Random Features Methods in Supervised Learning

by

Yitong Sun

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Applied and Interdisciplinary Mathematics)
in The University of Michigan
2019

Doctoral Committee:

Professor Anna Gilbert, Co-Chair
Associate Professor Ambuj Tewari, Co-Chair
Assistant Professor Laura Balzano
Professor Alfred Hero
Professor Mark Rudelson

Yitong Sun

syitong@umich.edu

ORCID iD: 0000-0002-6715-478X

In memory of my dad

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my two advisors, Anna Gilbert and Ambuj Tewari. They lead me into the fascinating field of random features, guide my research work with their knowledge and experience, and always have confidence in me. Without their help, I can never imagine to complete the degree and be ready for my future research career. I would also like to thank my committee members, Mark Rudelson, Alfred Hero and Laura Balzano.

I want to thank all the people who offer their support to me in the darkest period of my life. It is their kindnesses that help me overcome sorrow and move forward. I am always grateful to Wei Huang and Jun Zhang. Their wise and sincere advices played a crucial role in my decisions on continuely pursuing the research career in applied math.

I am fortunate to have Feng Wei and Yebin Tao as my closest friends these years at Ann Arbor. I learned so much skills and knowledge from them. Thank you to Dejiao Zhang for selflessly sharing her research experience and ideas with me, and taking me to a broader research area.

I am so blessed to meet my wife Wen Cui. No matter what happens, I know that she will always be there for me.

Finally, I would like to say thank you to my parents. They taught me to value hard working, to embrace challenges and to be myself. Their supports make it possible for me to realize my dream.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

ABSTRACT

Kernel methods and neural networks are two important schemes in the supervised learning field. The theory of kernel methods is well understood, but their performance in practice, particularly on large-size datasets, is not as good as neural networks. In contrast, neural networks are the most popular method in today's machine learning with a lot of fascinating applications, but even basic theoretical properties about neural networks, such as the universal consistency or sample complexity, have not been understood. Random features methods approximate kernel methods and resolve the scalability issues. Meanwhile, they have a deep connection with neural networks. Some empirical studies demonstrate the competitive performance of random features methods, but the theoretical guarantees on the performance are not available.

This thesis presents theoretical results on two aspects of random features method: the generalization performance and the approximation properties. We first study the generalization error of random features support vector machines using tools from the statistical learning theory. Then we establish the fast learning rate of random Fourier features corresponding to the Gaussian kernel, with the number of features far less than the sample size. This justifies the computational advantage of random features over kernel methods from the theoretical aspect. As an effort in exploring the possibility of designing random features, we then study the universality of random features and show that the random ReLU features method can be used in supervised

learning tasks as a universally consistent method. The depth separation result and the multi-layer approximation result point out the limitation of random features methods and shed light on the advantage of deep architectures.

CHAPTER I

Introduction

1.1 Motivation and Background

Supervised learning is one of the main fields of machine learning. It dates back to the pre-computer era as known as regression. The main goal of a supervised learning algorithm is to infer the target function using finitely many labeled data. As enormous data and computation power are available today, people have a stronger demand for supervised learning algorithms that can handle,

1. huge size of data,
2. high dimensional data,
3. and highly non-linear target functions without much prior knowledge.

For example, assigning correct labels to images from ImageNet is nowadays an important benchmark test for supervised learning algorithms. This dataset contains more than 14 million labeled images. Each image has more than 256 by 256 pixels. And there are no good theories on manually designing the map between images and their labels.

In general, to find the solution, a supervised learning algorithm first selects a family of functions parametrized in some way, and then adjusts the parameters of the model using empirical risk minimization. To understand the performance of a

supervised learning algorithm, we must answer the following questions:

1. What is the family of functions that can be learned by the algorithm?
2. How do the dimension and size of the training set affect the performance of the solution?
3. And how do the dimension and size of the dataset affect the computational cost of training and testing?

Vapnik developed a complete framework for studying these questions, and proposed a class of powerful supervised learning algorithms, kernel support vector machines[37]. Before the revival of neural networks, kernel support vector machines were popular for their impressive performance in practice and well-understood theory. Nowadays they are still competitive on tasks for which the training sets have size ranging from thousands to 10 thousands. A main drawback of kernel support vector machines, however, is their scalability. When a kernel support vector machine is trained on a dataset with m samples, it requires m^2 iterations to compute the Gram matrix by evaluating the kernel function over every pair of samples. In practice, to avoid occupying m^2 storage, each entry of the Gram matrix will be re-evaluated when queried and this costs $\Omega(m^2)$ operation time. Since the solution of a kernel support vector machine has the form of $\sum_{i=1}^{m} c_i k(x_i, \cdot)$, it requires m iterations to process a new data point in the inference stage. This is not acceptable for most modern machine learning applications, where models are trained on huge training sets like ImageNet and have to process vast data once deployed. In contrast, for a neural network with a fixed architecture, the number of iterations scales linearly with respect to the sample size during training, and the time cost to process a new data point in the inference stage is only relevant to the size of the neural network but irrelevant to the training sample size.

Though kernel support vector machines have above drawbacks in practice, they are still appealing from a theoretical point of view. The optimization problem to be solved in the training process of kernel support vector machines is a convex problem, and thus the global optimal solutions are guaranteed to be found by the algorithm, while it is still unclear whether the solution is close to the global optimum in the case of neural networks, since the training process involves solving a non-convex optimization problem. As people seek a method that possesses the benefits from both sides, random features methods catch people's attention.

