_2^2 \\ &= \frac{1}{n}\|\mathbf{f}_x - \tilde{\mathbf{K}}(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x\|_2^2 \\ &= \frac{1}{n}\|n\lambda(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x\|_2^2 \\ &= n\lambda^2\mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-2}\mathbf{f}_x \ .\end{aligned}$$

On the other hand, the second part can be transformed as

$$\begin{aligned}s\lambda\|\beta\|_2^2 &= s\lambda\frac{1}{s^2}\mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{Z}_q\mathbf{Z}_q^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x \\ &= \lambda\mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\tilde{\mathbf{K}}(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x \\ &= \lambda\mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}(\tilde{\mathbf{K}} + n\lambda\mathbf{I})(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x - n\lambda^2\mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-2}\mathbf{f}_x \\ &= \lambda\mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x - n\lambda^2\mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-2}\mathbf{f}_x \ .\end{aligned}$$

Now, summing up the first and the second part, we deduce

$$\begin{aligned}&\frac{1}{n}\|\mathbf{f}_x - \mathbf{Z}_q\beta\|_2^2 + s\lambda\|\beta\|_2^2 = \lambda\mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x \\ &= \lambda\mathbf{f}_x^T(\mathbf{K} + n\lambda\mathbf{I} + \tilde{\mathbf{K}} - \mathbf{K})^{-1}\mathbf{f}_x \\ &= \lambda\mathbf{f}_x^T(\mathbf{K} + n\lambda\mathbf{I})^{-\frac{1}{2}}\left(\mathbf{I} + (\mathbf{K} + n\lambda\mathbf{I})^{-\frac{1}{2}}(\tilde{\mathbf{K}} - \mathbf{K})(\mathbf{K} + n\lambda\mathbf{I})^{-\frac{1}{2}}\right)^{-1}(\mathbf{K} + n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{f}_x \ .\end{aligned}$$

From Lemma 4.10, it follows that when

$$s \geq d_{\tilde{l}}(\frac{1}{\epsilon^2} + \frac{2}{3\epsilon}) \log \frac{16 d_{\mathbf{K}}^{\lambda}}{\delta}$$

then $(\mathbf{K} + n\lambda\mathbf{I})^{-\frac{1}{2}}(\tilde{\mathbf{K}} - \mathbf{K})(\mathbf{K} + n\lambda\mathbf{I})^{-\frac{1}{2}} \succeq -\epsilon\mathbf{I}$.

We can now upper bound the objective function as follows (with $\epsilon = 1/2$)

$$\begin{aligned}\lambda \mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x &\leq \lambda \mathbf{f}_x^T(\mathbf{K} + n\lambda\mathbf{I})^{-\frac{1}{2}}(1-\epsilon)^{-1}(\mathbf{K} + n\lambda\mathbf{I})^{-\frac{1}{2}}\mathbf{f}_x \\ &= (1-\epsilon)^{-1}\lambda \mathbf{f}_x^T(\mathbf{K} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x \leq (1-\epsilon)^{-1}\lambda \mathbf{f}_x^T\mathbf{K}^{-1}\mathbf{f}_x \leq 2\lambda\,,\end{aligned}$$

where in the last inequality we have used Lemma 4.11. Therefore, we prove the first inequality.

For the second claim, we have that

$$\begin{aligned}s\|\beta\|_2^2 &= \mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x - n\lambda \mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-2}\mathbf{f}_x \\ &\leq \mathbf{f}_x^T(\tilde{\mathbf{K}} + n\lambda\mathbf{I})^{-1}\mathbf{f}_x \leq (1-\epsilon)^{-1}\mathbf{f}_x^T\mathbf{K}^{-1}\mathbf{f}_x \leq 2\,.\end{aligned}$$

Hence, the squared norm of our approximated function is bounded by $\|\tilde{f}\|_{\tilde{\mathcal{H}}}^2 \leq s\|\beta\|_2^2 \leq 2$. As such, problem (4.38) can now be written as $\min_\beta (1/n)\|\mathbf{f}_x - \tilde{\mathbf{f}}_x\|_2^2$ subject to $\|\tilde{f}\|_{\tilde{\mathcal{H}}}^2 \leq s\|\beta\|_2^2 \leq 2$, which is equivalent to

$$\inf_{\|\tilde{f}\|_{\tilde{\mathcal{H}}} \leq \sqrt{2}} \frac{1}{n}\|\mathbf{f}_x - \tilde{\mathbf{f}}_x\|_2^2\,,$$

and we have shown that this can be upper bounded by 2λ. Since we are approximating any $f \in \mathcal{H}$ with $\|f\|_{\mathcal{H}} \leq 1$, this can further be written as

$$\sup_{\|f\|_{\mathcal{H}} \leq 1} \inf_{\|\tilde{f}\|_{\tilde{\mathcal{H}}} \leq \sqrt{2}} \frac{1}{n}\|\mathbf{f}_x - \tilde{\mathbf{f}}_x\|_2^2 \leq 2\lambda\,.$$

4.8.3 Proof of Lemma 4.7 □

Lemma 4.7. *Denote the in-sample prediction of $\hat{f}^{\lambda}$ with*

$$\hat{\mathbf{f}}_x^\lambda = [\hat{f}^\lambda(x_1), \ldots, \hat{f}^\lambda(x_n)]^T \tag{4.8}$$

and let $\{v_i\}_{i=1}^s$ *be independent samples selected according to a probability density function* $q(v)$*, which define the feature matrix* $\mathbf{Z}_q$ *and the corresponding RKHS* $\tilde{\mathcal{H}}$*. Let* $\tilde{\beta}^\lambda$ *be the solution to the following optimization problem*

$$\tilde{\beta}^\lambda := \min_\beta \frac{1}{n}\|\hat{\mathbf{f}}_x^\lambda - \mathbf{Z}_q\beta\|_2^2 + \lambda s\|\beta\|_2^2,$$

and denote the in-sample prediction of the resulting hypothesis with $\tilde{\mathbf{f}}_x^\lambda = \mathbf{Z}_q\tilde{\beta}^\lambda$*. Then, we have*

$$\frac{1}{n}\langle Y - \hat{\mathbf{f}}_x^\lambda, \hat{\mathbf{f}}_x^\lambda - \tilde{\mathbf{f}}_x^\lambda\rangle \le \lambda\ .$$

Proof. By definition, $\tilde{\mathbf{f}}_x^\lambda$ has the format as $\tilde{\mathbf{f}}_x^\lambda = \mathbf{Z}_q\tilde{\beta}^\lambda$, where $\tilde{\beta}^\lambda \in \mathbb{R}^s$. In addition, definition of $\tilde{f}^\lambda$ can be reparameterized by the following optimization problem

$$\tilde{\beta}^\lambda := \min_{\tilde{\beta}} \frac{1}{n}\|\hat{\mathbf{f}}_x^\lambda - \mathbf{Z}_q\tilde{\beta}\|_2^2 + s\lambda\|\tilde{\beta}\|\ . \tag{4.39}$$

This gives the closed-form solution of $\tilde{\beta}^\lambda = \frac{1}{s}\mathbf{Z}_q^T(\frac{1}{s}\mathbf{Z}_q\mathbf{Z}_q^T + n\lambda\mathbf{I})^{-1}\hat{\mathbf{f}}_x^\lambda$. As a result, we have

$$\tilde{\mathbf{f}}_x^\lambda = \frac{1}{s}\mathbf{Z}_q\mathbf{Z}_q^T(\frac{1}{s}\mathbf{Z}_q\mathbf{Z}_q^T + n\lambda\mathbf{I})^{-1}\hat{\mathbf{f}}_x^\lambda = \tilde{\mathbf{K}}(\tilde{\mathbf{K}} + n\lambda I)^{-1}\hat{\mathbf{f}}_x^\lambda\ .$$

Now recall $\hat{\mathbf{f}}_x^\lambda$ is the in-sample prediction of the KRR estimator $\hat{f}^\lambda$, so it can be written as $\hat{\mathbf{f}}_x^\lambda = \mathbf{K}(\mathbf{K} + n\lambda I)^{-1}Y$. As a result, we have the following

$$\begin{aligned}
\frac{1}{n}\langle Y - \hat{\mathbf{f}}_x^\lambda, \hat{\mathbf{f}}_x^\lambda - \tilde{\mathbf{f}}_x^\lambda\rangle &= \frac{1}{n}\langle Y - \hat{\mathbf{f}}_x^\lambda, \hat{\mathbf{f}}_x^\lambda - \tilde{\mathbf{K}}(\tilde{\mathbf{K}} + n\lambda I)^{-1}\hat{\mathbf{f}}_x^\lambda\rangle \\
&= \frac{1}{n}\langle Y - \mathbf{K}(\mathbf{K} + n\lambda I)^{-1}Y, (I - \tilde{\mathbf{K}}(\tilde{\mathbf{K}} + n\lambda I)^{-1})\hat{\mathbf{f}}_x^\lambda\rangle \\
&= \frac{1}{n}Y^T(I - \mathbf{K}(\mathbf{K} + n\lambda I)^{-1})(I - \tilde{\mathbf{K}}(\tilde{\mathbf{K}} + n\lambda I)^{-1})\hat{\mathbf{f}}_x^\lambda \\
&\le \frac{1}{n}Y^T(I - \mathbf{K}(\mathbf{K} + n\lambda I)^{-1})\hat{\mathbf{f}}_x^\lambda \qquad (4.40) \\
&= \lambda Y^T(\mathbf{K} + n\lambda I)^{-1}\hat{\mathbf{f}}_x^\lambda \\
&= \lambda Y^T(\mathbf{K} + n\lambda I)^{-1}\mathbf{K}\mathbf{K}^{-1}\hat{\mathbf{f}}_x^\lambda \\
&= \lambda\hat{\mathbf{f}}_x^{\lambda T}\mathbf{K}^{-1}\hat{\mathbf{f}}_x^\lambda \le \lambda\ .
\end{aligned}$$

Note that in Eq. (4.40), we have used the fact that

$$\|I - \tilde{\mathbf{K}}(\tilde{\mathbf{K}} + n\lambda I)^{-1}\|_2 \leq 1 \, .$$

For the last inequality, since $\hat{f}^{\lambda} \in \mathcal{H}$, we employ Lemma 4.11. □

4.8.4 Property of Squared Error Loss

In this section, we state the property of square loss function.

Lemma 4.12. *(Bartlett et al., 2005, Section 5.2) Let l be the squared error loss function and $\mathcal{H}$ a convex and uniformly bounded hypothesis space. Assume that for every probability distribution P in a class of data-generating distributions, there is an $f_{\mathcal{H}} \in \mathcal{H}$ such that $\mathbb{E}[l_{f_{\mathcal{H}}}] = \inf_{f \in \mathcal{H}} \mathbb{E}[l_f]$. Then, there exists a constant $B \geq 1$ such that for all $f \in \mathcal{H}$ and for every probability distribution P*

$$\mathbb{E}[(f - f_{\mathcal{H}})^2] \leq B\mathbb{E}[l_f - l_{f_{\mathcal{H}}}] \, . \tag{4.41}$$

4.8.5 Proof of Lemma 4.8

Lemma 4.8. *Assume $\{x_i, y_i\}_{i=1}^n$ is an independent sample from a probability measure P defined on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y}$ has a bounded range. Let k be a positive definite kernel with the RKHS $\mathcal{H}$ and let $\hat{\lambda}_1 \geq \cdots, \geq \hat{\lambda}_n$ be the eigenvalues of the normalized kernel Gram-matrix. Denote the squared error loss function by $l(f(x), y) = (f(x) - y)^2$ and fix $\delta \in (0, 1)$. If*

$$\hat{\psi}_n(r) = 2Lc_1 \left(\frac{2}{n} \sum_{i=1}^{n} \min\{r, \hat{\lambda}_i\} \right)^{1/2} + \frac{c_2}{n} \log(\frac{1}{\delta}) \, ,$$

then for all $l_f \in l_{\mathcal{H}}$ and $D > 1$, with probability $1 - \delta$,

$$\mathbb{E}[l_f] \leq \frac{D}{D-1} \mathbb{E}_n[l_f] + \frac{6D}{B} \hat{r}^* + \frac{c_3}{n} \log(\frac{1}{\delta}) \, .$$

Moreover, the fixed point $\hat{r}^*$ defined with $\hat{r}^* = \hat{\psi}_n(\hat{r}^*)$ can be upper bounded by

$$\hat{r}^* \leq \min_{0 \leq h \leq n} \left(e_0 \frac{h}{n} + \sqrt{\frac{1}{n} \sum_{i>h} \hat{\lambda}_i} \right) ,$$

where e_0 is a constant.

Proof. It is easy to see that $\mathbb{E}[l_f^2] \leq \mathbb{E}[l_f]$. Hence, we can apply Lemma 4.4 to function class $l_{\mathcal{H}}$ and obtain that for all $l_f \in l_{\mathcal{H}}$

$$\mathbb{E}[l_f] \leq \frac{D}{D-1} \mathbb{E}_n[l_f] + \frac{6D}{B} \hat{r}^* + \frac{c_3}{n} \log(\frac{1}{\delta}) ,$$

as long as there is a sub-root function $\hat{\psi}_n(r)$ such that

$$\hat{\psi}_n(r) \geq c_1 \hat{R}_n \{ l_f \in star(l_{\mathcal{H}}, 0) \mid \mathbb{E}_n[l_f^2] \leq r \} + \frac{c_2}{n} \log(\frac{1}{\delta}) . \tag{4.42}$$

We have previously demonstrated that

$$\begin{aligned} & c_1 \hat{R}_n \{ l_f \in star(l_{\mathcal{H}}, 0) \mid \mathbb{E}_n[l_f^2] \leq r \} + \frac{c_2}{n} \log(\frac{1}{\delta}) \\ \leq & 2 c_1 L \hat{R}_n \left\{ f \in \mathcal{H} \mid \mathbb{E}_n[f^2] \leq e_1 r \right\} + \frac{c_2}{n} \log(\frac{1}{\delta}) \\ \leq & 2 c_1 L \left(\frac{2}{n} \sum_{i=1}^{n} \min \left\{ e_1 r, \hat{\lambda}_i \right\} \right)^{1/2} + \frac{c_2}{n} \log(\frac{1}{\delta}) . \quad \text{(by Lemma 4.5)} \end{aligned} \tag{4.43}$$

Hence, if we choose $\hat{\psi}_n(r)$ to be equal to the right hand side of Eq. (4.43), then $\hat{\psi}_n(r)$ is a sub-root function that satisfies Eq. (4.42). Now, the upper bound on the fixed point $\hat{r}^*$ follows from Corollary 6.7 in Bartlett et al. (2005). □

Chapter 5 Benign Overfitting and Noisy Features

This chapter is based on the following self-contained project Li et al. (2020):

***Zhu Li**, Weijie Su, and Dino Sejdinovic. Benign overfitting and noisy features. arXiv preprint arXiv:2008.02901, 2020*

Modern machine learning models often exhibit the *benign overfitting* phenomenon, which has recently been characterized using the *double descent* curves. In addition to the classical U-shaped learning curve, the learning risk undergoes another descent as we increase the number of parameters beyond a certain threshold. In this chapter, we examine the conditions under which benign overfitting occurs in the random feature (RF) models, i.e., in a two-layer neural network with fixed first layer weights. Adopting a novel view of random features, we show that benign overfitting emerges because of the noise residing in such features. The noise may already exist in the data and propagates to the features, or it may be added by the user to the features directly. Such noise plays an implicit yet crucial regularization role in the phenomenon. In addition, we derive the explicit trade-off between the number of parameters and the prediction accuracy, and demonstrate that overparameterized random feature model can achieve the *optimal learning rate* in the minimax sense. Finally, our results indicate that the learning risk for overparameterized random feature models has multiple, instead of double descent behavior, which is empirically verified in recent works.

5.1 Introduction

Conventional learning wisdom suggests that one should carefully balance between underfitting and overfitting by controlling the capacity of the function class employed (Friedman et al., 2001). For example, in supervised learning with training data $\{(x_i, y_i)\}_{i=1}^n$ drawn from a probability measure P defined on $\mathcal{X} \times \mathcal{Y}$, we learn a predictor f chosen from some hypothesis space $\mathcal{H}$:

$$\mathcal{H} := \left\{ f(x) = \sum_{i=1}^{s} \beta_i z(x, w_i), \quad \beta = [\beta_1, \ldots, \beta_s] \in \mathbb{R}^s \right\},$$

where s represents the number of features in the model. z is some non-linear function, and $w_1, \ldots, w_s$ are the parameters associated with z. In this setting, traditional learning theory states that one should control the capacity of $\mathcal{H}$ either explicitly (e.g., choosing the right quantity s) or implicitly (e.g., early stopping or regularization) in order to prevent overfitting and achieve small generalization error. However, this classical learning theory has been challenged in recent years. In kernel regression, it has been observed that the lowest prediction error often occurs when $\lambda = 0$ (see e.g., Belkin et al. (2018) and Liang et al. (2020a)). In addition, as demonstrated in Zhang et al. (2016), state-of-the-art deep networks are often trained to the *interpolation regime* where estimators perfectly fit the training data (i.e., they fit perfectly even when the noises are present in the labels, which indicates overfitting), yet they still generalize well to new examples. This *benign overfitting* phenomenon is also observed by interpolating kernel machines and deep networks even in the presence of significant label noise. Finally, in a series of insightful papers (Belkin et al., 2019a,b, 2018), it is pointed out that the prediction accuracy for interpolating models often exhibits the *double descent* behavior empirically. In this framework, when the model complexity is small,

i.e., $s < n$, we are in the classical learning regime. As s approaches n, the training risk converges to zero while the true risk grows very large. However, as soon as s passes the threshold of n, the true risk starts to decrease again. Empirically, it is also observed that the minimum true risk achieved as $s \to \infty$ is lower than the minimum risk achieved in the $s < n$ regime.

In this chapter, we study the benign overfitting behavior under the noisy feature setting, i.e., noise that may exist in the covariates x or in the features $z(x, w)$. Classical learning assumes that the training data $\{(x_i, y_i)\}_{i=1}^n$ are independently and identically distributed (i.i.d) samples from joint distribution P on which the true risk is also evaluated. The prediction function is estimated by applying learning algorithm to the observed x (the covariates) or $z(x, w)$ (the features), where x and $z(x, w)$ are assumed to have *no noise*. However, it is well known that noise resides in the input plays an implicit regularization role and hence significantly improve generalization performance (Bishop, 1995, 2006; Goodfellow et al., 2016b). For example, Bishop (1995) show that adding input noise during training can be treated as adding generalized Tikhonov regularizers. An (1996) study the effect of adding noise to the inputs, outputs and weights of the multilayer neural network during training. Their work reveals that input noise is effective in reducing prediction error for both regression and classification problems. Based on these observations, Maaten et al. (2013) propose a novel learning algorithm where they deliberately corrupt training samples with noise that comes from exponential family. Predictors trained with noisy training data are found to substantially outperform those without noise. In addition, Zhuo et al. (2015) design a Bayesian feature noising model to reduce parameter tuning and improve generalization performance. Finally, Liu et al. (2019) propose the Neural Stochastic Differential Equation network which injects random noise during training. The Neural SDE network can achieve better generalization performance and is more robust to adversarial attacks.

Motivated by the implicit regulariation role played by the feature noise, we consider a noisy covariates or noisy features regime in the benign overfitting context. The noise may already exist in the data (e.g., due to imperfect measurement equipment), or it may be added by the user (as we will elaborate later). In this contribution, we show that such noise acts as an implicit regularizer and gives rise to the benign overfitting phenomenon. Bartlett et al. (2020) are the first to propose that noisy feature might lead to benign overfitting under the linear regression setting. They show that if both the feature noise and eigenvalues of the covariance operator exhibit exponential decay, benign overfitting is observed. In Bunea et al. (2020), the feature noise is taken into account under the linear factor regression setting. They demonstrate that under appropriate conditions of the feature noise, the learning risk with interpolation decays, provided that the features and the response are jointly low-dimensional. In contrast, we study the effect of the covariates noises under the more general non-linear feature map setting. Through adopting a novel random feature model with features sampled via Gaussian process formalism, we are not only able to show that the learning risk of the interpolator decays with much more relaxed assumptions (as illustrated in Section 5.3), but also demonstrate that the interpolator can achieve *optimal learning rate* in the minimax sense.

In our framework, similar to the classical learning, we first transform the covariate x to a feature vector $z(x, w)$ through a non-linear random map z (parameterized by w), and then apply regression to the feature vector $z(x, w)$. However, a key difference in our work is that we assume each feature vector $z(x, w)$ to be corroded with some noise ξ. In practice, there are multiple scenarios where the feature will have noise ξ. For instance, the noise might already reside in the covariates x, i.e., $x_{\xi_0} = x + \xi_0$, giving rise to a noisy feature. Alternatively, a noise term ξ could be added deliberately by the user to the feature map $z(x, w)$. For example, in the neural network setting, once we apply the feature map z to the covariates x,

a bias term is typically added on top of $z(w, x)$, which can be treated as the noise term. Finally, in factor regression or principal component regression, $z(x, w)$ is either assumed to have noise or it is estimated by some algorithm (such as principal component analysis), which produces noisy features. In order to consider a general framework and to simplify the notation, we will thereafter write the noisy feature as $z_\xi(x, w) = z(x, w) + \xi$, where $z(x, w)$ is the true feature while ξ represents the noise attached to $z(x, w)$. Regression is then performed on the noisy feature $z_\xi(x, w)$. In this setting, we are interested in how the presence of ξ will affect the generalization performance of the interpolating estimator. Specifically, we make the following contributions:

- *Novel Construction of Random Feature.* We first propose a novel representation of the random feature models based on Mercer's decomposition and Karhunen-Loève expansion (see Section 5.3.1). This new understanding of the random features reduces the non-linear learning problem into a linear regression in the random feature space, and allows us to study the eigenspectrum of the sample covariance matrix, which is the key in analyzing the generalization properties of the interpolator;

- *Implicit Regularization of Feature Noise.* By assuming ξ to follow a normal distribution, Theorem 5.2 establishes a precise relationship between ξ and the excess learning risk. This characterization explains how ξ can act as an implicit regularization and prevent the explosion of the excess learning risk in the overparameterized setting. We remark that when considering linear regression where P has subgaussian distribution, recent works by Bartlett et al. (2020) and Bunea et al. (2020) provide alternatives to our bound in Theorem 5.2. However, our results apply to a more general non-linear regression setting with many relaxed assumptions regarding the data generating distribution and the eigenspectrum of the covariance operator (see

Section 5.3 for more details);

- *Minimax Optimal Estimator.* Our finite sample analysis reveals the detailed trade-off between the number of parameters s and the prediction accuracy represented by the excess learning risk convergence rate. Moreover, under appropriate conditions, we show that the interpolator can achieve the *optimal learning rate* in the minimax sense;

- *Multiple Descent Behavior.* In Corollary 5.1, we extend our analysis of the excess learning risk to the case where ξ follows a subgaussian distribution. Our results demonstrate that benign overfitting will occur as long as ξ decays with s at a prescribed rate, while the shape of the distribution is not a key component in driving the benign overfitting phenomenon. In addition, we analyze the behavior of the excess learning risk bound from Corollary 5.1 and explicitly show that incorporating feature noise into consideration leads to multiple descent of the learning risk. In particular, if the number of parameter s is beyond the $O(n^2)$ order, the learning risk starts to increase again, complementing the current findings of the double descent phenomenon. Our results are empirically verified in recent works by Adlam and Pennington (2020a) and Liang et al. (2020b).

Borrowing from the insight of Mercer's decomposition and Karhunen-Loève expansion, we provide a detailed analysis of the generalization properties of the interpolator. All of our results apply to both the finite sample and the asymptotic case. They are also valid for data of arbitrary dimension. In addition, our results only *impose very weak conditions* on the kernel structure, i.e., as long as its corresponding covariance operator is of trace class (see Assumption A.1). This is fundamentally different from the existing analysis of non-linear feature map (Liao et al., 2020; Mei and Montanari, 2019), which only work for the Gaussian

kernel. More importantly, our results have *no specific assumption* regarding the data generation distribution, which is a significant improvement on existing works (Bartlett et al., 2020; Bunea et al., 2020; Mei and Montanari, 2019), as they often assume the Gaussian or subgaussian data generating distribution.

5.1.1 Related Work

The benign overfitting phenomenon of the interpolating estimator has drawn much interest in the machine learning community recently. For example Liang et al. (2020a) derive the learning risk of the interpolating estimator in kernel ridgeless regression. They show that with certain properties of the kernel matrix and training data, there is an implicit regularization coming from the curvature of the kernel function. A follow-up work (Liang et al., 2020b) further demonstrates that in very high dimension, kernel interpolator exhibits the multiple descent phenomenon. Belkin et al. (2019a) experimentally demonstrate the double descent curve in both linear and non-linear regression cases, and Belkin et al. (2019b) subsequently provide a finite sample analysis of the excess risk for the interpolating estimator in some special settings (where it is assumed that the responses and features are jointly Gaussian). By appealing to random matrix theory, Hastie et al. (2019) obtain the asymptotic behavior of the prediction accuracy in the linear regression setting with correlated features, where sample size n and the covariate dimension d both approach infinity with asymptotic ratio $d/n \to \gamma \in (0, \infty)$. They also study the asymptotic behavior of the variance term in the random non-linear feature regression setting. Recently, by letting n, d and s all grow to infinity such that d/n and n/s remain bounded, Mei and Montanari (2019) derive the double descent curve in the random feature regression setting and obtain the asymptotic behavior. Later, Liao et al. (2020) further extend the analysis of Mei and Montanari (2019) by relaxing the Gaussian assumptions on the data distribution while holding all

other settings the same. A similar asymptotic behavior of the excess learning risk is obtained and the precise characterization of double descent is demonstrated. Finally, in a work that is most related to ours, Bartlett et al. (2020) study the upper and lower bound on the excess risk by assuming that the covariates belong to an infinite-dimensional Hilbert space and follow a sub-gaussian distribution. Through investigating the finite sample learning risk behavior, they explicitly list the conditions for the overfitted linear model to have decaying excess learning risk. Intuitively, for infinite dimensional covariates, the covariance operator spectrum has to decay slowly enough so that the sum of the tail of its eigenvalues is large compared to n. While for finite dimensional covariates, exponential decay for both the eigenspectrum of the covariance operator and the noise attached to the covariates is sufficient to guarantee the consistency of the interpolator.

5.2 Definitions and Notation

5.2.1 Regularised ERM and Kernel Learning

In this section, we provide the necessary definitions and notations as the background. We first revisit two important definitions of KRR and related notations from Chapter 2. In KRR, recall we defined the optimal estimating regression function as

$$f_*(x) = \mathbb{E}(\mathbf{y}|\mathbf{x} = x),$$

where $\mathbf{x}$ and $\mathbf{y}$ are random variables with joint distribution $P(x, y)$. Let $X = [x_1, \ldots, x_n]^T$ and $Y = [y_1, \cdots, y_n]^T$ denote the training inputs and outputs. Given the function $\hat{f}$ estimated based on (X, Y), we defined the notion of excess risk as a

measure of its generalization performance:

$$R_X(\hat{f}) = \mathbb{E}_{\mathbf{x}}\mathbb{E}_{Y|X}[(\hat{f}(\mathbf{x}) - f_*(\mathbf{x}))^2|X]. \tag{5.1}$$

Note that the excess risk is conditional on the training inputs X as emphasized by our notation R_X . However, when the context is clear, we will drop the subscript X for brevity.

We also recall from Chapter 2 the KRR problem:

$$\hat{f}^\lambda := \arg\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\|f\|_{\mathcal{H}}^2.$$

The solution to the KRR problem can be written as $\hat{f}(x) = K(x, X)(\mathbf{K}+n\lambda I)^{-1}Y$, where $K(x, X) = [K(x, x_1), \ldots, K(x, x_n)]^T$ and $\mathbf{K}_{i,j} = K(x_i, x_j)$ is the Gram-matrix.

Random Fourier Features We introduce the RFFs approximation for kernel method in Chapter 2. For translation invariant kernel K, it can be written as

$$k(x, y) = \int \big(cos(w^T x) + b\big)\big(cos(w^T y) + b\big)\, p(b)p(w)dbdw,$$

where $p(b), p(w)$ are certain probability distributions. As a result, if we sample $b_1, \ldots, b_s \sim p(v)$ and $w_1, \ldots, w_s \sim p(w)$, we can approximate the kernel as

$$K(x, y) \approx \frac{1}{s}\sum_{i=1}^{s}\big(cos(w_i^T x) + b_i\big)\big(cos(w_i^T y) + b_i\big) := \tilde{\mathbf{z}}_x^T\tilde{\mathbf{z}}_y.$$

We now let $\|A\|$ be the L_2 norm if A is a vector, or the operator norm if A is an operator. Since $\tilde{\mathbf{z}}_x$ is a random feature vector approximating the kernel k, we use

these features to perform standard (linear) ridge regression on these features.

Definition 5.1. *Given feature vectors and response variables* $\{(\tilde{\mathbf{z}}_{x_i}, y_i)\}_{i=1}^n$*, we define the random feature regression as*

$$\beta_\lambda := \arg\min_{\beta \in \mathbb{R}^s} \ \frac{1}{n}\|Y - \tilde{\mathbf{Z}}\beta\|^2 + \lambda s\|\beta\|^2, \tag{5.2}$$

where $\tilde{\mathbf{Z}} = [\tilde{\mathbf{z}}_{x_1}, \ldots, \tilde{\mathbf{z}}_{x_n}]^T \in \mathbb{R}^{n\times s}$ *is the feature matrix.*

Informally, we can see that instead of using functions in $\mathcal{H}$, RFFs approximation employs functions in the RKHS spanned by the feature matrix $\tilde{\mathbf{Z}}$ to perform learning. As discussed before, we will mainly focus on the interpolator that perfectly fits the training data in this chapter. However, because there are infinitely many estimators as such, we will be working with the one that has the minimum norm defined as following

Definition 5.2. *Given feature vectors and response variables* $\{(\tilde{\mathbf{z}}_{x_i}, y_i)\}_{i=1}^n$*, we define the minimum norm least squares (MNLS) estimator as*

$$\min_{\beta\in\mathbb{R}^s} \|\beta\|^2, \quad \textit{such that } \|\tilde{\mathbf{Z}}\beta - Y\|^2 = \min_{\beta_0} \|\tilde{\mathbf{Z}}\beta_0 - Y\|^2.$$

By the projection theorem, it is easy to see that the closed-form solution of the MNLS estimator is

$$\tilde{\beta} = \tilde{\mathbf{Z}}(\tilde{\mathbf{Z}}\tilde{\mathbf{Z}}^T)^\dagger Y = (\tilde{\mathbf{Z}}^T\tilde{\mathbf{Z}})^\dagger \tilde{\mathbf{Z}}^T Y, \tag{5.3}$$

where $A^\dagger$ denotes the pseudoinverse for matrix A.

Remark 5.1. *We remark that in analyzing the generalization properties of the RFFs interpolator* $\tilde{\beta}$*, a key issue is that the random variables* $w_1, \ldots, w_s$ *are*

embedded in the non-linear feature map $cos(\cdot)$. This non-linear feature embedding makes the analysis of the eigenspectrum of $\tilde{\mathbf{Z}}^T\tilde{\mathbf{Z}}$ significantly hard. In light of this, we propose a novel way to construct random features in Section 5.3.1 to overcome this issue.

5.2.2 Bias-Variance Decomposition

The analysis of the excess learning risk often starts with the bias-variance decomposition. Hence, we present the bias-variance decomposition and introduce some relevant notations here to ease our following discussion. The following lemma gives the bias-variance decomposition. Its proof in Appendix 5.6.1 is a simple non-linear extension of the one in Hastie et al. (2019). Note that instead of working with $\tilde{\mathbf{Z}}$, we will use a different feature matrix in the rest of the chapter. As a result, we state the following bias-variance decomposition with an arbitrary feature matrix $\mathbf{Z}$.

Lemma 5.1. *Let $\tilde{\beta}$ be the MNLS estimator as defined in Eq. (5.3) with feature matrix $\mathbf{Z}$. Let $\tilde{f}(x) = \mathbf{z}_x(\mathbf{W})^T\tilde{\beta}$ be the prediction from the MNLS estimator at a test point x. Define $\tilde{\mathcal{H}}$ as the RKHS spanned by the feature matrix $\mathbf{Z}$ and $f_{\tilde{\mathcal{H}}}$ as the best estimator such that $f_{\tilde{\mathcal{H}}} := \arg\min_{f\in\tilde{\mathcal{H}}} \mathbb{E}_P(f(x)-y)^2$. Since $f_{\tilde{\mathcal{H}}} \in \tilde{\mathcal{H}}$, we let $f_{\tilde{\mathcal{H}}}(x) = \mathbf{z}_x(\mathbf{W})^T\beta_{\tilde{\mathcal{H}}}$ for some $\beta_{\tilde{\mathcal{H}}} \in \mathbb{R}^s$. Define the bias and the variance as*

$$
\begin{aligned}
\mathbf{B}_R &:= \mathbb{E}_{\mathbf{x}}\left[\left(\mathbb{E}_{Y|X}[\tilde{f}(\mathbf{x})] - f_{\tilde{\mathcal{H}}}(\mathbf{x})\right)^2\right] = \mathbb{E}_{\mathbf{x}}\left\|\mathbf{z}_{\mathbf{x}}(\mathbf{W})^T\Pi\beta_{\tilde{\mathcal{H}}}\right\|^2, \\
\mathbf{V}_R &:= \mathbb{E}_{\mathbf{x}}\mathrm{Var}_{Y|X}(\tilde{f}(\mathbf{x})), \\
&= \mathbb{E}_{\mathbf{x}}\left\{\mathbb{E}_{Y|X}\left\|\mathbf{z}_{\mathbf{x}}(\mathbf{W})^T(\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T(Y - f_{\tilde{\mathcal{H}}}(X))\right\|^2\right\},
\end{aligned}
$$

where $f_{\tilde{\mathcal{H}}}(X) = [f_{\tilde{\mathcal{H}}}(x_1), \cdots, f_{\tilde{\mathcal{H}}}(x_n)]^T$. In addition, we define the misspecifica-

tion as

$$\mathbf{M}_R := \mathbb{E}_x \left\{ \mathbf{z}_x(\mathbf{w})^T (\mathbf{Z}^T\mathbf{Z})^\dagger \mathbf{Z}^T \left(f_*(X) - f_{\tilde{\mathcal{H}}}(X) \right) \right\}^2 + \mathbb{E}_x \left(f_*(x) - f_{\tilde{\mathcal{H}}}(x) \right)^2 .$$

Then the following decomposition of the excess risk of $\tilde{\beta}$ *holds*

$$R(\tilde{\beta}) \leq 3(\mathbf{M}_R + \mathbf{B}_R + \mathbf{V}_R).$$

We can see that the excess learning risk is comprised of the bias $\mathbf{B}_R$, the variance $\mathbf{V}_R$ and the misspecification error $\mathbf{M}_R$. While we do not have a thorough understanding on how $\mathbf{M}_R$ evolves with the number of parameters s, we can reasonably assume that $\mathbf{M}_R$ decreases as we increase s. In addition, $\mathbf{M}_R$ becomes zero once $\tilde{\mathcal{H}}$ is large enough to contain f_*. Finally, we note that even in the classical regime where we do not overfit, $\mathbf{M}_R$ is small. Hence, in the overfitting regime, it is reasonable to assume that the excess learning risk is dominated by $\mathbf{B}_R$ and $\mathbf{V}_R$. Therefore, our following analysis will mainly focus on $\mathbf{B}_R$ and $\mathbf{V}_R$ under the assumption that $f_* \in \tilde{\mathcal{H}}$. In other words, we have $\beta_* = \beta_{\tilde{\mathcal{H}}}$.

5.3 Main Results

5.3.1 Warm Up: Random Feature Approximation

Traditionally, RFFs approximation is a simple way to construct a finite-dimensional approximation of an infinite-dimensional kernel introduced by (Rahimi and Recht, 2007). However, in this chapter, we propose a new way of constructing random feature approximation based on Mercer's decomposition and Karhunen-Loève expansion. This novel construction can separate the random feature from the non-linear feature map and hence reduce the non-linear feature regression to a

linear one. To the best of our knowledge, this novel perspective on random feature approximation is new to the literature.

Suppose that we consider kernel regression learning with a continuous kernel $K(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Then by Mercer's Theorem (Steinwart and Christmann, 2008), we have the following decomposition

$$K(x,y) = \sum_{i=1}^{P} \lambda_i e_i(x) e_i(y), \tag{5.4}$$

where $(\lambda_i, e_i)_{i=1}^P$ is the eigensystem corresponding to Mercer's decomposition. $\{e_i\}_{i=1}^P$ is an at most countable orthonormal set of $L_2(dP)$ and $\{\lambda_i\}_{i=1}^P$ is a sequence of non-increasing strictly positive eigenvalues. Based on Mercer's decomposition, we have Karhunen-Loève expansion theorem (Kanagawa et al., 2018, Theorem 4.3) (also see (Steinwart, 2019, Lemma 3.3 and 3.7)) for Gaussian Process with kernel K.

Lemma 5.2. *(Karhunen-Loève Expansion (Kanagawa et al., 2018, Theorem 4.3)) Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a continuous kernel, ν be a finite Borel measure with support on $\mathcal{X}$, and $(e_i, \lambda_i)_{i \in P}$ be that defined in Eq. (5.4). For a Gaussian process $f_K \sim \mathcal{GP}(0, K)$, define*

$$w_i := \lambda_i^{-1/2} \int f(x) e_i(x) d\nu(x), \quad i \in P.$$

Then the following are true

1. *We have*

$$w_i \sim \mathcal{N}(0,1) \quad \textit{and} \quad \mathbb{E}[w_i w_i] = \delta_{ij}, \quad i,j \in P;$$

2. *For all $x \in \mathcal{X}$ and for all finite $J \subset P$, we have*

$$\mathbb{E}\left[\left(f_K(x) - \sum_{i\in J} w_i \lambda_i^{1/2} e_i(x)\right)^2\right] = k(x,x) - \sum_{j\in J} \lambda_j e_j^2(x);$$

3. *If $P = \mathbb{N}$, we have*

$$\lim_{n\to\infty} \mathbb{E}\left[\left(f_K(x) - \sum_{i=1}^{n} w_i \lambda_i^{1/2} e_i(x)\right)^2\right] = 0, \quad x \in \mathcal{X},$$

where the convergence is uniform in $x \in \mathcal{X}$.

Based on Karhunen-Loève expansion and denote w_i's as i.i.d $\sim \mathcal{N}(0,1)$, we can see that a Gaussian process $f_K \sim \mathcal{GP}(0, K)$ has the expansion as

$$f_K(x) = \sum_{i=1}^{P} \lambda_i^{1/2} e_i(x) w_i.$$

Since kernel K can be infinite-dimensional, i.e., $P = \infty$, to ease the discussion, we will define a low-rank kernel k that approximates K by only using the top $p < P$ eigenvalues and eigenvectors of Mercer's decomposition in our analysis, i.e.,

$$k(x,y) = \sum_{i=1}^{p} \lambda_i e_i(x) e_i(y) := V(x)^T D V(y),$$

where we denote $V(x) = [e_1(x), \ldots, e_p(x)]^T$ and $D = [\lambda_1, \ldots, \lambda_p]$.

Now for GP $f \sim \mathcal{GP}(0, k)$, we can express it using Karhunen-Loève expansion

$$f = V^T D^{1/2} \mathbf{w},$$

where $\mathbf{w}$ is a p-dimensional Gaussian random vector with each entry being a

standard normal random variable. From now on, we will use kernel k, and whenever we need to refer to K, we will treat K as the limit of k when $p \to P$.

If we sample $\{\mathbf{w}^{(i)}\}_{i=1}^{s}$ i.i.d $\sim \mathbf{w}$, and let $z(\mathbf{w}^{(i)}, \cdot) = V^T D^{1/2} \mathbf{w}^{(i)}$, $\{z(\mathbf{w}^{(i)}, \cdot)\}_{i=1}^{s}$ are i.i.d sample paths $\sim \mathcal{GP}(0, k)$, such that $\mathbb{E}_{\mathbf{w}}(z(\mathbf{w}^{(i)}, x)) = 0$ and

$$\mathbb{E}_{\mathbf{w}}(z(\mathbf{w}^{(i)}, x) z(\mathbf{w}^{(i)}, y)) = k(x, y).$$

In other words, we use $z(\mathbf{w}^{(i)}, \cdot)$ as the random feature and approximate the kernel as

$$k(x, y) \approx \frac{1}{s} \sum_{i=1}^{s} z(\mathbf{w}^{(i)}, x) z(\mathbf{w}^{(i)}, y).$$

In addition, we let

$$\begin{aligned} \mathbf{z}_x(\mathbf{W}) &= \frac{1}{\sqrt{s}} \mathbf{W}^T D^{1/2} V(x), \\ \mathbf{Z} = [\mathbf{z}_{x_1}(\mathbf{W}), \ldots, \mathbf{z}_{x_n}(\mathbf{W})]^T &= \frac{1}{\sqrt{s}} V(X) D^{1/2} \mathbf{W}, \end{aligned}$$

where

$$\mathbf{W} = [\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(s)}] \in \mathbb{R}^{p \times s}, \quad V(X) = [V(x_1), \ldots, V(x_n)]^T \in \mathbb{R}^{n \times p}.$$

It is easy to verify that $k(x, y) = \mathbb{E}_{\mathbf{w}}[\mathbf{z}_x(\mathbf{W})^T \mathbf{z}_y(\mathbf{W})]$ and $\mathbf{K} = \mathbb{E}_{\mathbf{w}}(\mathbf{Z}\mathbf{Z}^T)$. In addition, the MNLS estimator under the new random feature model is similar to Eq. (5.3) and has the following form

$$\tilde{\beta} = (\mathbf{Z}^T \mathbf{Z})^{\dagger} \mathbf{Z}^T Y.$$

Covariance Operator

Based on the new random feature, we define the following various forms of covariance operator

- Let Σ be the population covariance operator of kernel k with eigenvalue

matrix D [1]

$$\Sigma = \mathbb{E}_{\mathbf{x}} \left\{ D^{1/2} V(\mathbf{x}) V(\mathbf{x})^T D^{1/2} \right\};$$

- Let $\hat{\Sigma}$ be the sample estimate of Σ

$$\hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} D^{\frac{1}{2}} V(x_i) V(x_i)^T D^{\frac{1}{2}};$$

The eigenvalue matrix is denoted to be $\hat{D} = \text{diag}(\hat{\lambda}_1, \ldots, \hat{\lambda}_n)$;

- Let $\hat{\Sigma}^s$ be the random feature approximation of $\hat{\Sigma}$ with s features

$$\hat{\Sigma}^s = \frac{1}{n} \mathbf{Z}^T \mathbf{Z};$$

The eigenvalue matrix is denoted to be $\hat{D}^s = \text{diag}(\hat{\lambda}_1^s, \ldots, \hat{\lambda}_N^s)$.

5.3.2 Benign Overfitting with Noisy Random Features

In this section, we discuss how the behavior of the excess learning risk of the MNLS estimator is affected by the noise in the features. We demonstrate how the new evolution of the excess learning risk leads to benign overfitting and, in particular, to the double descent phenomenon. In the following discussion, we let $P > p > n, s$ without loss of generality. In addition, since we are mainly interested in the overparameterized regime, we let $s \geq n$.

As discussed, we consider the noisy feature setting: $z_\xi(x, w) = z(x, w) + \xi$. We denote the s-dimensional noisy feature as $\mathbf{z}_x^\xi(\mathbf{W}) = \mathbf{z}_x(\mathbf{W}) + \boldsymbol{\xi}$, where $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_s]^T$ with $\xi_1, \ldots, \xi_s$ i.i.d $\sim \xi$. In addition, recall that we define the feature matrix as $\mathbf{Z} = [\mathbf{z}_{x_1}(\mathbf{W}), \ldots, \mathbf{z}_{x_n}(\mathbf{W})]^T$. In the noisy setting, we write the noisy

[1] Note that we use D here because Σ has the same eigenvalues as Mercer's decomposition of kernel k.

feature matrix as $\mathbf{Z}_\xi = \mathbf{Z} + \Xi$, where $\Xi = [\xi_{ij}] \in \mathbb{R}^{n\times s}$ with each ξ_{ij} i.i.d $\sim \xi$. We let $\tilde{\mathcal{H}}_\xi$ to be the RKHS spanned by the noisy feature matrix $\mathbf{Z}_\xi$. Finally, similar to Eq. (5.3), we write the MNLS estimator for $\mathbf{Z}_\xi$ as $\tilde{\beta}_\xi$.

We first list our assumptions (which we will use throughout the chapter)

A.1 The RKHS Condition: Assume $P = \infty$ and $\int_{\mathcal{X}} K(x,x)\, dP_{\mathcal{X}}(x) = C_0$ $(0 < C_0 < \infty)$;

A.2 Label Noise Condition: $\mathcal{X} \subset \mathbb{R}^d$, and $y = f_*(x) + \epsilon$ with $\mathbb{E}(\epsilon) = 0$, and $\text{Var}(\epsilon) = \sigma^2$;

A.3 Feature Noise Condition: $\xi \sim \mathcal{N}(0, \frac{1}{s}\sigma_0^2)$ and $\sigma_0^2 = s^{-\alpha}$, with $\alpha \geq 0$.

Assumption A.1 is a weak condition to ensure K is trace class, i.e., $\text{Tr}(\Sigma_K) < \infty$, and admits Mercer's decomposition. A.1 further ensures that the low rank kernel k has Mercer's decomposition. A.2 is a standard regression assumption. A.3 describes the shape and size of the feature noise ξ. As discussed, the assumption of A.3 is motivated by the well known fact that feature noise can help to regularize (Bartlett et al., 2020; Bishop, 1995; Bunea et al., 2020). We formally discuss the motivation in Section 5.3.3. Note that we need the variance to be $\frac{1}{s}\sigma_0^2$ to ensure that the variance of the feature vector does not explode because $\text{Tr}\{\text{Var}(\mathbf{z}_x^\xi(\mathbf{W}))\} \geq \text{Tr}\{\text{Var}(\boldsymbol{\xi})\} = s\text{Var}(\xi) = \sigma_0^2$.

In the noisy setting, the excess learning risk $R(\tilde{\beta}_\xi)$ admits the bias-variance decomposition similar to Lemma 5.1. We will denote the bias and the variance term as $\mathbf{B}_\xi$ and $\mathbf{V}_\xi$ respectively. We are now ready to present our analysis of the excess learning risk in the noisy feature regime.

Theorem 5.2. *Under Assumptions A.1-3 and suppose we are in the overparameterized regime where $s \geq n$, denote $\Pi_\xi = (\mathbf{Z}_\xi^T\mathbf{Z}_\xi)^\dagger\mathbf{Z}_\xi^T\mathbf{Z}_\xi - I$. Recall that $\mathbf{W}$ is a $p \times s$ matrix with each $\mathbf{W}_{ij}$ i.i.d $\sim \mathcal{N}(0,1)$. Let $\lambda_W = \|\mathbf{W}^T\mathbf{W}\|$ and $a, b, c > 1$*

be some universal constants. Denote $\hat{\lambda}_i^{\xi} = \hat{\lambda}_i + \sigma_0^2/n$. *If we assume that there exists* k^* *defined as*

$$k^* = \min\left\{0 \le k \le n, \sum_{i>k}^{n} \frac{\hat{\lambda}_i^{\xi}}{\hat{\lambda}_{k+1}^{\xi}} \ge \frac{1}{a}n\right\}, \tag{5.5}$$

for any $\delta \in (0,1)$, *with probability at least* $1 - \delta - 6\exp(-n/b) - 5\exp(-n/c)$, *we have*

$$\mathbf{B}_{\xi} \le b\left(\frac{\lambda_W}{s}\|\Sigma\|\sqrt{\log(\frac{14r(\Sigma)}{\delta})/n} + \sigma_0\right)\|\Pi_{\xi}\|^2\|\beta_*^{\xi}\|^2, \tag{5.6}$$

$$\mathbf{V}_{\xi} \le c\sigma^2\mathrm{Tr}(\Sigma)\frac{s}{n^2}, \tag{5.7}$$

where $r(\Sigma) = \mathrm{Tr}(\Sigma)/\|\Sigma\|$.

Theorem 5.2 explains precisely how ξ affects the excess learning risk. In fact, the upper bound of the bias and the variance term indicate that the MNLS estimator $\tilde{\beta}_{\xi}$ asymptotically achieves the optimal prediction accuracy, i.e., the excess risk $\mathbf{B}_{\xi} + \mathbf{V}_{\xi}$ can decay to zero.

Before we analyze the asymptotic behavior of the excess risk, we would like to first discuss our key assumption: Eq. (5.5). There are two scenarios here: $\alpha = 0$ and $\alpha > 0$. We start with the first one. In this case, $\sigma_0^2 = 1$ is a constant. Eq. (5.5) states that there is a k^* such that we have $\sum_{i>k^*}^{n}(\hat{\lambda}_i^{\xi}/\hat{\lambda}_{k^*+1}^{\xi}) \ge \frac{1}{a}n$. This is equivalent to

$$\sum_{i>k^*}^{n}\frac{n\hat{\lambda}_i + \sigma_0^2}{n\hat{\lambda}_{k^*+1} + \sigma_0^2} = \sum_{i>k^*}^{n}\frac{n\hat{\lambda}_i + 1}{n\hat{\lambda}_{k^*+1} + 1} \ge \frac{1}{a}n.$$

Now, if $n\hat{\lambda}_{k^*+1} \le 1$, we have for each $i > k^*$, $(n\hat{\lambda}_i + 1)/(n\hat{\lambda}_{k^*+1} + 1) \ge \frac{1}{2}$. As a result, $\sum_{i>k^*}^{n}(\hat{\lambda}_i^{\xi}/\hat{\lambda}_{k^*+1}^{\xi}) \ge \frac{1}{2}(n - k^*)$. If $k^* \ll n$, there is some universal constant a, such that $\frac{1}{2}(n - k^*) \ge \frac{1}{a}n$. To conclude, if we have both $k^* \ll n$ and

$n\hat{\lambda}_{k^*+1} \leq 1$, $\sum_{i>k^*}^{n}(\hat{\lambda}_i^{\xi}/\hat{\lambda}_{k^*+1}^{\xi}) \geq \frac{1}{a}n$. On the other hand, both $\text{Tr}(\Sigma) < \infty$ and $\text{Tr}(\hat{\Sigma}) < \infty$, implying $\hat{\lambda}_k = \omega_1 k^{-\gamma}$ for some constant ω_1 and $1 < \gamma \leq \infty$. Based on different values of γ, there are three different cases

1. $\gamma = \infty$: $\hat{\Sigma}$ has finite rank, so there is some d such that $\hat{\lambda}_i = 0$ for $i > d$. As such, if we let $k^* = d$, we can easily see that $\sum_{i>k^*}^{n}(\hat{\lambda}_i^{\xi}/\hat{\lambda}_{k^*+1}^{\xi}) = (n-d) \geq \frac{1}{a}n$.

2. $\gamma \propto k$: $\hat{\Sigma}$ has exponential spectrum decay, i.e., $\hat{\lambda}_k = \omega_1 \exp(-k)$. Without loss of generality, we assume $\omega_1 = 1$. Then, if we let $k^* = \log n$, it is easy to see that $n\hat{\lambda}_{k^*+1} = \frac{n}{n+1} \leq 1$. We therefore have $\sum_{i>k^*}^{n}(\hat{\lambda}_i^{\xi}/\hat{\lambda}_{k^*+1}^{\xi}) = \frac{1}{2}(n - \log n) \geq \frac{1}{a}n$.

3. γ is a constant: $\hat{\Sigma}$ has polynomial decay, i.e., $\hat{\lambda}_k = \omega_1 k^{-\gamma}$. Again we assume $\omega_1 = 1$ and if we let $k^* = n^{1/\gamma}$, we have $\hat{\lambda}_{k^*+1} = (n^{1/\gamma}+1)^{-\gamma} \leq (n^{1/\gamma})^{-\gamma} \leq 1$. Therefore, we have $\sum_{i>k^*}^{n}(\hat{\lambda}_i^{\xi}/\hat{\lambda}_{k^*+1}^{\xi}) \leq \frac{1}{2}(n - n^{1/\gamma}) \geq \frac{1}{a}n$.

The analysis in the second scenario where $\alpha > 0$ is similar to the first one. The key is that there exists $k^* \ll n$ such that $n\hat{\lambda}_{k^*+1} \leq \sigma_0^2$. The difference here is that σ_0^2 decays with s at rate $\alpha > 0$. However, if we control the decay rate α such that $\alpha \ll \gamma$, we can guarantee the existence of k^*. In summary, as long as $\alpha \ll \gamma$, we can find a k^* such that Eq. (5.5) holds.

We are now ready to analyze the asymptotic behavior of the excess risk. We start with $\mathbf{V}_{\xi}$ where we can see that the variance $\mathbf{V}_{\xi}$ is governed by s/n^2. Thus, if we let $s = o(n^2)$, we have $\lim_{n\to\infty} \mathbf{V}_{\xi} = 0$. For the bias $\mathbf{B}_{\xi}$, σ_0 and σ_0^2 decay to zero as long as $\alpha > 0$. In addition, it is easy to see that $\lambda_W = O(p)$. Consequently, if we have $\lim_{n\to} \left(p\sqrt{\frac{1}{n}}\right)/s = 0$, $\lim_{n\to\infty} \mathbf{B}_{\xi} = 0$.

Therefore, the following two conditions are required to ensure the convergence of the excess learning risk

- $\lim_{n\to\infty} \frac{p}{s}\sqrt{\frac{1}{n}} = 0$;
- $s = o(n^2)$.

We can see that if we let $s = n^{\gamma_0}$ for some $\gamma_0 \in (1, 2)$, $s \gg n$ since $\lim_{n\to\infty} s/n = \infty$. In other words, even in the heavily overparameterized model, the excess learning risk of the MNLS estimator $\tilde{\beta}_\xi$ converges, since both $\mathbf{B}_\xi$ and $\mathbf{V}_\xi$ converge to zero. However, our theorem indicates that once s is beyond the order of n^2, the variance starts to increase again. This result is aligned with the recent study by Adlam and Pennington (2020a), where the excess risk is found to increase if s is close to the order of n^2.

Optimality We remark that Theorem 5.2 not only establishes the consistency of the MNLS estimator under the noisy feature setting but also provides the trade-off between the number of parameters s and the prediction accuracy (as represented by the excess learning risk convergence rate). In particular, it demonstrates that the interpolator is able to achieve the minimax optimal learning rate.

The minimax optimal learning rate for linear regression and nonparametric regression such as kernel ridge regression is extensively studied in literature. In linear regression with number of feature s, when the optimal regression function parameter is constrained to lie in some ball of $\mathbb{R}^s$, Tsybakov (2003) show that the minimax optimal learning rate is $O(s/n)$. Later, Mourtada (2019) extend the analysis to the whole linear function class and demonstrate the minimax optimal learning rate of $O(s/n)$. By leveraging the *statistical levearge scores* of the sample, they further show that the minimax optimal learning rate has a lower bound of $O(s/(n-s+1))$. In kernel ridge regression, Caponnetto and De Vito (2007) shows that the optimal learning rate for kernel nonparametric learning in the minimax sense is $O(n^{-1/2})$. Furthermore, Rudi and Rosasco (2017) and Li et al. (2019b)

s	p	α	LEARNING RATE
$O(n^{\frac{3}{2}})$	$O(n^{\frac{3}{2}})$	$\alpha = 1$	$O(\frac{1}{\sqrt{n}})$
$O(n \log n)$	$O(\sqrt{n} \log^2 n)$	$\alpha = 2$	$O(\frac{\log n}{n})$

Table 5.1: The trade-off between the learning rate and the number of parameters.

demonstrate the same minimax optimal learning rate in the random Fourier feature learning framework.

We now look at the excess learning risk of the MNLS estimator, which converges at $O(\frac{p}{s}\sqrt{\frac{1}{n}} + s^{-\frac{\alpha}{2}} + \frac{s}{n^2})$ rate (note that we use $O(p)$ to represent the order of λ_W and omit the $r(\Sigma)$ term because we often have $r(\Sigma) \ll n$). By letting $s = O(n^{\frac{3}{2}})$ and $\alpha = 1$, we can see that even in this heavily overparameterized regime, as long as p/s remains constant, the excess learning risk $R(\tilde{\beta})$ converges at the optimal $O(n^{-\frac{1}{2}})$ rate in the minimax sense. In addition, if the effective dimension of the kernel is small in the sense that $p = O(\sqrt{n} \log n)$, and if we let $s = n \log n$ and $\alpha = 2$, we can achieve an even faster learning rate at $O((\log n)/n)$. Note that the fast learning rate is also achieved in the heavily overparameterized settings because $\lim_{n\to\infty} s/n = \infty$. We have summarized our findings in Table 5.1.

Comparison to Existing Finite Sample Bounds Benign overfitting has attracted much research interest recently. Currently, there are two main lines of research addressing the finite sample bounds for the MNLS estimator. The first line (Bartlett et al., 2020; Bunea et al., 2020) assumes the linear regression setting, while the second line (Liang et al., 2020a,b) pursues the consistency of non-regularized kernel regression estimators, assuming high data dimension: $d \asymp n^{\iota}, \iota > 0$.

In the linear regression setting, the work of Bartlett et al. (2020) relates most closely to our results. Under the assumption that $\mathbf{x}$ and $\mathbf{y}$ follow a subgaussian distribution, Bartlett et al. (2020) provide the first finite sample bounds on the consistency of the

MNLS estimator. Specifically, in the finite dimensional case, if a small amount of noise is added into the covariates $\mathbf{x}$ where both the noise and the eigenspectrum of Σ decay exponentially with n, benign overfitting is observed. A noisy feature linear regression is also considered in Bunea et al. (2020) under the factor regression models. By assuming that the covariates have a low dimension representation, Bunea et al. (2020) obtain an improved convergence rate than that in Bartlett et al. (2020). They also demonstrate that their finite sample bound can apply to a more general setting. The consistency proof relies on the subgaussian property of $P(x, y)$. Moreover, Liang et al. (2020a,b) consider the regression with no regularization in the kernel regression setting and provide a finite sample analysis of the kernel ridgeless estimator. They are able to demonstrate the consistency of the kernel ridgeless estimator as long as the dimension d of the data is sufficiently high, i.e., $d \asymp n^{\iota}$ for $\iota > 0$.

In contrast, our work relaxes the subgaussian assumption on the data generating distribution, as the results impose no specific assumption on $P(x, y)$. The requirement of the exponential decay of the noise and eigenspectrum is also released. Our results demonstrate that as long as the decay rate of the noise α is smaller than that of the eigenspectrum γ, we would be able to observe benign overfitting. Hence, our results apply to much more general scenarios where the eigenspectrum is allowed to decay polynomially. Most importantly, our results not only prove the consistency of the MNLS estimator, but also provide a detailed trade-off between the number of parameters s and the prediction accuracy in Theorem 5.2. As discussed before, the MNLS estimator can obtain the minimax optimal learning rate in certain cases, which is the first result that demonstrates minimax optimality for overparameteried models. Finally, we remark that our results are valid for data with arbitrary dimension and therefore can be applied to both low and high dimensional settings.

5.3.3 Motivation

As discussed before, adding the covariates or feature noise is well known to help improve prediction performance (see e.g. An (1996); Bishop (1995, 2006); Goodfellow et al. (2016a)). Motivated by this, we provide a simple mathematical analysis that demonstrates the regularization effect of covariates and feature noise. The results arise from analyzing the excess risk of the MNLS estimator in the noiseless version, i.e., $\tilde{\beta}$ in Eq. (5.3), and serve as another motivation for considering the noisy feature $z_\xi(x, w)$. Proposition 5.1 establishes nearly matching upper and lower bounds for the excess risk of $\tilde{\beta}$. The proof is listed in Section 5.4.1.

Proposition 5.1. *Consider the regression problem Eq. (5.2) with feature matrix* $\mathbf{Z}$ *and suppose we are in the overparameterized regime where* $s \geq n$. *Let* $a, b, c, c' > 1$ *be some universal constants and assume* $s \geq n$. *Let* $\delta \in (0, 1)$ *and denote*

$$k^* = \min\left\{0 \leq k \leq n, \frac{\sum_{i>k}^n \hat{\lambda}_i}{\hat{\lambda}_{k+1}} \geq \frac{1}{a}n\right\}.$$

Then with probability greater than $1 - \delta - 2\exp(-n/c)$, *we have*

$$\begin{aligned} R(\tilde{\beta}) &= \mathbf{B}_R + \mathbf{V}_R, \\ &\leq b\frac{\lambda_W}{s}\|\Pi\|^2\|\beta_*\|^2\|\Sigma\|\sqrt{\log\left(\frac{14r(\Sigma)}{\delta}\right)/n} + c\sigma^2\frac{s}{n}\frac{\mathrm{Tr}(\Sigma)}{\sum_{i>k^*}^n \hat{\lambda}_i}. \end{aligned} \tag{5.8}$$

We can also lower bound the risk with probability greater than $1 - 2\exp(-n/c')$

$$R(\tilde{\beta}) \geq c'\sigma^2\frac{s}{n}\frac{\mathrm{Tr}(\Sigma)}{\sum_{i>k^*}^n \hat{\lambda}_i}. \tag{5.9}$$

Inspecting Eq. (5.8), we can see that the bias term $\mathbf{B}_R$ decays as long as

$$\lim_{p,s\to\infty} \left(p\sqrt{\frac{1}{n}} \right) / s = 0,$$

since $\lambda_W = O(p)$. Therefore, if the variance term decays, $R(\tilde{\beta})$ decreases to zero. As a result, Proposition 5.1 states that we need the following conditions for $\tilde{\beta}$ to obtain optimal prediction accuracy

1. The covariance operator is of trace-class;
2. The sum of the tail eigenvalues of $\hat{\Sigma}$ is on the order of N, i.e., there exists a k^* such that $\sum_{i>k^*}^{n} \hat{\lambda}_i = \Theta(n)$ and $\lim_{s,n\to\infty} \frac{s}{n} \frac{1}{\sum_{i>k^*}^{n} \hat{\lambda}_i} = 0$.

The first condition is a standard requirement for a typical learning problem, and the second condition states that we need s to be of order $o(n^2)$. While these two conditions seem reasonable individually, the second condition appears to be at odds with the first one. Namely, according to the classical concentration inequality, $\|\Sigma - \hat{\Sigma}\| \to 0$ as $n \to \infty$. We thus have $\sum_{i>k}^{n} \hat{\lambda}_i \leq \mathrm{Tr}(\hat{\Sigma})$ which is finite and does not grow with n. However, traditional concentration theory requires that the observed samples $\{x_i\}_{i=1}^{n}$ are i.i.d from the marginal distribution $P_{\mathbf{x}}(x)$. $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} D^{1/2} V_{x_i} V_{x_i}^T D^{1/2}$ is simply an empirical estimate of $\Sigma = \int D^{1/2} V_x V_x^T D^{1/2} dP(x)$. However, if the feature $z(x, w)$ is corroded with noise ξ, the noise will distort the behavior of $\hat{\Sigma}$. We qualitatively discuss the effect of ξ below to provide an intuition on how benign overfitting arises.

Recall the definitions of Σ, $\hat{\Sigma}$ and $\hat{\Sigma}^s$ in Section 5.3.1. In the noiseless setting, $\hat{\Sigma}^s = \frac{1}{n}\mathbf{Z}^T\mathbf{Z} \in \mathbb{R}^{s\times s}$. As $s \to \infty$, $\hat{\Sigma}_s \to \hat{\Sigma}$, implying $\hat{D}^s \to \hat{D}$. In the noisy setting, supposing the feature matrix is corroded with some i.i.d noise: $\mathbf{Z}_\xi = \mathbf{Z} + \Xi$,

the covariance matrix now is

$$\hat{\Sigma}_{\xi}^{s} = \frac{1}{n}\mathbf{Z}_{\xi}^{T}\mathbf{Z}_{\xi} = \frac{1}{n}(\mathbf{Z}^{T}\mathbf{Z} + \Xi^{T}\mathbf{Z} + \mathbf{Z}^{T}\Xi + \Xi^{T}\Xi).$$

Denote $\hat{D}_{\xi}^{s}$ as the eigenvalues of $\hat{\Sigma}_{\xi}^{s}$. As $s \to \infty$, we approximately have $\hat{D}_{\xi}^{s} \approx \text{diag}(\hat{\lambda}_1^s + \sigma_0^2, \ldots, \hat{\lambda}_n^s + \sigma_0^2)$. Since $\hat{\lambda}_i^s$ decays, there will be a $k^* < n, s$ such that $\hat{\lambda}_i^s > \sigma_0^2, \forall i < k^*$. However, $\hat{\lambda}_i^s$ is on the same scale of σ_0^2 for all $i > k^*$, in the sense that

$$\frac{\hat{\lambda}_i^s + \sigma_0^2}{\hat{\lambda}_j^s + \sigma_0^2} \approx \Theta(1), \text{for all } i, j > k^*.$$

Since $\hat{\Sigma}_{\xi}^{s}$ has at most n eigenvalues, summing up the tails gives

$$\frac{\sum_{i>k^*}^{n}(\hat{\lambda}_i^s + \sigma_0^2)}{\hat{\lambda}_{k^*+1}^s + \sigma_0^2} \approx \Theta(n).$$

This condition indicates that the sum of the tail eigenvalues of the covariance matrix $\hat{\Sigma}_{\xi}^{s}$ is of the order of n, leading to the decay of $R(\tilde{\beta})$. This analysis further motivates us to quantify how exactly the noise ξ affects the behavior of the excess learning risk in Theorem 5.2. Below we provide a sketch of the proof for Theorem 5.2.

Proof. (Sketch of Proof for Theorem 5.2) The proof starts with the Bias-Variance decomposition. We employ the noisy feature version of Lemma 5.1, where the excess learning risk is decomposed to the bias term $\mathbf{B}_{\xi}$ and the variance term $\mathbf{V}_{\xi}$. While the treatment of $\mathbf{B}_{\xi}$ is relatively standard, the technical part is how to analyze $\mathbf{V}_{\xi}$. The key is to express $\mathbf{V}_{\xi}$ as a sum of the outer product of random vectors with each entry being i.i.d standard Gaussian random variables. After that, we apply concentration inequalities to the outer products, which gives us the desired results. For detailed derivation, please refer to Section 5.4.2. □

5.3.4 Benign Overfitting with Subgaussian Noisy Features

Theorem 5.2 demonstrates that if ξ is Gaussian with decaying variance, benign overfitting can be observed. However, Gaussian noise is sometimes a strong assumption. A close investigation of Theorem 5.2 indicates that the key driving force of benign overfitting is that σ_0^2 decays with s, rather than the shape of ξ. Hence, we conjecture that benign overfitting will occur even if we have non-Gaussian distributions. It transpires that a simple extension of Theorem 5.2 would allow us to generalize our results to subgaussian noise. Thus, we modify Assumption A.3 to

A.3′ Feature Noise Condition: ξ is a subgaussian in the sense that $\xi = \frac{\sigma_0^2}{s} u$, where $\sigma_0^2 = s^{-\alpha}$ and u is zero mean, unit variance and σ_u^2-subgaussian, i.e.,

$$\mathbb{E}(\exp(tu)) \leq \exp\left(\frac{\sigma_u^2}{2} t^2\right).$$

Our results below confirm that benign overfitting can indeed be observed in the subgaussian noise setting.

Corollary 5.1. *Under A.1-3′, suppose $s \geq n$. Let $a, b, c > 1$ be some universal constants. Recall that $\hat{\lambda}_i^{\xi} = \hat{\lambda}_i + \sigma_0^2/n$. If we assume that there exists k^* defined as*

$$k^* = \min\left\{0 \leq k \leq n, \sum_{i>k}^{n} \frac{\hat{\lambda}_i^{\xi}}{\hat{\lambda}_{k+1}^{\xi}} \geq \frac{1}{a} n\right\}, \tag{5.10}$$

for any $\delta \in (0,1)$, with probability at least $1 - \delta - 6\exp(-n/b) - 5\exp(-n/c)$, we have

$$\mathbf{B}_{\xi} \leq b\left(\frac{\lambda_W}{s}\|\Sigma\|\sqrt{\log(\frac{14r(\Sigma)}{\delta})/n} + \sigma_0\right)\|\Pi_{\xi}\|^2\|\beta_*^{\xi}\|^2, \tag{5.11}$$

$$\mathbf{V}_{\xi} \leq c\sigma^2 \mathrm{Tr}(\Sigma)\frac{s}{n^2}. \tag{5.12}$$

The behavior of the excess learning risk in the subgaussian case is almost identical to the Gaussian case up to some constant. As discussed in the Gaussian case, the noise term leads to the asymptotical decay of the learning risk. The results verify our conjecture that as long as the noise ξ decays with s, we will observe benign overfitting. Corollary 5.1 further confirms that the noise ξ in the covariate or the feature vector can serve as an implicit regularizer to prevent overfitting.

5.3.5 The Double Descent Phenomenon

The classical U-shape learning curve (Friedman et al., 2001, Figure 2.11) has been greatly challenged recently (Belkin et al., 2019a,b, 2018), because empirically it is often observed that the relationship between the prediction accuracy and the complexity of the learning machine exhibits the double descent phenomenon. For instance, in Figure 5.1 (Belkin et al., 2019a), the double descent curve states that when we increase the capacity of the hypothesis space $\mathcal{H}$, the excess learning risk first decreases, but then starts to increase as we keep increasing the model complexity. The excess learning risk increases to the maximum (or potentially diverges to infinity) at some interpolation threshold. After that, as we keep increasing the complexity of $\mathcal{H}$, the excess learning risk decreases either to a global minimum or vanishes to zero. Overall, the behavior of the excess learning risk regarding the model complexity forms a double descent curve.

The double descent phenomenon has attracted much research interest recently, although a conclusive explanation has not been proposed. In this section, we try to explicate how double descent occurs as a result of noisy features through analyzing the behavior of the excess learning risk in Corollary 5.1.

Before delivering our results, we first state our notation and assumptions. Recall Lemma 5.1 where we have decomposed the excess learning risk into the misspeci-

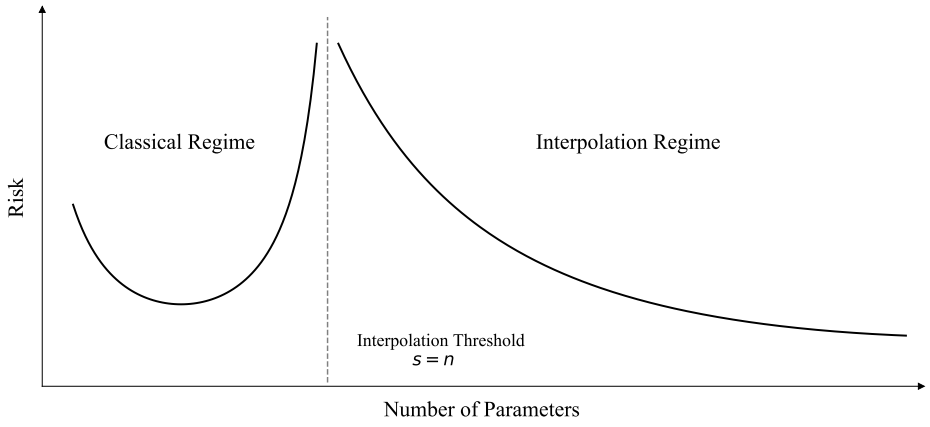

Figure 5.1: The double descent curve.

fication error $\mathbf{M}_R$ (or $\mathbf{M}_\xi$ in the noisy feature setting), the bias $\mathbf{B}_R$ (or $\mathbf{B}_\xi$) and the variance $\mathbf{V}_R$ (or $\mathbf{V}_\xi$). Corollary 5.1 and the double descent are connected through analyzing these errors in the noisy feature setting. We now demonstrate the finite sample behavior of the excess learning risk from Corollary 5.1.

Corollary 5.2. *Under Assumptions A.*1-3′*, the behavior of the excess learning risk* $R(\tilde{\beta}_\xi)$ *can be described as following*

a. If $s = n$

$$\mathbf{B}_\xi = O(\frac{p}{n^{3/2}}) + O(n^{-\alpha}), \mathbf{V}_\xi = O(n^{-1});$$

b. If $s = o(n^{\gamma_1})$, $\gamma_1 \in (1, 2)$

$$\mathbf{B}_\xi = O(\frac{p}{s} n^{-\frac{1}{2}}) + O(n^{-\alpha}), \mathbf{V}_\xi = O(n^{(\gamma_1 - 2)});$$

c. If $s = \Theta(n^{\gamma_2})$, $\gamma_2 > 2$

$$\mathbf{V}_\xi = \Theta(n^{(\gamma_2 - 2)}).$$

Corollary 5.2 describes the precise behavior of the excess learning risk in the overparameterized regime $s \geq n$. Before we give a detailed discussion on that, we first qualitatively discuss the behavior of the excess learning risk in the underparameterized regime, i.e., the first U-shape in Figure 5.1. When $s < n$, Corollary 5.1 indicates that $\mathbf{B}_\xi = 0$ by Lemma 5.9. However, Corollary 5.1 assumes that we are in the realizable case ($f_* \in \tilde{\mathcal{H}}$). When s is small, this is unlikely to happen. As a result, we have the misspecification error $\mathbf{M}_\xi$. In other words, in the finite sample case where $s < n$, the excess learning risk is governed by the misspecification error and the variance. Intuitively, we can see that $\mathbf{M}_\xi$ decreases as we increase s, and typically, when s is small, $\mathbf{M}_\xi$ dominates the excess learning risk. As we increase s, $\mathbf{M}_\xi$ decreases and $\mathbf{V}_\xi$ increases up to some point, where $\mathbf{V}_\xi$ starts to dominate the excess learning risk. Therefore, we will observe that the excess learning risk initially decreases with s and after some point, it starts to increase with s, which forms the classical U-shape curve.

When we approach the interpolation threshold where $s = n$, Corollary 5.2 shows that the bias term $\mathbf{B}_\xi$ starts to dominate the excess learning risk by the term $O\left(p/n^{3/2}\right)$, because $n, s \ll p$ at this point. In particular, if we use kernel K where $P = \infty$, the bias $\mathbf{B}_R$ diverges to infinity.

Furthermore, if we keep increasing s so that it passes the interpolation threshold, the excess learning risk is now dominated by $\mathbf{B}_\xi$ and $\mathbf{V}_\xi$, since the misspecification error becomes negligible. As discussed earlier, if $p/(s\sqrt{n})$ converges to zero, and $s = o(n^2)$, both terms vanish to zero asymptotically, driving the learning risk to its global minimum. In particular, if $s = n^\gamma$ with $\gamma \in (1, 2)$, the overfitted model with $s \gg n$ can still have excess learning risk to converge. In addition, if we keep increase s such that it is beyond the order of n^2, we can see that the variance starts to increase again. Overall, Corollary 5.2 gives us the precise description of the double descent curve up to the point where s is within the n^2 order. Our theoretical

results are empirically verified by recent findings from Adlam and Pennington (2020a).

In Figure 5.2, we give a sample path of how our upper bound evolves with the number of features s. We can see that it closely resembles the double descent curve in Figure 5.1 from (Belkin et al., 2019a).

5.4 Proofs

In the proof, we use $b_1, b_2, \cdots > 1$ to denote universal constants in the theorem statement, and we use $c_1, c_2, \cdots > 1$ to denote universal constants in the proof of the theorems.

5.4.1 Proof of Proposition 5.1

The proof of Proposition 5.1 starts with the analysis of each term in the bias-

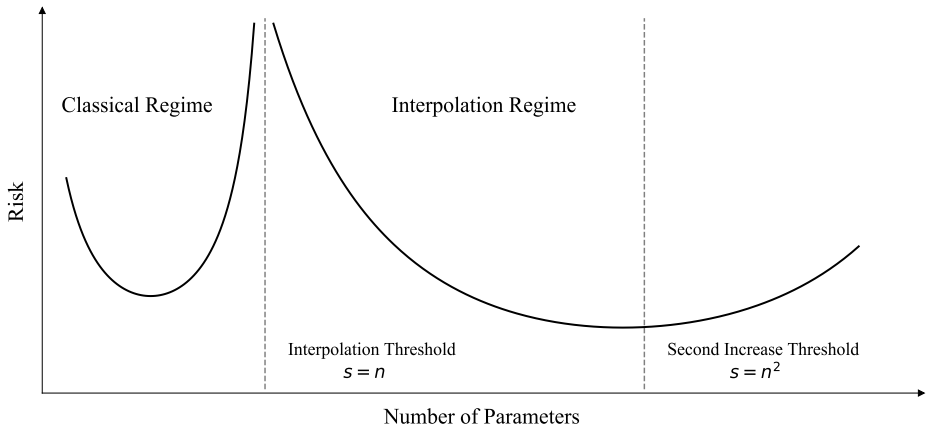

Figure 5.2: The evolution of the excess learning risk from Corollary 5.2. Note that our characterization has the property where the excess learning risk starts to grow beyond the n^2 regime, which is aligned with the recent empirical observation of the triple-double descent learning curve from Adlam and Pennington (2020a).

Upper Bound of Bias

The following lemma gives the upper bound on the bias term $\mathbf{B}_R$.

Lemma 5.3. *The bias can be upper bounded as*

$$\int_{\mathcal{X}} \|\mathbf{z}_x(\mathbf{W})^T \Pi \beta_*\|^2 dP(x) \leq \frac{\lambda_W}{s} \|\Pi\|^2 \, \|\Sigma - \hat{\Sigma}\| \, \|\beta_*\|^2.$$

Proof. Using the property of pseudoinverse yields

$$\mathbf{Z}^T \mathbf{Z} \Pi = \mathbf{Z}^T \mathbf{Z} (I - (\mathbf{Z}^T \mathbf{Z})^\dagger \mathbf{Z}^T \mathbf{Z}) = 0.$$

Hence, we have

$$\begin{aligned}
\mathbf{B}_R &= \mathbb{E}_x \left\{ \beta_* \Pi \mathbf{z}_x(\mathbf{W}) \mathbf{z}_x(\mathbf{W})^T \Pi \beta_* \right\} = \beta_* \Pi \left(\frac{1}{s} \mathbf{W}^T \Sigma \mathbf{W} \right) \Pi \beta_*, \\
&= \beta_* \Pi \left(\frac{1}{s} \mathbf{W}^T \Sigma \mathbf{W} - \frac{1}{n} \mathbf{Z}^T \mathbf{Z} \right) \Pi \beta_* = \beta_* \Pi \left(\frac{1}{s} \mathbf{W}^T \Sigma \mathbf{W} - \frac{1}{s} \mathbf{W}^T \hat{\Sigma} \mathbf{W} \right) \Pi \beta_*, \\
&= \frac{1}{s} \beta_* \Pi \left(\mathbf{W}^T (\Sigma - \hat{\Sigma}) \mathbf{W} \right) \Pi \beta_* \leq \frac{1}{s} \|\beta_*\|^2 \|\Pi\|^2 \|\Sigma - \hat{\Sigma}\| \|\mathbf{W}^T \mathbf{W}\|, \\
&= \frac{\lambda_W}{s} \|\Pi\|^2 \|\Sigma - \hat{\Sigma}\| \|\beta_*\|^2.
\end{aligned}$$

□

The following lemma provides an upper bound for $\|\Sigma - \hat{\Sigma}\|$.

Lemma 5.4. *Let Σ and $\hat{\Sigma}$ denote the covariance operator and sample covariance operator respectively, then for $\delta \in (0, 1)$, we have with probability at least $1 - \delta$,*

$$\|\Sigma - \hat{\Sigma}\| \leq c \|\Sigma\| \sqrt{\log(\frac{14 r(\Sigma)}{\delta})/n},$$

where $r(\Sigma) = \frac{\mathrm{Tr}(\Sigma)}{\|\Sigma\|}$ and c_0 is some universal constant.

Proof. For $\Sigma - \hat{\Sigma}$, we first notice that[2]

[2]As $p \to \infty$, both Σ and $\hat{\Sigma}$ are operators. Hence, we have misused the notation $A^T B$ to represent inner product between operator A and B. But our analysis is not affected by this notation.

$$\begin{aligned}\Sigma - \hat{\Sigma} &= \Sigma - \frac{1}{n} D^{1/2} V(X)^T V(X) D^{1/2} = \Sigma - \frac{1}{n}\sum_{i=1}^{n} D^{1/2} V_{x_i} V_{x_i}^T D^{1/2}, \\ &:= \Sigma - \frac{1}{n}\sum_{i=1}^{n} \phi(x_i)\phi(x_i)^T = \sum_{i=1}^{n} \frac{1}{n}(\Sigma - \phi(x_i)\phi(x_i)^T), \\ &:= \sum_{i=1}^{n} R_i.\end{aligned}$$

Now we trivially have $\mathbb{E}(R_i) = 0$. In addition,

$$\begin{aligned}R_i &\preccurlyeq \frac{1}{n}\Sigma \preccurlyeq \frac{\|\Sigma\|}{n} I = \frac{\|\Sigma\|}{n} I, \\ R_i &\succcurlyeq -\frac{1}{n}\phi(x_i)\phi(x_i)^T \succcurlyeq -\frac{1}{n}\|\phi(x_i)\|^2 I \succcurlyeq -\frac{\|\Sigma\|}{n} I.\end{aligned}$$

As a result, we have $\|R_i\| \leq \frac{\lambda_1}{n}$.

$$\begin{aligned}\mathbb{E}(R_i^2) &= \frac{1}{n^2}\mathbb{E}\left(\Sigma - \phi(x_i)\phi(x_i)^T\right)^2, \\ &= \frac{1}{n^2}\left\{\mathbb{E}\left(\phi(x_i)\phi(x_i)^T\right)^2 - 2\Sigma\phi(x_i)\phi(x_i)^T + \Sigma^2\right\}, \\ &= \frac{1}{n^2}\left\{\mathbb{E}(\phi(x_i)\phi(x_i)^T)^2 - \Sigma^2\right\} \preceq \frac{1}{n^2}\mathbb{E}\left(\phi(x_i)\phi(x_i)^T\right)^2, \\ &= \frac{1}{n^2}\mathbb{E}\left(\phi(x_i)\phi(x_i)^T\phi(x_i)\phi(x_i)^T\right) \preceq \frac{\|\Sigma\|}{n^2}\mathbb{E}(\phi(x_i)\phi(x_i)^T), \\ &= \frac{\|\Sigma\|}{n^2}\Sigma \preceq \frac{\|\Sigma\|^2}{n^2} I.\end{aligned}$$

Thus we have, $\sum_{i=1}^{n}\mathbb{E}(R_i^2) \preccurlyeq \frac{\|\Sigma\|^2}{n} I$. Hence, we have $\|\sum_{i=1}^{n}\mathbb{E}(R_i^2)\| \leq \frac{\|\Sigma\|^2}{n}$. Using (Minsker, 2017, Operator version of Theorem 3.1), for any $\delta \in (0,1)$, with probability greater than $1-\delta$, we have

$$\|\Sigma - \hat{\Sigma}\| \leq c\|\Sigma\|\sqrt{\log(\frac{14r(\Sigma)}{\delta})/n}.$$

□

Upper Bound of Variance:

The upper bound of the variance term $\mathbf{V}_R$ is a bit involving, we split it into several steps. We first note that some simple algebra yield a basic upper bound:

$$
\begin{aligned}
\mathbf{V}_R &= \int_{\mathcal{X}} \mathbb{E}_{\boldsymbol{\epsilon}} \left\{ \mathbf{z}_x(\mathbf{W})^T (\mathbf{Z}^T\mathbf{Z})^\dagger \mathbf{Z}^T \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T \mathbf{Z} (\mathbf{Z}^T\mathbf{Z})^\dagger \mathbf{z}_x(\mathbf{W}) \right\}, \\
&\leq \sigma^2 \mathbb{E}_x \left(\mathbf{z}_x(\mathbf{W})^T (\mathbf{Z}^T\mathbf{Z})^\dagger \mathbf{z}_x(\mathbf{W}) \right), \\
&= \sigma^2 \mathbb{E}_x \mathrm{Tr}(\mathbf{z}_x(\mathbf{W}) \mathbf{z}_x(\mathbf{W})^T (\mathbf{Z}^T\mathbf{Z})^\dagger), \\
&= \sigma^2 \frac{1}{s} \mathrm{Tr} \left(\mathbf{W}^T \mathbb{E}_x (D^{1/2} V_x V_x^T D^{1/2}) \mathbf{W} (\mathbf{Z}^T\mathbf{Z})^\dagger \right), \\
&= \sigma^2 \frac{1}{s} \mathrm{Tr} \left(\mathbf{W}^T \Sigma \mathbf{W} (\mathbf{Z}^T\mathbf{Z})^\dagger \right), \\
&= \sigma^2 \frac{1}{s} \mathrm{Tr} \left(\mathbf{W}^T \Sigma \mathbf{W} (\mathbf{W}^T \frac{1}{s} D^{1/2} V(X) V(X) D^{1/2} \mathbf{W})^\dagger \right), \\
&= \frac{\sigma^2}{n} \mathrm{Tr} \left(\mathbf{W}^T \Sigma \mathbf{W} (\mathbf{W}^T \hat{\Sigma} \mathbf{W})^\dagger \right). \qquad (5.13)
\end{aligned}
$$

As point out before, Σ and $\hat{\Sigma}$ are both positive semidefinite, so they admit eigendecompostion, denoted as $\Sigma = VDV^T$ and $\hat{\Sigma} = \hat{V}\hat{D}\hat{V}^T$. We now denote $\mathbf{w}_i \in \mathbb{R}^s$ to be the i-th column of $\mathbf{W}^T$. Since standard normal vector is invariant under orthonormal transformation,

$$
\begin{aligned}
Eq.\ (5.13) &= \frac{\sigma^2}{n} \mathrm{Tr} \left(\mathbf{W}^T V D V^T \mathbf{W} (\mathbf{W}^T \hat{V} \hat{D} \hat{V}^T \mathbf{W})^\dagger \right), \\
&= \frac{\sigma^2}{n} \mathrm{Tr} \left(\mathbf{W}^T D \mathbf{W} (\mathbf{W}^T \hat{D} \mathbf{W})^\dagger \right) = \frac{\sigma^2}{n} \sum_{i=1}^{p} \lambda_i \mathbf{w}_i^T \left(\mathbf{W}^T \hat{D} \mathbf{W} \right)^\dagger \mathbf{w}_i, \\
&= \frac{\sigma^2}{n} \sum_{i=1}^{p} \lambda_i \mathbf{w}_i^T \left(\sum_{i=1}^{n} \hat{\lambda}_i \mathbf{w}_i \mathbf{w}_i^T \right)^\dagger \mathbf{w}_i, \qquad (5.14)
\end{aligned}
$$

note last equality is because we are in the overparameterized regime where $s \geq n$, hence $\hat{\Sigma}$ has n non-zero eigenvalues. To control the variance term, we need to study Eq. (5.14). To this end, we define the following terms:

$$
\hat{A} = \sum_{i=1}^{n} \hat{\lambda}_i \mathbf{w}_i \mathbf{w}_i^T, \quad \hat{A}_k = \sum_{i>k}^{n} \hat{\lambda}_i \mathbf{w}_i \mathbf{w}_i^T
$$

$\hat{A}$ is a sum of n rank one operator, so it has at most N non-negative eigenvalues. We let $\mu_1(\hat{A}) \geq \cdots \geq \mu_n(\hat{A})$ to be its eigenvalues. We now study the properties of these eigenvalues (the proof follows closely from (Bartlett et al., 2020, Lemma 4)).

Lemma 5.5. *Let $\hat{A} = \sum_{i=1}^{n} \hat{\lambda}_i \mathbf{w}_i \mathbf{w}_i^T$, where $\mathbf{w}_i \in \mathbb{R}^s$ be a random vector with each entry being i.i.d, unit variance and σ_w-subgaussian random variables. There is a universal constant b_1 such that with probability at least $1 - 2e^{-t}$, we have*

$$\sum_{i=1}^{n} \hat{\lambda}_i - \Lambda \leq \mu_n(\hat{A}) \leq \mu_1(\hat{A}) \leq \sum_{i=1}^{n} \hat{\lambda}_i + \Lambda, \tag{5.15}$$

where

$$\Lambda = b_1 \left(\hat{\lambda}_1 (t + n \log 9) + \sqrt{(t + n \log 9) \sum_{i=1}^{n} \hat{\lambda}_i^2} \right).$$

Further, there is a universal constant b_2 such that with probability at least $1 - 2e^{-n/b_2}$

$$\frac{1}{b_2} \sum_{i=1}^{n} \hat{\lambda}_i - b_2 \hat{\lambda}_1 n \leq \mu_n(\hat{A}) \leq \mu_1(\hat{A}) \leq b_2 \sum_{i=1}^{n} \hat{\lambda}_i + b_2 \hat{\lambda}_1 n. \tag{5.16}$$

In addition, with the same probability bound, we have

$$\frac{1}{b_2} \sum_{i>k}^{n} \hat{\lambda}_i - b_2 \hat{\lambda}_{k+1} n \leq \mu_n(\hat{A}_k) \leq \mu_1(\hat{A}_k) \leq b_2 \sum_{i>k}^{n} \hat{\lambda}_i + b_2 \hat{\lambda}_{k+1} n. \tag{5.17}$$

Proof. For any unit vector $\mathbf{v} \in \mathbb{R}^s$, we have $\mathbf{v}^T \mathbf{w}_i$ is still σ_w^2-subgaussian, this implies that $\mathbf{v}^T \mathbf{u}_i \mathbf{w}_i^T \mathbf{v} - 1 = (\mathbf{v}^T \mathbf{w}_i)^2 - 1$ is centered and σ_w^2 subexponential. Applying Lemma 5.11, we have for any unit vector $\mathbf{v}$, there is a universal constant c_1, with probability at least $1 - 2e^{-t}$,

$$\left| \mathbf{v}^T \hat{A} \mathbf{v} - \sum_{i=1}^{n} \hat{\lambda}_i \right| \leq c_1 \left(\hat{\lambda}_1 t + \sqrt{t \sum_{i=1}^{n} \hat{\lambda}_i^2} \right).$$

Now since $\hat{A}$ has at most n non-negative eigenvalues, we let the n dimensional subspace spanned by $\hat{A}$ as $\mathcal{A}^n$, let $\mathcal{N}_\omega$ to be the ω-net of $\mathcal{S}^{n-1}$ with respect to the

Euclidean distance, where $\mathcal{S}^{n-1}$ is the unit sphere in $\mathcal{A}^n$. We let $\omega = \frac{1}{4}$, implying that $|\mathcal{N}_\omega| \leq 9^n$. Apply union bound, for every $\mathbf{v} \in \mathcal{N}_\epsilon$, we have with probability at least $1 - 2e^{-t}$

$$\left| \mathbf{v}^T \hat{A} \mathbf{v} - \sum_{i=1}^n \hat{\lambda}_i \right| \leq c_1 \left(\hat{\lambda}_1 (t + n \log 9) + \sqrt{(t + n \log 9) \sum_{i=1}^n \hat{\lambda}_i^2} \right).$$

By Lemma 5.13, since $\omega = \frac{1}{4}$, for any $\mathbf{v} \in \mathcal{S}^{n-1}$, we have

$$\left| \mathbf{v}^T \hat{A} \mathbf{v} - \sum_{i=1}^n \hat{\lambda}_i \right| \leq c_2 \left(\hat{\lambda}_1 (t + n \log 9) + \sqrt{(t + n \log 9) \sum_{i=1}^n \hat{\lambda}_i^2} \right) := \Lambda.$$

Thus, with probability $1 - 2e^{-t}$, we have

$$\left\| \hat{A} - \sum_{i=1}^n \hat{\lambda}_i I_n \right\| \leq \Lambda.$$

We further simplify Λ now. Notice that when $t \leq \frac{n}{c_3}$, $(t + n \log 9) \leq c_4 n$. Hence,

$$\begin{aligned} \Lambda &\leq c_5 \hat{\lambda}_1 n + \sqrt{c_6 n \hat{\lambda}_1 \sum_{i=1}^n \hat{\lambda}_i} \\ &\leq c_5 \hat{\lambda}_1 n + \frac{1}{2} c_6 c_7 \hat{\lambda}_1 n + \frac{1}{2c_7} \sum_{i=1}^n \hat{\lambda}_i \quad \left(\text{we use } \sqrt{xy} \leq \frac{x+y}{2} \right). \end{aligned}$$

Combining this with Eq. (5.15) yields Eq. (5.16). Using the same proof with $\hat{A}_k$ replacing $\hat{A}$, we obtain Eq. (5.17). □

Lemma 5.6. *For universal constants b, recall the definition of k^* as*

$$k^* = \min \left\{ 0 \leq j \leq n, \frac{\sum_{i>j}^N \hat{\lambda}_i}{\hat{\lambda}_{j+1}} \geq bn \right\},$$

we then have with probability greater than $1-2e^{-\frac{n}{c}}$,

$$\mathbf{V}_R \leq b\sigma^2 \frac{s}{n} \frac{\text{Tr}(\Sigma)}{\sum_{i>k^*}^{n} \hat{\lambda}_i}.$$

Proof. Since $\hat{A} - \hat{A}_k = \sum_{i=1}^{k} \hat{\lambda}_i \mathbf{w}_i \mathbf{w}_i^T$ is positive semidefinite, we have that $\mu_n(\hat{A}) \geq \mu_n(\hat{A}_k)$. By Lemma 5.5, with probability greater than $1-e^{-n/c_1}$, we can lower bound the smallest non-zero eigenvalue of $\hat{A}$ as

$$\mu_n(\hat{A}) \geq \mu_n(\hat{A}_k) \geq \frac{1}{c_1} \sum_{i>k}^{n} \hat{\lambda}_i - c_1 \hat{\lambda}_{k+1} n.$$

Assuming there is $0 \leq j \leq n$ such that $\sum_{i>j}^{n} \hat{\lambda}_i \geq c\hat{\lambda}_{j+1} n$ and $c > c_1^2$, we have

$$\mu_n(\hat{A}) \geq \frac{1}{c_1} \sum_{i>j}^{n} \hat{\lambda}_i - \frac{c_1}{c} \sum_{i>j}^{n} \hat{\lambda}_i = \frac{1}{c_1 c_2} \sum_{i>j}^{n} \hat{\lambda}_i.$$

Thus,

$$\begin{aligned} \mathbf{V}_R &\leq Eq.\ (5.14) \leq \frac{\sigma^2}{n} \sum_{i=1}^{p} \mu_n(\hat{A})^{-1} \lambda_i \mathbf{w}_i^T \mathbf{w}_i, \\ &\leq \frac{\sigma^2}{n} \left(\frac{1}{c_1 c_2} \sum_{i>j}^{n} \hat{\lambda}_i \right)^{-1} \sum_{i=1}^{p} \lambda_i \mathbf{w}_i^T \mathbf{w}_i \end{aligned}$$

since $\mathbf{w}_i$ are standard Gaussian random vector, using Lemma 5.12, we have for some universal constants c_3, c_4, with probability greater than $1-e^{-s/c_3}$, $\mathbf{w}_i^T \mathbf{w}_i \leq c_4 s$. Hence, if we choose $c > c_1^2$ such that $1-e^{-n/c} \leq 1-e^{-s/c_3}$, then with probability greater than $1-2e^{-n/c}$, we have for k^*,

$$\mathbf{V}_R \leq c\sigma^2 \frac{s}{n} \frac{\text{Tr}(\Sigma)}{\sum_{i>k^*}^{N} \hat{\lambda}_i}.$$

□

Lemma 5.7. *There exists a universal constant b such that with probability greater than $1 - 2e^{-\frac{n}{b}}$, we have*

$$\mathbf{V}_R \geq b\sigma^2 \frac{s}{n} \frac{\mathrm{Tr}(\Sigma)}{\sum_{i>k^*}^{n} \hat{\lambda}_i}.$$

Proof. The proof is similar to the upper bound, where the difference is that we use the upper bound of $\mu_1(\hat{A})$ to obtain the lower bound of $\mathbf{V}_R$. □

Now equipped with the above tools, we are ready to prove Proposition 5.1.

Proof. For the upper bound, combining Lemma 5.1, 5.3, 5.4 and 5.6 yields the result. For the lower bound, we simply notice that $R(\tilde{\beta}) \geq \mathbf{V}_R$ and apply Lemma 5.7 to achieve the result. □

5.4.2 Proof of Theorem 5.2

We deal with the bias first in the next section.

5.4.2.1 Upper Bound on Bias

We first notice that

$$\begin{aligned}\mathbf{B}_\xi &= \int_{\mathcal{X}} \left\| \mathbf{z}_x^\xi(\mathbf{W})^T \Pi_\xi \beta_*^\xi \right\|^2 dP(x) = \mathbb{E}_x \left\{ \beta_*^{\xi T} \Pi_\xi \mathbf{z}_x^\xi(\mathbf{W}) \mathbf{z}_x^\xi(\mathbf{W})^T \Pi_\xi \beta_*^\xi \right\}, \\ &= \beta_*^{\xi T} \Pi_\xi \left\{ \mathbb{E}_x(\mathbf{z}_x^\xi(\mathbf{W}) \mathbf{z}_x^\xi(\mathbf{W})^T) - \frac{1}{n} \mathbf{Z}_\xi^T \mathbf{Z}_\xi \right\} \Pi_\xi \beta_*^\xi.\end{aligned}$$

By definition,

$$\frac{1}{n} \mathbf{Z}_\xi^T \mathbf{Z}_\xi = \frac{1}{n} (\mathbf{Z} + \Xi)^T (\mathbf{Z} + \Xi) = \frac{1}{n} (\mathbf{Z}^T \mathbf{Z} + \mathbf{Z}^T \Xi + \Xi^T \mathbf{Z} + \Xi^T \Xi).$$

Thus,

$$\begin{aligned} & \mathbb{E}_x(\mathbf{z}_x^\xi(\mathbf{W})\mathbf{z}_x^\xi(\mathbf{W})^T) - \frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi \\ = \; & \mathbb{E}_x(\mathbf{z}_x^\xi(\mathbf{W})\mathbf{z}_x^\xi(\mathbf{W})^T) - \frac{1}{n}\mathbf{Z}^T\mathbf{Z} && (5.18) \\ & -\frac{1}{n}\mathbf{Z}^T\Xi && (5.19) \\ & -\frac{1}{n}\Xi^T\mathbf{Z} && (5.20) \\ & -\frac{1}{n}\Xi^T\Xi. && (5.21) \end{aligned}$$

Upper Bound Eq. (5.18) By definition, we have

$$\begin{aligned} \mathbf{z}_x^\xi(\mathbf{W})\mathbf{z}_x^\xi(\mathbf{W})^T &= (\mathbf{z}_x(\mathbf{W}) + \boldsymbol{\xi})(\mathbf{z}_x(\mathbf{W}) + \boldsymbol{\xi})^T, \\ &= \mathbf{z}_x(\mathbf{W})\mathbf{z}_x(\mathbf{W})^T + \mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T + \boldsymbol{\xi}^T\mathbf{z}_x(\mathbf{W}) + \boldsymbol{\xi}\boldsymbol{\xi}^T. \end{aligned}$$

As a result, Eq. (5.18) can be written as

$$\begin{aligned} & \mathbb{E}_x\left\{\mathbf{z}_x(\mathbf{W})\mathbf{z}_x(\mathbf{W})^T - \frac{1}{n}\mathbf{Z}^T\mathbf{Z}\right\} && (5.22) \\ & +\mathbb{E}_x(\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T) && (5.23) \\ & +\mathbb{E}_x(\boldsymbol{\xi}\mathbf{z}_x(\mathbf{W})^T) && (5.24) \\ & +\boldsymbol{\xi}\boldsymbol{\xi}^T. && (5.25) \end{aligned}$$

Eq. (5.22) $= \frac{1}{s}(\mathbf{W}^T\Sigma\mathbf{W} - \mathbf{W}^T\hat{\Sigma}\mathbf{W})$, its operator norm can be trivially upper bounded by $\frac{\lambda_W}{s}\|\Sigma - \hat{\Sigma}\|$, where we recall $\lambda_W = \|\mathbf{W}^T\mathbf{W}\|$. We now denote $\boldsymbol{\xi} = \frac{\sigma_0}{\sqrt{s}}\mathbf{u}$, where $\mathbf{u} \in \mathbb{R}^s$ with each entry being i.i.d standard normal by definition of $\boldsymbol{\xi}$. Recall that $\mathbf{z}_x(\mathbf{W}) = \frac{1}{\sqrt{s}}\mathbf{W}^T D^{\frac{1}{2}} V(x)$, we have

$$\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T = \frac{\sigma_0}{s}\mathbf{W}^T D^{\frac{1}{2}} V(x)\mathbf{u}^T,$$

where $\mathbf{w}^{(i)} \in \mathbb{R}^p$ as the i-th column of $\mathbf{W}$. It is easy to see

$$\mathbf{w}^{(i)T} D^{\frac{1}{2}} V(x) = \sum_{j}^{p} \sqrt{\lambda_j} e_j(x) \mathbf{w}_j^{(i)} \in \mathbb{R}.$$

is a normal random variable. Immediately, we can see that it has mean 0 and variance $\text{Var}(\mathbf{w}^{(i)T} D^{\frac{1}{2}} V_x) = \sum_j^p \lambda_j e_j(x) e_j(x) = k(x,x)$. As a result, we have $\mathbf{w}^{(i)T} D^{\frac{1}{2}} V(x) \sim \mathcal{N}(0, k(x,x))$. Hence,

$$\mathbf{W}^T D^{\frac{1}{2}} V(x) = \left[\mathbf{w}^{(1)T} D^{\frac{1}{2}} V(x), \ldots, \mathbf{w}^{(s)T} D^{\frac{1}{2}} V(x) \right]^T = \sqrt{k(x,x)} \mathbf{v},$$

where $\mathbf{v} \in \mathbb{R}^s$ is a Gaussian random vector with each entry being i.i.d $\sim \mathcal{N}(0,1)$. As a result,

$$\mathbf{z}_x(\mathbf{W}) \boldsymbol{\xi}^T = \frac{\sigma_0}{s} \mathbf{W}^T D^{\frac{1}{2}} V(x) \mathbf{u}^T = \frac{\sigma_0}{s} \sqrt{k(x,x)} \mathbf{v} \mathbf{u}^T,$$

Hence, by Lemma 5.14, for some universal constant c_1, c_2, c_3, if we let $t \leq s/c_1$, then with probability greater than $1 - 2e^{-s/c_1}$, we can now upper bound $\mathbb{E}_x(\mathbf{z}_x(\mathbf{W}) \boldsymbol{\xi}^T)$ as

$$\begin{aligned} \|\mathbb{E}_x(\mathbf{z}_x(\mathbf{W}) \boldsymbol{\xi}^T)\| &= \left\| \mathbb{E}_x \left(\frac{\sigma_0}{s} \sqrt{k(x,x)} \mathbf{v} \mathbf{u}^T \right) \right\|, \\ &\leq \frac{\sigma_0}{s} \left\| \sqrt{\mathbb{E}_x(k(x,x))} \right\| \|\mathbf{v}\mathbf{u}^T\|, \\ &\leq \frac{\sigma_0}{s} \sqrt{C_0} \left\| \sum_{i=1}^{s} v_i u_i \right\|, \\ &\leq \frac{\sigma_0}{s} \sqrt{C_0} c_2 s = c_3 \sigma_0. \end{aligned} \tag{5.26}$$

Thus, Eq. (5.23) $\leq c_3 \sigma_0 I$. Eq. (5.24) has the same upper bound as Eq. (5.23) since they have exactly the same eigenvalues.

For Eq. (5.25), we apply Lemma 5.12, $\|\boldsymbol{\xi}\boldsymbol{\xi}^T\| = \frac{\sigma_0^2}{s} \mathbf{u}^T \mathbf{u}$, we have $\|\boldsymbol{\xi}\boldsymbol{\xi}^T\| \leq c_3 \sigma_0^2$ with probability greater than $1 - e^{-s/c_4}$. Combining this all together, we have with

probability greater than $1-3e^{-s/c_5} \geq 1-3e^{-n/c_5}$, $c_5 = \max\{c_1, c_4\}$,

$$\begin{aligned} \|\text{Eq.}(5.18)\| &\leq \|\text{Eq.}(5.22)\| + \|\text{Eq.}(5.24)\| + \|\text{Eq.}(5.23)\| + \|\text{Eq.}(5.25)\| \\ &\leq \frac{\lambda_W}{s}\|\Sigma - \hat{\Sigma}\| + 2c_2\sigma_0 + c_3\sigma_0^2. \end{aligned} \tag{5.27}$$

Upper Bound Eq. (5.19) & Eq. (5.20) Recall that $\mathbf{Z} = \frac{1}{\sqrt{s}}V(X)D^{\frac{1}{2}}\mathbf{W}$, and $\hat{\Sigma} = \frac{1}{n}D^{\frac{1}{2}}V(X)^TV(X)D^{\frac{1}{2}}$ has eigenvalues $\hat{\lambda}_1, \cdots, \hat{\lambda}_n \geq 0$. Hence, $\frac{1}{\sqrt{n}}D^{\frac{1}{2}}V(X)^T$ has singular values as $\{\hat{\sigma}_i\}_{i=1}^n$ with $\hat{\sigma}_i^2 = \hat{\lambda}_i$. And for some orthonormal basis $\hat{\mathcal{U}}_X \in \mathbb{R}^{p\times p}$ and $\hat{\mathcal{V}}_X \in \mathbb{R}^{n\times n}$, $\frac{1}{\sqrt{n}}D^{\frac{1}{2}}V(X)^T$ must admit singular value decomposition as

$$\frac{1}{\sqrt{n}}D^{\frac{1}{2}}V(X)^T = \hat{\mathcal{U}}_X\hat{\mathcal{D}}_X\hat{\mathcal{V}}_X,$$

where $\hat{\mathcal{D}}_X \in \mathbb{R}^{p\times n}$ is the matrix with diagonal elements being the singular values and 0 otherwise. Therefore:

$$\begin{aligned} \frac{1}{n}\mathbf{Z}^T\Xi &= \frac{1}{n\sqrt{s}}\mathbf{W}^TD^{\frac{1}{2}}V(X)^T\Xi = \frac{1}{\sqrt{ns}}\mathbf{W}^T\hat{\mathcal{U}}_X\hat{\mathcal{D}}_X\hat{\mathcal{V}}_X\Xi, \\ &= \frac{1}{\sqrt{ns}}\mathbf{W}^T\hat{\mathcal{D}}_X\mathbf{U}\frac{\sigma_0}{\sqrt{s}} = \frac{\sigma_0}{s\sqrt{n}}\sum_i^n \hat{\sigma}_i\mathbf{w}_i\mathbf{u}_i^T. \end{aligned}$$

where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]^T \in \mathbb{R}^{n\times s}$ with each entry being i.i.d $\sim \mathcal{N}(0,1)$.

We would like to apply Lemma 5.15 to the above equation. But we need to consider the value of $\{\hat{\sigma}_i\}_{i=1}^n$ as $\hat{\sigma}_i$ can either be $\sqrt{\hat{\lambda}_i}$ or $-\sqrt{\hat{\lambda}_i}$. We can see that $\mathbf{w}_i$ is a standard Gaussian random vector, $-\mathbf{w}_i$ and $\mathbf{w}_i$ have the same distribution. As a result, without loss of generality, we may assume that $\hat{\sigma}_i = \sqrt{\hat{\lambda}_i}$, so $\{\hat{\sigma}_i\}_{i=1}^n$ is non-negative and non-increasing. Now apply Lemma 5.15 and notice that $\sum_{i=1}^n \hat{\sigma}_i^2 = \sum_i \hat{\lambda}_i$ is finite by our assumption, we conclude with probability greater than $1-e^{-n/c_6}$,

$$\left\|\frac{1}{n}\Xi^T\mathbf{Z}\right\| = \left\|\frac{1}{n}\mathbf{Z}^T\Xi\right\| \leq \frac{\sigma_0}{s\sqrt{n}}c_7 n\sqrt{\sum_i \hat{\lambda}_i} \leq c_8\frac{\sigma_0}{\sqrt{s}}. \tag{5.28}$$

Upper Bound Eq. (5.21) For this, we simply notice that

$$\frac{1}{n}\Xi^T\Xi = \frac{\sigma_0^2}{sn}\mathbf{U}^T\mathbf{U} = \frac{\sigma_0^2}{sn}\sum_i^n \mathbf{u}_i\mathbf{u}_i^T.$$

Note that since we are in the overparameterized regime ($s \geq n$), $\mathbf{U}^T\mathbf{U}$ has at most n non-zero eigenvalues. Therefore by Lemma 5.5, Eq. (5.16), with probability greater than $1 - e^{-n/c_9}$

$$\begin{aligned}
\left\|\frac{1}{n}\Xi^T\Xi\right\| &= \left\|\frac{\sigma_0^2}{sn}\sum_i^n \mathbf{u}_i\mathbf{u}_i^T\right\|, \\
&\leq \frac{\sigma_0^2}{sn}\left\|\sum_i^n \mathbf{u}_i\mathbf{u}_i^T\right\|, \\
&\leq \frac{\sigma_0^2}{sn}\left(\frac{1}{c_{10}} + c_{10}n\right) \quad \text{by Eq. (5.16)}, \\
&= c_{11}\frac{\sigma_0^2 n}{sn} \leq c_{11}\frac{\sigma_0^2}{s}.
\end{aligned} \tag{5.29}$$

Now combine Eq. (5.27), (5.28) and (5.29) and Lemma 5.4, taking the universal constants $c = \max\{c_1, \ldots, c_{11}\}$, with probability greater than $1 - \delta - 6e^{-n/c}$

$$\mathbf{B}_\xi \leq c\left(\frac{\lambda_W}{s}\|\Sigma\|\sqrt{\log(\frac{14r(\Sigma)}{\delta})/n} + \sigma_0\right)\|\Pi_\xi\|^2\|\beta_*^\xi\|^2,$$

note that we have omitted the σ_0^2, $\frac{\sigma_0}{\sqrt{s}}$ and $\frac{\sigma_0^2}{s}$ terms because they are dominated by σ_0.

5.4.2.2 Upper Bound on Variance

We study the property of the variance $\mathbf{V}_\xi$. As shown in the following equations, to obtain the upper bound on the variance, we need to study the properties of the eigenvalues from $\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi$. The key part is to obtain a high probability bound on

the least positive eigenvalue of the $\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi$ matrix. As explained later, our core technique is to express the $\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi$ matrix as a sum of the outer product of i.i.d Gaussian random vectors. Since the product of the Gaussian random variable is sub-exponential, we then utilize the concentration inequality bounds for sub-exponential random variables to obtain the result. Finally, we apply the standard ϵ-net argument to bound the norm of the $\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi$ matrix, which then provides a lower bound on the least positive eigenvalue of $\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi$.

$$
\begin{aligned}
\mathbf{V}_\xi &= \int_{\mathcal{X}} \mathbb{E}_{\boldsymbol{\epsilon}} \left\| \mathbf{z}_x^\xi(\mathbf{W})^T \left(\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \mathbf{Z}_\xi^T\boldsymbol{\epsilon} \right\|^2 dP(x), \\
&= \int_{\mathcal{X}} \mathbb{E}_{\boldsymbol{\epsilon}} \left\{ \mathbf{z}_x^\xi(\mathbf{W})^T \left(\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \mathbf{Z}_\xi^T\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\mathbf{Z}_\xi \left(\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \mathbf{z}_x^\xi(\mathbf{W}) \right\} dP(x), \\
&\leq \sigma^2 \int_{\mathcal{X}} \mathbf{z}_x^\xi\left(\mathbf{W}\right)^T (\mathbf{Z}_\xi^T\mathbf{Z}_\xi)^\dagger \mathbf{Z}_\xi^T\mathbf{Z}_\xi \left(\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \mathbf{z}_x^\xi(\mathbf{W}) dP(x), \\
&= \sigma^2\mathbb{E}_x \left\{ \mathbf{z}_x^\xi(\mathbf{W})^T \left(\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \mathbf{z}_x^\xi(\mathbf{W}) \right\}, \\
&= \sigma^2\mathbb{E}_x \left\{ \mathrm{Tr}\left[\mathbf{z}_x^\xi(\mathbf{W})\mathbf{z}_x^\xi(\mathbf{W})^T \left(\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \right] \right\}, \\
&= \frac{\sigma^2}{n}\mathrm{Tr}\left\{ \mathbb{E}_x\left[\mathbf{z}_x^\xi(\mathbf{W})\mathbf{z}_x^\xi(\mathbf{W})^T \right] \left(\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \right\}, \\
&= \frac{s\sigma^2}{n}\mathrm{Tr}\left\{ \mathbb{E}_x\left[\mathbf{z}_x(\mathbf{W})\mathbf{z}_x(\mathbf{W})^T \right] \left(\frac{s}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \right\}, && (5.30) \\
&\quad + \frac{s\sigma^2}{n}\mathrm{Tr}\left\{ \mathbb{E}_x\left[\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T \right] \left(\frac{s}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \right\}, && (5.31) \\
&\quad + \frac{s\sigma^2}{n}\mathrm{Tr}\left\{ \mathbb{E}_x\left[\boldsymbol{\xi}\mathbf{z}_x(\mathbf{W})^T \right] \left(\frac{s}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \right\}, && (5.32) \\
&\quad + \frac{s\sigma^2}{n}\mathrm{Tr}\left\{ \boldsymbol{\xi}\boldsymbol{\xi}^T \left(\frac{s}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \right\}. && (5.33)
\end{aligned}
$$

In order to do that, we first provide a more refined way of studying the eigenvalues of $\hat{A} = \sum_{i=1}^N \hat{\lambda}_i \mathbf{w}_i\mathbf{w}_i^T$ than Lemma 5.5.

Lemma 5.8. *Let $A = \sum_{i=1}^n \lambda_i \mathbf{w}_i\mathbf{w}_i^T$, where $\mathbf{w}_i \in \mathbb{R}^s$ is a random vector with each entry being i.i.d σ_w^2-subgaussian random variable and $\{\lambda_i\}_{i=1}$ is a sequence*

of non-negative, non-increasing numbers with finite sum $\sum_{i=1}^{n}\lambda_i < \infty$. *Assume* $s > n$, *with probability at least* $1-2e^{-n/b}$,

$$(1-\frac{1}{bb_1}\sigma_w^2)\sum_{i=1}^{n}\lambda_i - \frac{b_2}{\sqrt{b}}\sigma_w^2\lambda_1 n \le \mu_n(A) \le \mu_1(A) \le (1+\frac{1}{bb_1}\sigma_w^2)\sum_{i=1}^{n}\lambda_i + \frac{b_2}{\sqrt{b}}\sigma_w^2\lambda_1 n,$$

where $b, b_1, b_2 > 1$ *are some universal constants. In addition, with the same probability bound, we have*

$$(1-\frac{1}{bb_1}\sigma_w^2)\sum_{i>k}^{n}\lambda_i - \frac{b_2}{\sqrt{b}}\sigma_w^2\lambda_{k+1} n \le \mu_n(A_k) \le \mu_1(A_k) \le (1+\frac{1}{bb_1}\sigma_w^2)\sum_{i>k}^{n}\lambda_i + \frac{b_2}{\sqrt{b}}\sigma_w^2\lambda_{k+1} n,$$

Proof. For any unit vector $\mathbf{v}\in\mathbb{R}^s$, we have $\mathbf{v}^T\mathbf{w}_i$ is $c_1\sigma_w^2$-subgaussian random variable. Hence for any $\mathbf{v}$, $\mathbf{v}^TA\mathbf{v} = \sum_{i=1}^{n}\lambda_i(\mathbf{v}^T\mathbf{w}_i)^2$. Applying Lemma 5.11, for any unit vector $\mathbf{v}$, there is a universal constant c_1, a constant c_2 and $t>0$, with probability at least $1-2e^{-t/c_2}$,

$$|\mathbf{v}^TA\mathbf{v} - \sum_{i=1}\lambda_i| \le c_1\sigma_w^2\max\left(\lambda_1\frac{t}{c_2}, \sqrt{\frac{t}{c_2}\sum_{i=1}\lambda_i^2}\right) \le \frac{c_1}{\sqrt{c_2}}\sigma_w^2\left(\lambda_1 t + \sqrt{t\sum_{i=1}\lambda_i^2}\right).$$

Now since A has at most n non-negative eigenvalues, we let the n dimensional subspace spanned by A as $\mathcal{A}^n$, let $\mathcal{N}_\omega$ to be the ω-net of $\mathcal{S}^{n-1}$ with respect to the Euclidean distance, where $\mathcal{S}^{n-1}$ is the unit sphere in $\mathcal{A}^n$. We let $\omega=\frac{1}{4}$, implying that $|\mathcal{N}_\omega|\le 9^n$. Apply union bound, for every $\mathbf{v}\in\mathcal{N}_\epsilon$, we have with probability at least $1-2e^{-t/c_2}$

$$\left|\mathbf{v}^TA\mathbf{v} - \sum_{i=1}^{n}\lambda_i\right| \le \frac{c_1}{\sqrt{c_2}}\sigma_w^2\left(\lambda_1(t+n\log 9) + \sqrt{(t+n\log 9)\sum_{i=1}\lambda_i^2}\right).$$

By Lemma 5.13, since $\omega=\frac{1}{4}$, for any $\mathbf{v}\in\mathcal{S}^{n-1}$, we have

$$\left|\mathbf{v}^T A\mathbf{v} - \sum_{i=1}^{n}\lambda_i\right| \leq \frac{32}{9}\frac{c_1}{\sqrt{c_2}}\sigma_w^2\left(\lambda_1(t+n\log 9) + \sqrt{(t+n\log 9)\sum_{i=1}\lambda_i^2}\right),$$
$$= \frac{c_3}{\sqrt{c_2}}\sigma_w^2\left(\lambda_1(t+n\log 9) + \sqrt{(t+n\log 9)\sum_{i=1}\lambda_i^2}\right) := \Lambda.$$

Thus, with probability $1-2e^{-t/c_2}$, we have

$$\left\|A - \sum_{i=1}\lambda_i I_n\right\| \leq \Lambda.$$

We further simplify Λ now. Notice that when $t \leq n$, $(t+n\log 9) \leq c_4 n$. Hence,

$$\begin{aligned}
\Lambda &\leq \frac{c_3}{\sqrt{c_2}}\sigma_w^2\left(c_4\lambda_1 n + \sqrt{c_4 n\sum_{i=1}^{n}\lambda_i^2}\right), \\
&\leq \frac{c_3}{\sqrt{c_2}}\sigma_w^2\left(c_4\lambda_1 n + \sqrt{c_4 n\lambda_1\sum_{i=1}^{n}\lambda_i}\right), \\
&\leq \frac{c_3}{\sqrt{c_2}}\sigma_w^2\left(c_4\lambda_1 n + \frac{c_4 c_6 n\lambda_1}{2} + \frac{1}{2c_6}\sum_{i=1}^{n}\lambda_i\right) \quad (\text{we use } \sqrt{xy} \leq \frac{x+y}{2}), \\
&= \frac{1}{\sqrt{c_2}}\sigma_w^2(c_7\lambda_1 n + \frac{1}{c_8}\sum_{i=1}^{n}\lambda_i).
\end{aligned}$$

Therefore, with probability $1-2e^{-n/c_2}$, we have:

$$(1 - \frac{1}{c_8\sqrt{c_2}}\sigma_w^2)\sum_{i=1}^{n}\lambda_i - \frac{c_7}{\sqrt{c_2}}\sigma_w^2\lambda_1 n \leq \mu_n(A)$$
$$\leq \mu_1(A) \leq (1 + \frac{1}{c_8\sqrt{c_2}}\sigma_w^2)\sum_{i=1}^{n}\lambda_i + \frac{c_7}{\sqrt{c_2}}\sigma_w^2\lambda_1 n.$$

Using the same proof with A_k replacing A, we obtain the bound for $\mu_n(A_k)$ and $\mu_1(A_k)$. □

We first investigate the property of $\frac{1}{\sqrt{n}}(\mathbf{Z}+\mathbf{\Xi})$. As discussed before, for some orthonormal basis $\hat{\mathcal{U}}_X \in \mathbb{R}^{p\times p}$ and $\hat{\mathcal{V}}_X \in \mathbb{R}^{n\times n}$, $\frac{1}{\sqrt{n}}D^{\frac{1}{2}}V(X)^T$ must admit singular

value decomposition as

$$\frac{1}{\sqrt{n}} D^{\frac{1}{2}} V(X)^T = \hat{\mathcal{U}}_X \hat{\mathcal{D}}_X \hat{\mathcal{V}}_X,$$

where $\hat{\mathcal{D}}_X \in \mathbb{R}^{p\times n}$ is the matrix with diagonal elements being the singular values $\{\hat{\sigma}_i\}_{i=1}^n$ and 0 otherwise. As a result, we can write

$$\frac{1}{\sqrt{n}}\mathbf{Z} = \frac{1}{\sqrt{n}} V(X) D^{\frac{1}{2}} \mathbf{W} = \hat{\mathcal{V}}_X \hat{\mathcal{D}}_X^T \hat{\mathcal{U}}_X \mathbf{W}.$$

Since Gaussian random variable is invariant under orthogonal transformation, we can further simplify the above equation as

$$\frac{1}{\sqrt{n}}\mathbf{Z} = \hat{\mathcal{V}}_X \hat{\mathcal{D}}_X^T \mathbf{W}.$$

In addition, the singular value matrix $\hat{\mathcal{D}}_X \in \mathbb{R}^{p\times n}$ has only n diagonal entry being non-zero, if we let $\mathcal{D}_X = \text{diag}[\hat{\sigma}_1, \ldots, \hat{\sigma}_n] \in \mathbb{R}^{n\times n}$, then equivalently, we can write

$$\frac{1}{\sqrt{n}}\mathbf{Z} = \hat{\mathcal{V}}_X \mathcal{D}_X \mathbf{W}.$$

Note that the $\mathbf{W}$ in the last equation represents a $n \times s$ Gaussian random matrix.

Therefore,

$$\frac{1}{\sqrt{n}}(\mathbf{Z} + \mathbf{\Xi}) = \frac{1}{\sqrt{s}} \left(\hat{\mathcal{V}}_X \mathcal{D}_X \mathbf{W} + \frac{\sigma_0}{\sqrt{n}} \mathbf{U} \right) = \frac{1}{\sqrt{s}} \hat{\mathcal{V}}_X \left(\mathcal{D}_X \mathbf{W} + \frac{\sigma_0}{\sqrt{n}} \mathbf{U} \right).$$

It is easy to see that $\mathcal{D}_X \mathbf{W} + \frac{\sigma_0}{\sqrt{n}}\mathbf{U}$ is a $n \times s$ Gaussian random matrix. Its i, j-th entry can be written as $\hat{\sigma}_i \mathbf{W}_{ij} + \frac{\sigma_0}{\sqrt{n}} \mathbf{U}_{ij}$. It is a Gaussian random variable with mean 0 and variance $\lambda_i + \frac{\sigma_0^2}{n}$. Hence, we can write

$$\frac{1}{\sqrt{n}}(\mathbf{Z} + \mathbf{\Xi}) = \frac{1}{\sqrt{s}} \hat{\mathcal{V}}_X \left(\hat{D} + \frac{\sigma_0^2}{n} I \right)^{\frac{1}{2}} \mathbf{W}.$$

Finally, we have

$$\frac{s}{n}(\mathbf{Z}+\mathbf{\Xi})^T(\mathbf{Z}+\mathbf{\Xi}) = \mathbf{W}^T\left(\hat{D}+\frac{\sigma_0^2}{n}I\right)\mathbf{W} = \sum_{i=1}^{n}\left(\hat{\lambda}_i+\frac{\sigma_0^2}{n}\right)\mathbf{w}_i\mathbf{w}_i^T$$
$$= \sum_{i=1}^{n}\hat{\lambda}_i^{\xi}\mathbf{w}_i\mathbf{w}_i^T := A^{\xi}.$$

Note that A^{ξ} has at most n positive eigenvalues since we are in the overparameterized regime. Now we apply Lemma 5.8 to A^{ξ} with $\sigma_w^2 = 1$, we have with probability greater than $1-2e^{-n/c_1}$,

$$\begin{aligned}\mu_n(A^{\xi}) \geq \mu_n(A_k^{\xi}) &\geq (1-\frac{1}{c_1c_2})\sum_{i>k}^{n}\hat{\lambda}_i^{\xi} - \frac{c_3}{\sqrt{c_1}}\hat{\lambda}_{k+1}^{\xi}n,\\ &= \hat{\lambda}_{k+1}^{\xi}\left((1-\frac{1}{c_1c_2})\sum_{i>k}^{n}\frac{\hat{\lambda}_i^{\xi}}{\hat{\lambda}_{k+1}^{\xi}} - \frac{c_3}{\sqrt{c_1}}n\right),\\ &\geq \hat{\lambda}_{k+1}\left((1-\frac{1}{c_1c_2})\sum_{i>k}^{n}\frac{\hat{\lambda}_i^{\xi}}{\hat{\lambda}_{k+1}^{\xi}} - \frac{c_3}{\sqrt{c_1}}n\right). \qquad (5.34)\end{aligned}$$

Since we assume that $\sum_{i>k}^{n}\hat{\lambda}_i^{\xi}/\hat{\lambda}_{k+1}^{\xi} = \sum_{i>k}^{n}(\hat{\lambda}_i+\sigma_0^2/n)/(\hat{\lambda}_{k+1}+\sigma_0^2/n) \geq \frac{1}{a}n$. If we adjust $c_1 > 1$ such that $\left((1-\frac{1}{c_1c_2}))an - \frac{c_3}{\sqrt{c_1}}n\right) \geq \frac{1}{c_4}n$, we then have that $\mu_n(A^{\xi}) \geq \frac{1}{c_5}n$.

Now for Eq. (5.30), with probability greater than $1-e^{-n/c_1}$,

$$\begin{aligned}\text{Eq.}(5.30) &= \frac{s\sigma^2}{n}\text{Tr}\left\{\mathbb{E}_x\left[\mathbf{z}_x(\mathbf{W})\mathbf{z}_x(\mathbf{W})^T\right]\left(\frac{s}{n}\mathbf{Z}_{\xi}^T\mathbf{Z}_{\xi}\right)^{\dagger}\right\}\\ &= \frac{\sigma^2}{n}\text{Tr}\left\{\mathbf{W}^T\Sigma\mathbf{W}\left(\frac{s}{n}\mathbf{Z}_{\xi}^T\mathbf{Z}_{\xi}\right)^{\dagger}\right\},\\ &= \frac{\sigma^2}{n}\text{Tr}\left\{\mathbf{U}^TD\mathbf{U}\left(\frac{s}{n}\mathbf{Z}_{\xi}^T\mathbf{Z}_{\xi}\right)^{\dagger}\right\} = \frac{\sigma^2}{n}\sum_{i}^{p}\lambda_i\mathbf{u}_i^T\left(\frac{s}{n}\mathbf{Z}_{\xi}^T\mathbf{Z}_{\xi}\right)^{\dagger}\mathbf{u}_i,\\ &\leq \frac{\sigma^2}{n}\mu_n\left(\frac{s}{n}\mathbf{Z}_{\xi}^T\mathbf{Z}_{\xi}\right)^{-1}\sum_{i}^{p}\lambda_i\mathbf{u}_i^T\mathbf{u}_i, \qquad (5.35)\\ &\leq \frac{\sigma^2}{n}\left\{\frac{1}{c_5}n\right\}^{-1}c_6s\sum_{i=1}^{p}\lambda_i = c_7\sigma^2\text{Tr}(\Sigma)\frac{s}{n^2}.\end{aligned}$$

Note for Eq. (5.35), $\sum_{i=1}^{p}\lambda_i \mathbf{u}_i^T\mathbf{u}_i$ is a weighted sum of χ_1^2 random variables, with weights given by the λ_i in block size of s. Hence, if we let $t < s/c_8$, Lemma 5.11 gives that with probability $1-2e^{-t}$,

$$\begin{aligned}\sum_{i=1}^{p}\lambda_i\mathbf{u}_i^T\mathbf{u}_i &\le s\sum_{i=1}^{p}\lambda_i + b\max\left\{\lambda_1 t, \sqrt{st\sum_{i=1}^{p}\lambda_i^2}\right\},\\ &\le s\sum_{i=1}^{p}\lambda_i + b\max\left\{t\sum_{i=1}^{p}\lambda_i, \sqrt{st}\sum_{i=1}^{p}\lambda_i\right\} \le c_6 s\sum_{i=1}^{p}\lambda_i.\end{aligned}$$

Also, with probability greater than $1-e^{-n/c_9}$

$$\begin{aligned}\text{Eq.}(5.31) &= \frac{s\sigma^2}{n}\mathbb{E}_x\left\{\boldsymbol{\xi}^T\left(\frac{s}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \mathbf{z}_x(\mathbf{W})\right\},\\ &\le \frac{s\sigma^2}{n}\left\|\mathbb{E}_x\left\{\boldsymbol{\xi}^T\left(\frac{s}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger \mathbf{z}_x(\mathbf{W})\right\}\right\|\\ &\le \frac{s\sigma^2}{n}\mu_n\left(\frac{s}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^{-1}\left\|\mathbb{E}_x(\boldsymbol{\xi}^T\mathbf{z}_x(\mathbf{W}))\right\|,\\ &\le \frac{s\sigma^2}{n}\left\{\frac{1}{c_5}n\right\}^{-1} c_{10}\sigma_0 \quad \text{(by Eq. (5.26))},\\ &= c_{11}\sigma^2\frac{s}{n^2}\sigma_0.\end{aligned}$$

Finally, with probability greater than $1-e^{-N/c_{12}}$

$$\begin{aligned}\text{Eq.}(5.33) &= \frac{s\sigma^2}{n}\text{Tr}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\left(\frac{s}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger\right\} = \frac{s\sigma^2}{n}\left\{\boldsymbol{\xi}^T\left(\frac{s}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger\boldsymbol{\xi}\right\}\\ &\le \frac{s\sigma^2}{n}\boldsymbol{\xi}^T\boldsymbol{\xi}\left\{\frac{1}{c_5}n\right\}^{-1} \le \frac{s\sigma^2}{n}\frac{\sigma_0^2}{s}\mathbf{u}^T\mathbf{u}\left\{\frac{1}{c_6}n\right\}^{-1} \quad \text{(by Lemma 5.12)}\\ &= c_{13}\sigma^2\frac{s\sigma_0^2}{n^2}.\end{aligned}$$

Combining above all together, if we choose $c=\max\{c_1,\dots\}$, then with probability greater than $1-5e^{-n/c}$

$$\mathbf{V}_\xi \le c\sigma^2\text{Tr}(\Sigma)\frac{s}{n^2}.$$

Note that we have omitted the $\frac{s\sigma_0}{n^2}$, $\frac{s\sigma_0^2}{n^2}$ terms because they are strictly dominated

by the $\frac{s}{n^2}$ term.

5.5 Discussion

The benign overfitting phenomenon has attracted great research interest since it was first observed by Zhang et al. (2016) and Belkin et al. (2019a, 2018). Our work continues the lines of work of Belkin et al. (2019b), Bartlett et al. (2020), and Hastie et al. (2019), and focuses on developing a theoretical understanding of this phenomenon. Through analyzing the learning risk of the MNLS estimator, we first provide a nearly matching upper and lower bound for the excess learning risk and point out one possible cause of benign overfitting: the noise ξ in the covariates or in the feature vectors. Moreover, we discover that ξ plays an important implicit regularization role during learning. By incorporating ξ into our analysis, we explicitly derive how the learning risk is affected by ξ. We also provide a detailed trade-off between the number of parameters and the learning rate, which further demonstrates the optimality in the minimax sense for the MNLS estimator. Our results may shed light on the theoretical understandings of modern deep learning, which open doors for future studies on the design of deep learning architectures. Furthermore, our results can apply to any finite sample data size or asymptotic case with arbitrary data dimension. They rely on very weak assumptions of the kernel and require almost no assumption about the data generating distribution.

We believe that several extensions are worth exploring. First, although our results shed light on the two-layer neural network with fixed first layer weights, we would like to understand what would happen if we could optimize the first layer weights, i.e., optimizing $\mathbf{W}$ in our model. Second, in the existing literature, there are many tools in analyzing neural network optimization with connection to its generalization error. Examples include the transport map formulation (Suzuki, 2020), the mean-

field analysis (Mei et al., 2018; Sirignano and Spiliopoulos, 2020), and the neural tangent kernel (Du et al., 2019; Jacot et al., 2018). How to utilize these tools in investigating the effect of the noise ξ during neural network training would be an interesting direction. Another direction is to analyze the noise ξ in models with different loss functions.

5.6 Auxilliary Results

5.6.1 Bias-Variance trade-off: Unrealizable Case

Proof. We decompose the excess risk as follows:

$$\begin{aligned}
R(\tilde{\beta}) &= \mathbb{E}_{x,\epsilon}\left(\tilde{f}(x) - f_*(x)\right)^2 = \mathbb{E}_{x,\epsilon}\left(\mathbf{z}_x(\mathbf{w})^T\tilde{\beta} - f_*(x)\right)^2, \\
&= \mathbb{E}_{x,\epsilon}\left\{\mathbf{z}_x(\mathbf{w})^T(\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T Y - f_{\tilde{\mathcal{H}}}(x) + [f_{\tilde{\mathcal{H}}}(x) - f_*(x)]\right\}^2, \\
&\leq 3\mathbb{E}_{x,\epsilon}\left\{\mathbf{z}_x(\mathbf{w})^T(\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T\left(f_{\tilde{\mathcal{H}}}(X) + \boldsymbol{\epsilon}\right) - f_{\tilde{\mathcal{H}}}(x)\right\}^2 := A \\
&+ 3\left\{\mathbb{E}_x\left\{\mathbf{z}_x(\mathbf{w})^T(\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T\left(f_*(X) - f_{\tilde{\mathcal{H}}}(X)\right)\right\}^2 + \mathbb{E}_x\left(f_*(x) - f_{\tilde{\mathcal{H}}}(x)\right)^2\right\} := \mathbf{M}_R.
\end{aligned}$$

Now we can see that the risk has been decomposed into the A term and the misspecification error term $\mathbf{M}_R$. For the A term, we can further decompose it into:

$$\begin{aligned}
A &= \mathbb{E}_{x,\epsilon}\left\{\mathbf{z}_x\left(\mathbf{w}\right)^T(\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T(f_{\tilde{\mathcal{H}}}(X) + \boldsymbol{\epsilon}) - \mathbf{z}_x(\mathbf{w})^T\beta_{\tilde{\mathcal{H}}}\right\}^2, \\
&= \mathbb{E}_{x,\epsilon}\left\{\mathbf{z}_x(\mathbf{w})^T\left(\mathbf{Z}^T\mathbf{Z}\right)^\dagger\mathbf{Z}^T\boldsymbol{\epsilon} + \mathbf{z}_x(\mathbf{w})^T\left((\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T\mathbf{Z} - I\right)\beta_{\tilde{\mathcal{H}}}\right\}^2, \\
&= \int_{\mathcal{X}}\|\mathbf{z}_x(\mathbf{W})^T\left[(\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T\mathbf{Z} - I\right]\beta_{\tilde{\mathcal{H}}}\|^2 dP(x) \\
&\quad + \int_{\mathcal{X}}\mathbb{E}_\epsilon\|\mathbf{z}_x(\mathbf{W})^T(\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T\boldsymbol{\epsilon}\|^2 dP(x).
\end{aligned}$$

□

5.6.2 Bias-related Inequaity

Lemma 5.9. *Let $\mathbf{Z} \in \mathbb{R}^{n\times s}$ be a feature matrix. Recall $\Pi = (\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T\mathbf{Z} - I$, we have $\|\Pi\| = 0$, if $s \leq n$; Otherwise, $\|\Pi\| \leq 1$.*

Proof. In case $s < n$, we have $(\mathbf{Z}^T\mathbf{Z})^\dagger = (\mathbf{Z}^T\mathbf{Z})^{-1}$. As a result $\|\Pi\| = 0$. If $s > n$, by (Hastie et al., 2019, Lemma 1), we know that $I - (\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T\mathbf{Z}$ is a projection onto the null space of $\mathbf{Z}$. Hence, if we pick any vector from that space, we have $(I - (\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T\mathbf{Z})v = v$. Thus,

$$\|I - (\mathbf{Z}^T\mathbf{Z})^\dagger\mathbf{Z}^T\mathbf{Z}\| \leq 1.$$

□

5.6.3 Probability Bound on Sequences and Matrix Norms

Lemma 5.10. *Let x, y be centered σ_x^2-subgaussian and σ_y^2-subgaussian random variables respectively, i.e.,*

$$\mathbb{E}(\exp(tx)) \leq \exp(\frac{\sigma_x^2}{2}t^2), \quad \mathbb{E}(\exp(ty)) \leq \exp(\frac{\sigma_y^2}{2}t^2).$$

We then have that, for a universal constant b, the product of x and y is a centered $b\sigma_x\sigma_y$-subexponential random variable, i.e.,

$$\mathbb{E}(\exp(txy)) \leq \exp(b^2\sigma_x^2\sigma_y^2t^2).$$

Proof. We compute the moment generating function directly,

$$\begin{aligned}
\mathbb{E}(\exp(txy)) &= \mathbb{E}\left\{\mathbb{E}\left(\exp(txy)\middle|y\right)\right\}, \\
&\leq \mathbb{E}\left\{\exp\left(\frac{\sigma_x^2}{2}t^2y^2\right)\right\}, \\
&\leq \exp\left(c_1\frac{\sigma_x^2}{2}\frac{\sigma_y^2}{2}t^2\right), \\
&= \exp(c\sigma_x^2\sigma_y^2t^2),
\end{aligned}$$

provided $|t| \leq (c\sigma_x\sigma_y)^{-1}$. Note that for the last inequality, we have used the results from (Vershynin, 2018, Proposition 2.5.2). □

Below are two concentration results for subexponential and subgaussian random variables from (Bartlett et al., 2020, Corollary S.6 & S.7).

Lemma 5.11. *Suppose we have a sequence of non-increasing and non-negative numbers* $\{\lambda_i\}_{i=1}^{\infty}$ *such that* $\sum_{i=1}^{\infty}\lambda_i < \infty$. *In addition, we have a sequence of i.i.d centered,* σ*-subexponential random variables* $\{\xi_i\}_{i=1}^{\infty}$, *then there is a universal constant* b *such that for probability greater than* $1-2e^{-t}$, $t>0$, *we have*

$$\left|\sum_i \lambda_i\xi_i\right| \leq b\sigma \max\left(\lambda_1 t, \sqrt{t\sum_i \lambda_i^2}\right).$$

Lemma 5.12. *let* $\mathbf{w} \in \mathbb{R}^n$ *be a random vector with each coordinate being a mean* 0, *unit variance,* σ_w^2*-subgaussian random variable. Then there is a universal constant* b *such that with probability greater than* $1-2e^{-t}$, *we have*

$$\|\mathbf{w}\|^2 \leq n + b\sigma_w^2\left(t+\sqrt{nt}\right).$$

In particular, if $t < \frac{n}{b_1}$,

$$\|\mathbf{w}\|^2 \leq b_2 n,$$

for some universal constants b_1, b_2.

Lemma 5.13 are from (Bartlett et al., 2020, Lemma S.8).

Lemma 5.13. (ϵ*-net argument*) *Let* $A \in \mathbb{R}^{n\times n}$ *be a symmetric matrix, and* $\mathcal{N}_\epsilon$ *is an* ϵ*-net on the unit sphere* $\mathcal{S}^{n-1}$ *with* $\epsilon < \frac{1}{2}$. *Then we have*

$$\|A\| \leq (1-\epsilon)^{-2}\max_{\mathbf{a}\in\mathcal{N}_\epsilon}\left|\mathbf{a}^T A\mathbf{a}\right|.$$

Lemma 5.14. *Let $\{w_i\}_{i=1}^{\infty}, \{u_i\}_{i=1}^{\infty}$ be two sequences of i.i.d random variables distributed as standard normal. Moreover, let $\{\lambda_i\}_{i=1}^{\infty}$ be a sequence of non-negative, non-increasing numbers such that $\sum_{i=1}^{\infty}\lambda_i^2 < \infty$. Then with probability greater than $1-2e^{-t}$, we have*

$$-\left(b_1\lambda_1 t + \sqrt{b_2 t\sum_i \lambda_i^2}\right) \le \sum_{i=1}^{\infty}\lambda_i w_i u_i \le b_1\lambda_1 t + \sqrt{b_2 t\sum_i \lambda_i^2}.$$

Proof. Using Markov Inequality and for any $\tau > 0$, we have that

$$\begin{aligned} P(\sum_{i=1}^{\infty}\lambda_i w_i u_i \ge t) &= P(\exp(\tau\sum_{i=1}^{\infty}\lambda_i w_i u_i) \ge e^{\tau t}) \\ &\le e^{-\tau t}\prod_{i=1}\mathbb{E}(e^{\tau\lambda_i w_i u_i}) \\ &= e^{-\tau t}\prod_{i=1}(\frac{1}{1-(\tau\lambda_i)^2})^{\frac{1}{2}} \quad \text{(by Lemma 5.17)} \\ &= \exp(-\tau t - \frac{1}{2}\sum_{i=1}\log(1-(\tau\lambda_i)^2)), \end{aligned}$$

provided $\tau\lambda_i < 1$. Now for any $x \in (0,1)$,

$$\begin{aligned} \log(1-x) &= -\int_0^x \frac{1}{1-t}dt \\ &\ge -\int_0^x \frac{1}{(1-t)^2}dt \\ &= -\frac{x}{1-x}. \end{aligned}$$

As a result

$$\begin{aligned} &\exp(-\tau t - \frac{1}{2}\sum_{i=1}\log(1-(\tau\lambda_i)^2)) \\ \le\ &\exp(-\tau t - \sum_{i=1} -\frac{(\tau\lambda_i)^2}{1-(\tau\lambda_i)^2}) \\ \le\ &\exp(-\tau t + \frac{\tau^2}{1-(\tau\lambda_1)^2}\sum_{i=1}\lambda_i^2) \\ \le\ &\exp\left\{-\tau t + \frac{\tau^2}{1-(\tau\lambda_1)^2}\sum_{i=1}\lambda_i^2 + \frac{\tau}{2\lambda_1}[\log(\frac{1}{\tau}+\lambda_1) - \log(\frac{1}{\tau}-\lambda_1)]\right\} \end{aligned}$$

Note that for the last inequality, we have $\frac{1}{\tau} - \lambda_1 > 0$, since $\tau\lambda_1 < 1$. Now we let

$$x = \tau t - \frac{\tau^2}{1-(\tau\lambda_1)^2}\sum_{i=1}\lambda_i^2 - \frac{\tau}{2\lambda_1}[\log(\frac{1}{\tau}+\lambda_1) - \log(\frac{1}{\tau}-\lambda_1)],$$

we have

$$t = \frac{x}{\tau} - \frac{1}{\tau}\frac{1}{(\frac{1}{\tau})^2-\lambda_1}\sum_{i=1}\lambda_i^2 - \frac{1}{2\lambda_1}[\log(\frac{1}{\tau}+\lambda_1) - \log(\frac{1}{\tau}-\lambda_1)].$$

Optimize over τ^{-1}, this gives

$$\tau^{-1} = \sqrt{\lambda_1^2 + \sqrt{\lambda_1^2\frac{\sum_i\lambda_i^2}{x}}}.$$

Checking the conditions, we see that $\tau\lambda_i = \lambda_i\left(\lambda_1^2 + \sqrt{\lambda_1^2\frac{\sum_i\lambda_i^2}{x}}\right)^{-\frac{1}{2}} < 1$, implying τ^{-1} is a valid choice. Now with this choice, some simple algebra shows that there are constants c_1, c_2 such that

$$t \le c_1\lambda_1 x + \sqrt{c_2 x\sum_i\lambda_i^2}.$$

This gives

$$P\left(\sum_{i=1}^{\infty}\lambda_i w_i u_i \ge c_1\lambda_1 x + \sqrt{c_2 x\sum_i\lambda_i^2}\right) \le \exp(-x).$$

For the lower bound we use the symmetry of Gaussian distribution as

$$\begin{aligned} P\left(\sum_{i=1}^{\infty}\lambda_i w_i u_i \le -t\right) &= P\left(-\sum_{i=1}^{\infty}\lambda_i w_i u_i \ge t\right) \\ &= P\left(\sum_{i=1}^{\infty}\lambda_i(-w_i)u_i \ge t\right) \\ &= P\left(\sum_{i=1}^{\infty}\lambda_i w_i u_i \ge t\right). \end{aligned}$$

We complete the proof by using the union bound for the probability for both the upper and lower bound. □

Lemma 5.15. *Let* $\{\mathbf{w}_i\}_{i=1}^{\infty}$ *and* $\{\mathbf{u}_i\}_{i=1}^{\infty}$ *be two sequence of random vectors where the entry of each* $\mathbf{w}_i \in \mathbb{R}^n$ *and* $\mathbf{u}_i \in \mathbb{R}^n$ *is i.i.d* $\mathcal{N}(0,1)$*. Furthermore, let* $\{\lambda_i\}_{i=1}^{\infty}$ *be a non-negative sequence such that* $\{\lambda_i^2\}_{i=1}^{\infty}$ *is non-increasing and* $\sum_{i=1}\lambda_i^2 < \infty$*. Denote* $A = \sum_{i=1}\lambda_i \mathbf{w}_i \mathbf{u}_i^T$*. Let* $\mu_1(A) \geq, \cdots, \geq \mu_n(A)$ *be its eigenvalues. Then with probability at least* $1 - 2e^{-n/b_1}$*, we have*

$$-b_2 n \leq \mu_n(A) \leq \mu_1(A) \leq b_2 n.$$

Proof. For any unit vector $\mathbf{v} \in \mathbb{R}^n$, we have that both $\mathbf{v}^T\mathbf{w}_i$ and $\mathbf{v}^T\mathbf{u}_i$ are independent and distributed as $\mathcal{N}(0,1)$. Using Lemma 5.14, we have with $1 - 2e^{-t}$,

$$\begin{aligned} |\mathbf{v}^T A \mathbf{v}| &= |\sum_{i=1}\lambda_i \mathbf{v}^T\mathbf{w}_i\mathbf{u}_i^T\mathbf{v}| \\ &\leq c_1\lambda_1 t + \sqrt{c_2 t \sum_i \lambda_i^2} \end{aligned}$$

Now we apply the ϵ-net method to $\mathcal{S}^{n-1}$ with $\epsilon = \frac{1}{4}$, implying $|\mathcal{N}_\epsilon| < 9^n$. This gives

$$\|A\| \leq c_3\lambda_1(t + n\log 9) + c_4\sqrt{(t + n\log 9)\sum_i \lambda_i^2}.$$

Now when $t \leq \frac{n}{c_5}$. Hence, we have

$$\|A\| \leq c_6\lambda_1 n + c_7\sqrt{n\sum_{i=1}\lambda_i^2}.$$

Since $\sum_{i=1}\lambda_i^2 < \infty$, we can further simplify that $\|A\| \leq c_8 n$. □

Lemma 5.16. *Let* $\{\mathbf{w}_i\}_{i=1}^{\infty}$ *and* $\{\mathbf{u}_i\}_{i=1}^{\infty}$ *be two sequence of random vectors where*

the entry of each $\mathbf{w}_i \in \mathbb{R}^n$ *and* $\mathbf{u}_i \in \mathbb{R}^n$ *are i.i.d centered, unit variance* σ_w^2 *and* σ_u^2*-subgaussian random variables. Furthermore, let* $\{\lambda_i\}_{i=1}^{\infty}$ *be a sequence of non-negative, non-increasing numbers such that* $\sum_{i=1} \lambda_i^2 < \infty$*. Denote* $A = \sum_{i=1} \lambda_i \mathbf{w}_i \mathbf{u}_i^T$*. Let* $\mu_1(A) \geq, \cdots, \geq \mu_n(A)$ *be its eigenvalues. Then with probability at least* $1 - e^{-n/b_1}$*, we have*

$$-b_2\sigma_w\sigma_u n \leq \mu_n(A) \leq \mu_1(A) \leq b_2\sigma_w\sigma_u n.$$

Proof. The proof is similar to Lemma 5.15 by noticing that for any unit vector $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v}^T\mathbf{w}_i\mathbf{u}_i^T\mathbf{v}$ is a $c_1\sigma_w\sigma_u$-subexponential random variable according to Lemma 5.10. Applying Lemma 5.11 and ϵ-net argument to every unit vector in $\mathcal{S}^{n-1}$ yields our results. □

5.6.4 Moment Generating Function of Product of Random Variables

Lemma 5.17. *Let* x, y *be two independent random variables from* $\mathcal{N}(0,1)$*, then the moment generating function of the product is as following*

$$M_{xy}(t) = \frac{1}{\sqrt{1-t^2}},$$

provided $t < 1$.

Proof. By definition, assuming $t < 1$, we have

$$\begin{aligned}
\mathbb{E}(e^{txy}) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{txy} e^{-\frac{x^2}{2}} e^{-\frac{y^2}{2}} dxdy, \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}(x^2 - 2txy)\right) dxdy, \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-\frac{y^2}{2}} e^{\frac{t^2}{2}y^2} \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}(x-ty)^2\right) dxdy, \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}(1-t^2)y^2\right) dy, \\
&= \frac{1}{\sqrt{1-t^2}}.
\end{aligned}$$

□

5.6.5 Proof of Corollary 5.1

The proof is similar to the proof of Theorem 5.2 except we will use a different concentration inequality. As usual, we start with bias-variance decomposition. In the subgaussian scenario, the bias-viarance decomposition is exactly the same as that in the Gaussian case, with the only difference being the shape of the noise. As a result, we will adopt the same notation as in Section 5.4.2 except that the noise is now $\sigma_0^2\sigma_u^2$-subgaussian.

5.6.5.1 Upper Bound on Bias

$$\mathbf{B}_\xi = \beta_*^{\xi T}\Pi_\xi \left\{ \mathbb{E}_x \left(\mathbf{z}_x^\xi(\mathbf{W})\mathbf{z}_x^\xi(\mathbf{W})^T \right) - \frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi \right\} \Pi_\xi \beta_*^\xi,$$

where

$$\begin{aligned}\mathbb{E}_x \left(\mathbf{z}_x^\xi(\mathbf{W})\mathbf{z}_x^\xi(\mathbf{W})^T \right) - \frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi = \mathbb{E}_x &\left\{ \mathbf{z}_x(\mathbf{W})\mathbf{z}_x(\mathbf{W})^T - \frac{1}{n}\mathbf{Z}^T\mathbf{Z} \right\}, \\ &+\mathbb{E}_x(\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T) + \mathbb{E}_x(\boldsymbol{\xi}\mathbf{z}_x(\mathbf{W})^T) + \boldsymbol{\xi}\boldsymbol{\xi}^T, \\ &-\frac{1}{n}\mathbf{Z}^T\Xi - \frac{1}{n}\Xi^T\mathbf{Z} - \frac{1}{n}\Xi^T\Xi.\end{aligned}$$

We need to upper bound $\mathbb{E}_x(\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T)$, $\boldsymbol{\xi}\boldsymbol{\xi}^T$, $\frac{1}{n}\mathbf{Z}^T\Xi$ and $\frac{1}{n}\Xi^T\Xi$.

Firstly, for $\mathbb{E}_x(\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T)$, we have

$$\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T = \frac{\sigma_0}{s}\mathbf{W}^T D^{\frac{1}{2}} V_x \mathbf{u}^T = \frac{\sigma_0}{s}\sqrt{k(x,x)}\mathbf{v}\mathbf{u}^T.$$

where we recall $\mathbf{v} \in \mathbb{R}^s$ is a vector with each entry being i.i.d $\sim \mathcal{N}(0,1)$ and $\mathbf{u} \in \mathbb{R}^s$ is a vector with each entry being i.i.d σ_u^2-subgaussian. We can upper bound

$\mathbb{E}_x(\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T)$ with probability greater than $1-e^{-s/c_1}$ as

$$\begin{aligned}\left\|\mathbb{E}_x(\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T)\right\| &\leq \frac{\sigma_0}{s}\left\|\sqrt{\mathbb{E}_x(k(x,x))}\right\|\|\mathbf{w}\mathbf{u}^T\|,\\ &= \frac{\sigma_0}{s}\left\|\sqrt{\mathbb{E}_x(k(x,x))}\right\|\left|\sum_{i=1}^{s}u_iv_i\right|,\\ &\leq \frac{\sigma_0}{s}\sqrt{C_0}c_2 s, \quad \text{By Lemma 5.16 and let } n=1\\ &= c_3\sigma_0.\end{aligned}$$

Secondly, for $\boldsymbol{\xi}\boldsymbol{\xi}^T$, $\|\boldsymbol{\xi}\boldsymbol{\xi}^T\| = \boldsymbol{\xi}^T\boldsymbol{\xi} = \frac{\sigma_0^2}{s}\|\mathbf{u}\|^2 \leq c_4\sigma_0^2$ with probability greater than $1-e^{-s/c_5}$ by Lemma 5.12.

Moreover, for $\frac{1}{n}\mathbf{Z}^T\Xi$, using the rotation invariance for subgaussian distribution, we similarly have

$$\frac{1}{n}\mathbf{Z}^T\Xi = \frac{\sigma_0}{s\sqrt{n}}\sum_{i}^{n}\sqrt{\lambda_i}\mathbf{w}_i\mathbf{u}_i^T$$

Now each entry of $\mathbf{w}_i$ is 1-subgaussian and $\mathbf{u}_i$ is σ_u^2-subgaussian, applying Lemma 5.16, with probability greater than $1-e^{-s/c_6}$ we obtain

$$\left\|\frac{1}{n}\Xi^T\mathbf{Z}\right\| \leq \frac{\sigma_0}{s\sqrt{n}}c_7\sigma_u n\sqrt{\sum_i\hat{\lambda}_i} = c_8\sigma_0\frac{n}{s\sqrt{n}} \leq c_9\frac{\sigma_0}{\sqrt{s}}.$$

Finally, for $\frac{1}{n}\Xi^T\Xi$, by appealing to (Bartlett et al., 2020, Lemma 4), which is the subgaussian version of Lemma 5.5 and asserts that $\|\sum_i^n \mathbf{u}_i\mathbf{u}_i^t\| \leq c_{10}n$ with probability greater than $1-e^{-s/c_{11}}$, we can upper bound

$$\begin{aligned}\left\|\frac{1}{n}\Xi^T\Xi\right\| &= \left\|\frac{\sigma_0^2}{n}\sum_i^n\mathbf{u}_i\mathbf{u}_i^T\right\|,\\ &\leq \frac{\sigma_0^2}{n}\left\|\sum_i^n\mathbf{u}_i\mathbf{u}_i^T\right\|,\\ &\leq c_{12}\frac{\sigma_0^2 n}{ns} \leq c_{13}\frac{\sigma_0^2}{s}.\end{aligned}$$

Combining above results together, with probability greater than $1-\delta-e^{-n/c}$ we upper bound the bias as

$$\mathbf{B}_\xi \leq c\left\{\frac{\lambda_W}{s}\|\Sigma\|\sqrt{\log(\frac{14r(\Sigma)}{\delta})/n}+\sigma_0\right\}\|\Pi_\xi\|^2\|\beta_*^\xi\|^2.$$

5.6.5.2 Upper Bound on Variance

$$\begin{aligned}\mathbf{V}_\xi = &\frac{\sigma^2}{n}\text{Tr}\left\{\mathbb{E}_x\left[\mathbf{z}_x(\mathbf{W})\mathbf{z}_x(\mathbf{W})^T\right]\left(\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger\right\},\\ &+\frac{\sigma^2}{n}\text{Tr}\left\{\mathbb{E}_x\left[\mathbf{z}_x(\mathbf{W})\boldsymbol{\xi}^T\right]\left(\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger\right\},\\ &+\frac{\sigma^2}{n}\text{Tr}\left\{\mathbb{E}_x\left[\boldsymbol{\xi}\mathbf{z}_x(\mathbf{W})^T\right]\left(\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger\right\},\\ &+\frac{\sigma^2}{n}\text{Tr}\left\{\boldsymbol{\xi}\boldsymbol{\xi}^T\left(\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi\right)^\dagger\right\},\end{aligned}$$

Through similar analysis to the Gaussian case, use Eq. (5.10), we can also have that $\mu_n(\frac{1}{n}\mathbf{Z}_\xi^T\mathbf{Z}_\xi)\geq\frac{1}{a}n$. Following the similar argument as in the Gaussian case we can upper bound the variance with probability greater than $1-e^{-n/c}$

$$\mathbf{V}_\xi \leq c\sigma^2\text{Tr}(\Sigma)\frac{s}{n^2}.$$

5.6.6 Proof of Corollary 5.2

Proof. Proof of Corollary 5.2 is simply applying $\sigma_0^2=O(s^{-\alpha})$ into Eq. (5.11) and Eq. (5.12). □

Chapter 6 Conclusion

This thesis contributes towards the theoretical understanding of random feature methods in two folds. First, it treats the random feature models as a way to speed up kernel methods. As previously discussed, kernel methods often suffer from the massive computational cost which is typically on the order of $O(n^2)$ or even $O(n^3)$ (where n is the number of data points). This is prohibitive in the modern big data era. As a result, Rahimi and Recht (2007) propose the RFFs methods to speed up the computation. Despite its empirical success, the exact computation-accuracy trade-off of the RFFs methods remains unclear. In the first part of the thesis (Chapters 3 and 4), we tackle this problem from two different perspectives and provide a detailed analysis on the trade-off between the computational cost (the number of features required) and the statistical prediction accuracy (the learning rate) in the cases of both regression and classification. Apart from being a simply low rank approximation method, recent studies (Belkin et al., 2019b; Liao et al., 2020; Mei and Montanari, 2019) view the random feature methods as a two-layer neural network with fixed first layer weights. This view provides us with a flexible framework to theoretically analyze the generalization properties of neural networks and serves as the foundation towards explaining superior empirical performance of modern deep learning. The second part of our thesis (Chapter 5) adopts this view and attempts to understand the recent double descent phenomenon discovered in the context of deep learning. Our results explain how double descent happens as a result of the noise in the features under the random feature framework and potentially shed light on how to understand the phenomenon in the deep learning framework.

6.1 Summary of Contributions

RFFs are first introduced as an efficient way to approximate the kernel matrix and hence to speed up the computation of many kernel methods. As such, a clear understanding of the trade-off between computational cost and statistical accuracy is of paramount importance when employing RFFs in practice. To this end, Chapter 3 provides an initial analysis on this problem from the kernel matrix approximation perspective. Specifically, we first derive a precise relationship between the expected maximum kernel matrix error due to RFFs and the number of features, which demonstrates that the kernel matrix approximation $\tilde{\mathbf{K}}$ using RFFs is a consistent estimator of the original kernel matrix $\mathbf{K}$. Subsequently, we characterize how the excess learning risk of the RFFs estimator from KRR evolves as we increase the number of features. We are able to show that as long as the number of features is on the order of $O(n)$, the excess learning risk converges to zero at the same minimax rate as that in the case of the kernel ridge regression estimator. Hence, we prove that the RFFs regression estimator is also consistent. Finally, we apply the similar analysis to the more involved distribution regression setting and show that both the RFFs kernel matrix approximation and the RFFs estimator in distribution regression setting are consistent.

Despite the consistency proof, one problem of our analysis in Chapter 3 is that we are not able to theoretically verify the efficacy of RFFs estimators. In particular, whether RFFs can reduce computational cost without incurring any loss in prediction accuracy is not clear. In light of this, Chapter 4 is devoted to addressing this issue through directly analyzing the learning risk. We first devise a theoretical framework that allows us to conduct a unified analysis of the generalization properties of RFFs estimators from both regression and classification. In regression, we show that $\Omega(\sqrt{n}\log n)$ features are enough to guarantee the minimax learning rate

of the kernel ridge regression estimator if we use the plain RFFs sampling scheme. We further demonstrate that the required number of features can be reduced significantly when we use the leverage weighted sampling method. In addition, we prove that a sharp $O(1/n)$ learning rate is possible under the regression setting. In particular, RFFs regression estimators can obtain $O(1/n)$ learning rate with only $\Omega(\log n \log\log n)$ features if the eigenspectrum of the kernel Gram-matrix displays exponential decay and the leverage weighted RFFs sampling is used. Our theoretical framework can also extend to the study of the computation and accuracy trade-off in the classification setting with various loss functions such as Lipschitz continuous loss and 0-1 loss. For plain RFFs sampling method, our results improve upon the existing results when the eigenspectrum of the kernel matrix has finite rank or exhibits exponential decay. Moreover, when leverage weighted RFFs sampling is employed in the classification setting, we provide a detailed analysis for the computational cost and prediction accuracy trade-off. Our results demonstrate that the RFFs estimators can not only reduce the computation time ($\Omega(1)$ features when eigenspectrum has finite rank), but also enjoy a fast learning rate ($O(1/n)$). Finally, we propose a fast algorithm to generate feature samples that come from the approximated leverage score distribution. We empirically verify that this algorithm can provide a significant reduction in prediction time compared with the original RFFs method without incurring any extra loss. We also give a rigorous proof that demonstrates the trade-off between the computational cost and the statistical accuracy of the proposed algorithm.

In Chapter 5, we adopt the view that RFFs can be treated as the two-layer neural network with fixed first layer weights, and attempt to examine the conditions under which the double descent and benign overfitting phenomena arise. By incorporating the Gaussian feature noise ξ into our model, our first result states that ξ can serve as the implicit regularizer that prevents the divergence of the learning risk even in

the case where the number of parameters is much larger than that of the data points. We also provide a finite sample analysis of the excess learning risk for the MNLS estimator which demonstrates that it is possible for overparameterized estimator to obtain minimax optimal learning rate. Subsequently, we relax the Gaussian feature noise assumption into the case of subgaussian class of distributions, thereby extending our results into a more general setting. Finally, we derive the evolution of the upper bound of the excess learning risk and show that the learning risk undergoes a multiple descent, instead of a double descent behavior. While requiring a very weak assumption on the kernel structure, our results explain the possible driving force of the benign overfitting phenomenon under no specific assumption on the data generating distribution.

6.2 Future Research

Although we address some of the important questions for random feature methods, several problems on the theoretical properties of random features remain unclear. An example is whether we can obtain a sharper trade-off between the computational cost and the prediction accuracy of RFFs estimators in the classification setting when the plain RFFs sampling is used. Our previous results demonstrate that it is possible for the RFFs estimators to achieve the fast $O(1/\sqrt{n})$ or even $O(1/n)$ learning rate. However, to obtain these learning rate, one has to sample at least $\Omega(n)$ features using plain RFFs sampling. This result means that we would not asymptotically benefit from any computational savings, while empirically (Li et al., 2019b; Rahimi and Recht, 2007; Rudi et al., 2017) we observe that the RFFs can indeed dramatically reduce the computational cost even in the classification setting. Therefore, we lack a critical understanding of how RFFs can help to reduce the computational cost in the classification setting with plain RFFs sampling. In

addition, sampling features according to the leverage score can provide a promising way to further reduce the computational cost. However, it is generally costly to compute these leverage scores. As such, how to efficiently construct leverage scored features in the context of classification is an interesting future direction.

Furthermore, RFFs are widely adopted in hypothesis testing problems (Jitkrittum et al., 2017; Zhang et al., 2018), which is another frequent use of kernel methods. In particular, Zhang et al. (2018) provide an extensive study of the computational cost and testing power trade-off in the context of RFFs approximation. Through a variety of experiments, they are able to demonstrate that the RFFs approximation can give comparable testing performance with the original kernel hypothesis testing while significantly reducing the computation and memory requirement. However, a key question to the RFFs hypothesis testing is whether we could provide any theoretical guarantees of the testing power given a certain number of features used in the approximation. Currently, there are no explicit analyses on the computation and test power trade-off in literature and this can be an interesting future direction to pursue. In addition, our analysis reveals that the usage of weighted RFFs sampling based on the leverage scores allows one to sample less features while achieving the same learning rate in both regression and classification. However, the use of leverage score is not explored in the context of hypothesis testing. One could empirically explore whether weighted RFFs sampling can provide any computational gain in kernel hypothesis testing, and theoretically examine the computation and testing power trade-off under the weighted sampling scheme.

In Chapter 5, we utilize the random feature model as a proxy to two-layer neural network to study its generalization properties. Although we derive many theoretical results in the random feature regime, this regime has certain limitations. In particular, while assuming the first layer of weights as fixed, we omit the effect of the whole optimization procedure. However, optimizing the first layer weights is

known to play an important role in the prediction of neural network models. Hence, it is critical to investigate the neural network models with optimized features. Specifically, one would like to discover how double descent and benign overfitting arise in the optimized regime. Moreover, in the study of the overparameterized random feature model, we demonstrate that feature noise, whether it resides in the covariates itself or whether it is manually added by user, is the key driving force for the double descent and the benign overfitting phenomena, as long as the number of features is on the order $o(n^2)$. However, a recent study by Adlam and Pennington (2020a) discovers that the learning curve undergoes another descent beyond the $\Omega(n^2)$ regime. Hence, it would be interesting to understand how this happens and how we could explain this multiple descent phenomenon as the current regime only predicts the cases up to $o(n^2)$.

Bibliography

Adlam, B. and Pennington, J. (2020a). The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization. In *International Conference on Machine Learning*, pages 74–84. PMLR.

Adlam, B. and Pennington, J. (2020b). Understanding double descent requires a fine-grained bias-variance decomposition. *Advances in Neural Information Processing Systems*, 33.

Agrawal, R., Campbell, T., Huggins, J., and Broderick, T. (2019). Data-dependent compression of random features for large-scale kernel approximation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1822–1831. PMLR.

Alaoui, A. and Mahoney, M. W. (2015). Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, pages 775–783.

Alvarez, M., Rosasco, L., and Lawrence, N. (2012). Kernels for vector-valued functions: a review. *Foundations and Trends® in Machine Learning*, 4(3):195–266.

An, G. (1996). The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404.

Avron, H., Kapralov, M., Musco, C., Musco, C., Velingker, A., and Zandieh, A. (2017). Random Fourier features for kernel ridge regression: Approximation

bounds and statistical guarantees. In *Proceedings of the International Conference on Machine Learning*, pages 253–262.

Ba, J., Erdogdu, M., Suzuki, T., Wu, D., and Zhang, T. (2019). Generalization of two-layer neural networks: An asymptotic viewpoint. In *International conference on learning representations*.

Bach, F. (2013). Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pages 185–209.

Bach, F. (2017a). Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53.

Bach, F. (2017b). On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21):1–38.

Bartlett, P. L., Bousquet, O., Mendelson, S., et al. (2005). Local Rademacher complexities. *Annals of Statistics*, 33(4):1497–1537.

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156.

Bartlett, P. L., Long, P. M., Lugosi, G., and Tsigler, A. (2020). Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482.

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019a). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854.

Belkin, M., Hsu, D., and Xu, J. (2019b). Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571*.

Belkin, M., Ma, S., and Mandal, S. (2018). To understand deep learning we need to understand kernel learning. In *Proceedings of the International Conference on Machine Learning*, pages 541–549.

Berlinet, A. and Thomas-Agnan, C. (2011). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

Bishop, C. M. (1995). Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Bochner, S. (1932). Vorlesungen über Fouriersche Integrale. In *Akademische Verlagsgesellschaft*.

Bojarski, M., Choromanska, A., Choromanski, K., Fagan, F., Gouy-Pailler, C., Morvan, A., Sakr, N., Sarlos, T., and Atif, J. (2017). Structured adaptive and random spinners for fast machine learning computations. In *Artificial Intelligence and Statistics*, pages 1020–1029. PMLR.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Bottou, L. and Bousquet, O. (2007). The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*.

Bunea, F., Strimas-Mackey, S., and Wegkamp, M. (2020). Interpolation under latent factor regression models. *arXiv preprint arXiv:2002.02525*.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

Camoriano, R., Angles, T., Rudi, A., and Rosasco, L. (2016). Nytro: When subsampling meets early stopping. In *Artificial Intelligence and Statistics*, pages 1403–1411.

Caponnetto, A. and De Vito, E. (2007). Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368.

Caponnetto, A., Micchelli, C. A., Pontil, M., and Ying, Y. (2008). Universal multi-task kernels. *The Journal of Machine Learning Research*, 9:1615–1646.

Caponnetto, A. and Yao, Y. (2010). Adaptive rates for regularization operators in learning theory. *Analysis and Applications*, 8.

Carmeli, C., De Vito, E., Toigo, A., and Umanitá, V. (2010). Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications*, 8(01):19–61.

Carratino, L., Rudi, A., and Rosasco, L. (2018). Learning with sgd and random features. In *Advances in Neural Information Processing Systems*, pages 10213–10224.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chatterji, N. S. and Long, P. M. (2020). Finite-sample analysis of interpolating linear classifiers in the overparameterized regime. *arXiv preprint arXiv:2004.12019*.

Chinot, G. and Lerasle, M. (2020). Benign overfitting in the large deviation regime. *arXiv preprint arXiv:2003.05838*.

Choromanski, K., Rowland, M., and Weller, A. (2017). The unreasonable effectiveness of structured random orthogonal embeddings. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 218–227.

Choromanski, K. and Sindhwani, V. (2016). Recycling randomness with structure for sublinear time kernel expansions. In *International Conference on Machine Learning*, pages 2502–2510. PMLR.

Cortes, C., Mohri, M., and Talwalkar, A. (2010). On the impact of kernel approximation on learning accuracy. In *International Conference on Artificial Intelligence and Statistics*, pages 113–120.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Csorgo, S. and Totik, V. (1983). On how long interval is the empirical characteristic function uniformly consistent. *Acta Scientiarum Mathematicarum*, 45(1-4):141–149.

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

Dao, T., De Sa, C., and Ré, C. (2017). Gaussian quadrature for kernel features. *Advances in neural information processing systems*, 30:6109.

d'Ascoli, S., Refinetti, M., Biroli, G., and Krzakala, F. (2020). Double trouble in double descent: Bias and variance (s) in the lazy regime. *arXiv preprint arXiv:2003.01054*.

De Vito, E., Caponnetto, A., and Rosasco, L. (2005). Model selection for regularized least-squares algorithm in learning theory. *Foundations of Computational Mathematics*, 5(1):59–85.

Dheeru, D. and Karra Taniskidou, E. (2017). UCI machine learning repository.

Dhifallah, O. and Lu, Y. M. (2020). A precise performance analysis of learning with random features. *arXiv preprint arXiv:2008.11904*.

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

Doran, G. (2013). Distribution kernel methods for multiple-instance learning. In *AAAI*.

Doran, G., Muandet, K., Zhang, K., and Schölkopf, B. (2014). A permutation-based kernel conditional independence test. In *UAI*, pages 132–141.

Drineas, P., Magdon-Ismail, M., Mahoney, M. W., and Woodruff, D. P. (2012). Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13(1):3475–3506.

Drineas, P. and Mahoney, M. W. (2005). On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR.

Engl, H. W., Hanke, M., and Neubauer, A. (1996). *Regularization of inverse problems*, volume 375. Springer Science & Business Media.

Feng, C., Hu, Q., and Liao, S. (2015). Random feature mapping with signed

circulant matrix projection. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Feuerverger, A. and Mureika, R. A. (1977). The empirical characteristic function and its applications. *Annals of Statistics*, pages 88–97.

Flaxman, S. R., Wang, Y.-X., and Smola, A. J. (2015). Who supported obama in 2012?: Ecological inference through distribution regression. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 289–298. ACM.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin.

Fukumizu, K., Bach, F. R., and Jordan, M. I. (2009). Kernel dimension reduction in regression. *Annals of Statistics*, pages 1871–1905.

Fukumizu, K., Gretton, A., Sun, X., and Schölkopf, B. (2007). Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems*, volume 20, pages 489–496.

Fukumizu, K., Gretton, A., Sun, X., and Schölkopf, B. (2008). Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems*, pages 489–496.

Fukumizu, K., Song, L., and Gretton, A. (2011). Kernel bayes' rule. In *Advances in Neural Information Processing Systems*, pages 1737–1745.

Gärtner, T. (2003). A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1):49–58.

Geer, S. A. (2000). *Applications of empirical process theory*. Cambridge University Press.

Genton, M. G. (2001). Classes of kernels for machine learning: a statistics perspective. *Journal of machine learning research*, 2(Dec):299–312.

Gerace, F., Loureiro, B., Krzakala, F., Mézard, M., and Zdeborová, L. (2020). Generalisation error in learning with random features and the hidden manifold model. In *International Conference on Machine Learning*, pages 3452–3462. PMLR.

Gerfo, L. L., Rosasco, L., Odone, F., Vito, E. D., and Verri, A. (2008). Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897.

Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. (2021). Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029–1054.

Giné, E. and Zinn, J. (1984). Some limit theorems for empirical processes. *The Annals of Probability*, pages 929–989.

Gittens, A. and Mahoney, M. (2013). Revisiting the nyström method for improved large-scale machine learning. In *Proceedings of the International Conference on Machine Learning*, pages 567–575.

Goodfellow, I., Bengio, Y., and Courville, A. (2016a). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016b). *Deep learning*. MIT press Cambridge.

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., and Smola, A. J. (2006). A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems*, pages 513–520.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012).

A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773.

Hastie, T., Montanari, A., Rosset, S., and Tibshirani, R. J. (2019). Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*.

Hastie, T. J. (2017). Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge.

Hofmann, T., Schölkopf, B., and Smola, A. J. (2008). Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220.

Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, pages 8571–8580.

Jacot, A., Şimşek, B., Spadaro, F., Hongler, C., and Gabriel, F. (2020). Implicit regularization of random feature models. *arXiv preprint arXiv:2002.08404*.

Jitkrittum, W., Gretton, A., Heess, N., Eslami, S., Lakshminarayanan, B., Sejdinovic, D., and Szabó, Z. (2015). Kernel-based just-in-time learning for passing expectation propagation messages. In *Uncertainty in Artificial Intelligence-Proceedings of the 31st Conference, UAI 2015*, volume 31, pages 405–414. AUAI Press.

Jitkrittum, W., Xu, W., Szabó, Z., Fukumizu, K., and Gretton, A. (2017). A linear-time kernel goodness-of-fit test. In *Advances in Neural Information Processing Systems*, pages 262–271.

Kadri, H., Duflos, E., Preux, P., Canu, S., Rakotomamonjy, A., and Audiffren, J. (2016). Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17(20):1–54.

Kadri, H., Rakotomamonjy, A., Bach, F., and Preux, P. (2012). Multiple operator-valued kernel learning. In *Neural Information Processing Systems (NIPS)*.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*.

Kohler, M. and Krzyzak, A. (2001). Nonparametric regression estimation using penalized least squares. *IEEE Transactions on Information Theory*, 47(7):3054–3058.

Koltchinskii, V. (2011). *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media.

Le, Q., Sarlós, T., and Smola, A. (2013). Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85.

Li, J., Liu, Y., and Wang, W. (2019a). Distributed learning with random features. *arXiv preprint arXiv:1906.03155*.

Li, P. (2017). Linearized gmm kernels and normalized random fourier features. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 315–324.

Li, Z., Su, W., and Sejdinovic, D. (2020). Benign overfitting and noisy features. *arXiv preprint arXiv:2008.02901*.

Li, Z., Ton, J.-F., Oglic, D., and Sejdinovic, D. (2019b). Towards a unified analysis of random Fourier features. In *Proceedings of the International Conference on Machine Learning*, pages 3905–3914.

Li, Z., Ton, J.-F., Oglic, D., and Sejdinovic, D. (2021). Towards a unified analysis of random Fourier features. *Journal of Machine Learning Research*, 22(108):1–51.

Lian, H. (2007). Nonlinear functional models for functional responses in reproducing kernel hilbert spaces. *Canadian Journal of Statistics*, 35(4):597–606.

Liang, T., Rakhlin, A., et al. (2020a). Just interpolate: Kernel "ridgeless" regression can generalize. *Annals of Statistics*, 48(3):1329–1347.

Liang, T., Rakhlin, A., and Zhai, X. (2020b). On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels. In *Conference on Learning Theory*, pages 2683–2711. PMLR.

Liao, Z., Couillet, R., and Mahoney, M. W. (2020). A random matrix analysis of random fourier features: beyond the gaussian kernel, a precise phase transition, and the corresponding double descent. *arXiv preprint arXiv:2006.05013*.

Liu, F., Huang, X., Chen, Y., Yang, J., and Suykens, J. (2020). Random fourier features via fast surrogate leverage weighted sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4844–4851.

Liu, X., Si, S., Cao, Q., Kumar, S., and Hsieh, C.-J. (2019). Neural sde: Stabilizing neural ode networks with stochastic noise. *arXiv preprint arXiv:1906.02355*.

Liu, Y., Liu, J., and Wang, S. (2021). Effective distributed learning with random features: Improved bounds and algorithms. In *International Conference on Learning Representations*.

Lopez-Paz, D., Muandet, K., Schölkopf, B., and Tolstikhin, I. (2015). Towards a learning theory of cause-effect inference. In *Proceedings of the International Conference on Machine Learning*, pages 1452–1461.

Lyu, Y. (2017). Spherical structured feature maps for kernel approximation. In *International Conference on Machine Learning*, pages 2256–2264. PMLR.

Maaten, L., Chen, M., Tyree, S., and Weinberger, K. (2013). Learning with marginalized corrupted features. In *International Conference on Machine Learning*, pages 410–418. PMLR.

Mahdaviyeh, Y. and Naulet, Z. (2019). Risk of the least squares minimum norm estimator under the spike covariance model. *arXiv*, pages arXiv–1912.

Mahoney, M. W. and Drineas, P. (2009). CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702.

Mei, S., Misiakiewicz, T., and Montanari, A. (2021). Generalization error of random features and kernel methods: hypercontractivity and kernel matrix concentration. *arXiv preprint arXiv:2101.10588*.

Mei, S. and Montanari, A. (2019). The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*.

Mei, S., Montanari, A., and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671.

Mendelson, S. (2002). Improving the sample complexity using global data. *IEEE transactions on Information Theory*, 48(7):1977–1991.

Micchelli, C. A. and Pontil, M. (2005). On learning vector-valued functions. *Neural computation*, 17(1):177–204.

Minsker, S. (2017). On some extensions of bernstein's inequality for self-adjoint operators. *Statistics & Probability Letters*, 127:111–119.

Mitrovic, J., Sejdinovic, D., and Teh, Y.-W. (2016). Dr-abc: approximate bayesian computation with kernel-based distribution regression. In *Proceedings of the International Conference on Machine Learning*, pages 1482–1491.

Mourtada, J. (2019). Exact minimax risk for linear least squares, and the lower tail of sample covariance matrices. *arXiv preprint arXiv:1912.10754*.

Muandet, K., Fukumizu, K., Sriperumbudur, B., and Schölkopf, B. (2016). Kernel mean embedding of distributions: A review and beyond. *arXiv preprint arXiv:1605.09522*.

Muandet, K. and Schölkopf, B. (2013). One-class support measure machines for group anomaly detection. *arXiv preprint arXiv:1303.0309*.

Nyström, E. J. (1930). Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*.

Ong, C. S. and Canu, S. (2004). Regularization by early stopping. Technical report, Technical report, Computer Sciences Laboratory, RSISE, ANU.

Ong, C. S., Mary, X., Canu, S., and Smola, A. J. (2004). Learning with non-positive kernels. In *Proceedings of the International Conference on Machine learning*, page 81.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Póczos, B., Xiong, L., and Schneider, J. (2012). Nonparametric divergence estimation with applications to machine learning on distributions. *arXiv preprint arXiv:1202.3758*.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184.

Rahimi, A. and Recht, B. (2009). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in Neural Information Processing Systems*, pages 1313–1320.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press.

Richards, D., Rebeschini, P., and Rosasco, L. (2020). Decentralised learning with random features and distributed gradient descent. In *International Conference on Machine Learning*, pages 8105–8115. PMLR.

Rosasco, L., De Vito, E., and Verri, A. (2005). Spectral methods for regularization in learning theory. *DISI, Universita degli Studi di Genova, Italy, Technical Report DISI-TR-05-18*.

Rudi, A., Calandriello, D., Carratino, L., and Rosasco, L. (2018). On fast leverage score sampling and optimal learning. In *Advances in Neural Information Processing Systems*, pages 5672–5682.

Rudi, A., Camoriano, R., and Rosasco, L. (2015). Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665.

Rudi, A., Carratino, L., and Rosasco, L. (2017). Falkon: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, pages 3891–3901.

Rudi, A. and Rosasco, L. (2017). Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3218–3228.

Rudin, W. (2017). *Fourier analysis on groups*. Courier Dover Publications.

Saunders, C., Gammerman, A., and Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In *Proceedings of the International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann.

Schölkopf, B. and Smola, A. J. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.

Schölkopf, B., Smola, A. J., Bach, F., et al. (2002). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schölkopf, B., Tsuda, K., and Vert, J.-P. (2004). *Kernel methods in computational biology*. MIT press.

Sejdinovic, D., Strathmann, H., Garcia, M. L., Andrieu, C., and Gretton, A. (2014). Kernel adaptive metropolis-hastings. In *Proceedings of the International Conference on Machine Learning*, pages 1665–1673.

Shahrampour, S., Beirami, A., and Tarokh, V. (2018). On data-dependent random features for improved generalization in supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

Sinha, A. and Duchi, J. C. (2016). Learning kernels with random features. In *Advances in Neural Information Processing Systems*, pages 1298–1306.

Sirignano, J. and Spiliopoulos, K. (2020). Mean field analysis of neural networks: A

central limit theorem. *Stochastic Processes and their Applications*, 130(3):1820–1852.

Smola, A., Gretton, A., Song, L., and Schölkopf, B. (2007). A Hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pages 13–31. Springer.

Smola, A. J. and Schölkopf, B. (2000). Sparse greedy matrix approximation for machine learning. In *Proceedings of the International Conference on Machine Learning*.

Song, L. (2008). *Learning via Hilbert space embedding of distributions*. Citeseer.

Song, L., Fukumizu, K., and Gretton, A. (2013). Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *Signal Processing Magazine, IEEE*, 30(4):98–111.

Song, L., Huang, J., Smola, A., and Fukumizu, K. (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the International Conference on Machine Learning*, pages 961–968. ACM.

Song, L., Zhang, X., Smola, A., Gretton, A., and Schölkopf, B. (2008). Tailoring density estimation via reproducing kernel moment matching. In *Proceedings of the International conference on Machine learning*, pages 992–999. ACM.

Sriperumbudur, B. and Szabó, Z. (2015). Optimal rates for random Fourier features. In *Advances in Neural Information Processing Systems*, pages 1144–1152.

Sriperumbudur, B. K., Fukumizu, K., and Lanckriet, G. R. (2011). Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(Jul):2389–2410.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. (2010). Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561.

Steinwart, I. (2019). Convergence types and rates in generic karhunen-loève expansions with applications to sample path properties. *Potential Analysis*, 51(3):361–395.

Steinwart, I. and Christmann, A. (2008). *Support vector machines*. Springer Science & Business Media.

Sun, Y., Gilbert, A., and Tewari, A. (2018). But how does it work in theory? linear svm with random features. In *Advances in Neural Information Processing Systems*, pages 3379–3388.

Sutherland, D. J., Oliva, J. B., Póczos, B., and Schneider, J. (2016). Linear-time learning on distributions with approximate kernel embeddings. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2073–2079.

Sutherland, D. J. and Schneider, J. (2015). On the error of random Fourier features. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 862–871. AUAI Press.

Suzuki, T. (2020). Generalization bound of globally optimal non-convex neural network training: Transportation map estimation by infinite dimensional langevin dynamics. *arXiv preprint arXiv:2007.05824*.

Szabó, Z., Sriperumbudur, B., Póczos, B., and Gretton, A. (2016). Learning theory for distribution regression. *Journal of Machine Learning Research*, 17(152):1–40.

Tolstikhin, I., Sriperumbudur, B. K., and Muandet, K. (2017). Minimax estimation

of kernel mean embeddings. *Journal of Machine Learning Research*, 18(1):3002–3048.

Tropp, J. A. (2015). An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230.

Tsybakov, A. B. (2003). Optimal rates of aggregation. In *Learning theory and kernel machines*, pages 303–313. Springer.

Tsybakov, A. B. et al. (2004). Optimal aggregation of classifiers in statistical learning. *Annals of Statistics*, 32(1):135–166.

Tu, S., Roelofs, R., Venkataraman, S., and Recht, B. (2016). Large scale kernel learning using block coordinate descent. *arXiv preprint arXiv:1602.05310*.

Ullah, E., Mianjy, P., Marinov, T. V., and Arora, R. (2018). Streaming kernel pca with $\tilde{O}(\sqrt{n})$ random features. In *Advances in Neural Information Processing Systems*, pages 7311–7321.

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Vito, E. D., Rosasco, L., Caponnetto, A., Giovannini, U. D., and Odone, F. (2005). Learning from examples as an inverse problem. *Journal of Machine Learning Research*, 6(May):883–904.

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

Wang, S. and Zhang, Z. (2013). Improving cur matrix decomposition and the nyström approximation via adaptive sampling. *Journal of Machine Learning Research*, 14(1):2729–2769.

Wang, S. and Zhang, Z. (2014). Efficient algorithms and error analysis for the modified nystrom method. In *Artificial Intelligence and Statistics*, pages 996–1004.

Wang, Y. and Shahrampour, S. (2019). A general scoring rule for randomized kernel approximation with application to canonical correlation analysis. *arXiv preprint arXiv:1910.05384*.

Williams, C. K. and Seeger, M. (2001a). Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pages 682–688.

Williams, C. K. I. and Seeger, M. (2001b). Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems 13*.

Xiong, L., Póczos, B., Schneider, J., Connolly, A., and VanderPlas, J. (2011). Hierarchical probabilistic models for group anomaly detection. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 789–797.

Yang, J., Sindhwani, V., Avron, H., and Mahoney, M. (2014). Quasi-monte carlo feature maps for shift-invariant kernels. In *International Conference on Machine Learning*, pages 485–493. PMLR.

Yang, T., Li, Y.-F., Mahdavi, M., Jin, R., and Zhou, Z.-H. (2012). Nyström method vs random Fourier features: A theoretical and empirical comparison. In *Advances in Neural Information Processing Systems*, pages 476–484.

Yao, Y., Rosasco, L., and Caponnetto, A. (2007). On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315.

Yashima, S., Nitanda, A., and Suzuki, T. (2021). Exponential convergence rates of

classification errors on learning with sgd and random features. In *International Conference on Artificial Intelligence and Statistics*, pages 1954–1962. PMLR.

Yehudai, G. and Shamir, O. (2019). On the power and limitations of random features for understanding neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yu, F. X., Suresh, A. T., Choromanski, K., Holtmann-Rice, D., and Kumar, S. (2016). Orthogonal random features. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1983–1991.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. (2011). Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 804–813.

Zhang, Q., Filippi, S., Gretton, A., and Sejdinovic, D. (2018). Large-scale kernel methods for independence testing. *Statistics and Computing*, 28(1):113–130.

Zhang, Y., Duchi, J., and Wainwright, M. (2013). Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, pages 592–617.

Zhang, Y., Duchi, J., and Wainwright, M. (2015). Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16(1):3299–3340.

Zhuo, J., Zhu, J., and Zhang, B. (2015). Adaptive dropout rates for learning with corrupted features. In *IJCAI*, pages 4126–4133.