Generally speaking, using random features in a supervised learning task includes following steps: first selecting a set of non-linear functions $\{\phi_{\omega_i}\}_{i=1}^N$, which are defined on the domain of data, from a family of functions $\{\phi_\omega\}_{\omega\in\Omega}$ according to a certain probability distribution over Ω, then mapping each data point x to a vector $\phi_N(x) \coloneqq [\phi_{\omega_i}(x)]_{i=1}^N$, and finally running a linear regression or classification algorithm on top of new feature vectors and labels, $\{\phi_N(x_i), y_i\}_{i=1}^m$. This idea dates back to decades ago when people searched for a better way to obtain a neural network model than the backpropagation method. When the non-linear functions $\{\phi_{\omega_i}\}_{i=1}^N$ are selected randomly and then fixed in the training phase, the process of adjusting the coefficients on top of the random nodes is a convex optimization problem and thus more tools are available to solve it (see [18], [16] and references therein). Rahimi and Recht showed in their seminal work that certain types of random features can also be treated as approximations to kernel methods [29]. Since then, more and more researchers have viewed random features method as a way to accelerate kernel methods. Some experiments on large-size datasets confirm that the random features method in supervised learning tasks can achieve comparable performance to deep neural nets [17, 9].

To understand the performance of the random features method in supervised learning tasks, we need to answer the three questions brought up at the beginning of this section. In other words, we need to study the generalization, approximation and optimization properties of algorithms using random features. The training phase of the classification or regression algorithms based on random features is not different from linear support vector machines or ridge regressions, and some powerful optimization methods like stochastic dual coordinate descent have been intensively studied. We direct readers interesting in this topic to the relevant references [32, 14]. This dissertation will focus on the generalization and approximation aspects of random features method. In particular, we study the generalization performance of random features methods on classification tasks and prove that the fast learning rate is achievable with the number of features being far less than the number of samples. This result together with previous works on the performance of random features methods on regression tasks justifies the computational advantage of random features over kernel methods. We then explore the possibility of choosing different feature maps other than the well-known random Fourier features. We prove several sufficient conditions for the universality of hypothesis classes of random features with neural network type feature maps. We also show how the functions in the reproducing kernel Hilbert space induced by the random ReLU features can be approximated by ReLU networks with bounded weights. As part of comparison with the deep networks, we study the depth separation and multi-layer networks approximation results for the composition of functions in the reproducing kernel Hilbert space.

The organization of the dissertation will be as follows: in the rest of the first chapter, we will review related references on the generalization and approximation properties of random features methods; in Chapter II, we will present our main

theoretical and experimental results on the generalization performance of random features method on classification tasks; in Chapter III, we will discuss the universality and related quantitative approximation results of random features; we will close this dissertation with the discussion on future works in Chapter IV.

1.2 Generalization Performance of Random Features Methods

For the rest of the dissertation, we will always use N for the number of random features and m for the number of samples. In our theoretical work, we always consider support vector machines on top of random features in binary classification tasks and ridge regression in regression tasks unless stated otherwise. We will call the former one random features support vector machine (RFSVM) and the latter one random features kernel ridge regression (RFKRR). The abbreviation RFKRR comes from [30], which we will discuss in this section.

Despite competitive practical performance shown in experiments, there is a lack of theoretical guarantees for the learning rate of supervised learning algorithms based on random features. In the paper proposing random features methods, Rahimi and Recht only gave the approximation gap of order $O(1/\sqrt{N})$ between the Gram matrices generated by N random features and the corresponding kernels. In a follow-up paper [29], they obtained a risk gap of order $O(1/\sqrt{N})$ between the best classifier of random features methods and that of kernel support vector machines. Although the order of the error bound matches the underlying approximation gap, it is too pessimistic to explain the actual computational benefits of random features method in practice. For example, without knowledge of data distribution, the generalization error of kernel support vector machines is of order $O(1/\sqrt{m})$. According to the risk gap of order $O(1/\sqrt{N})$ for random features methods, we have to use at least

m random features to achieve a comparable performance guarantee. However, when the number of random features is the same with the sample size, the computational benefits brought by random features methods do not exist any more.

[8] and [36] considered the performance of RFSVMs as a perturbed optimization problem. The dual form of a kernel support vector machine (KSVM) is a constrained quadratic optimization problem determined by the Gram matrix generated by the kernel function on the samples. Although the maximizer of a quadratic function depends continuously on the quadratic form, its dependence is weak and thus, both papers failed to obtain an informative bound on the excess risk of RFSVMs in classification problems. Moreover, such an approach requires the RFSVM and KSVM that we want to compare share the same hyper-parameters. This assumption is problematic because the optimal configuration of hyper-parameters for RFSVMs is not necessarily the same as that for the corresponding KSVMs. In fact, since the random features methods lead to finite dimensional hypothesis class once the random features are selected, they are less vulnerable to overfitting than kernel methods. Therefore, in this dissertation, we will treat RFSVMs more like an independent learning model instead of just an approximation to KSVMs.

In the regression setting, the learning rate of RFKRR was studied by [30] under the assumption that the regression function is in the reproducing kernel Hilbert space (RKHS), namely the *realizable* case. They show that the uniform feature sampling only requires $O(\sqrt{m}\log(m))$ features to achieve an $O(1/\sqrt{m})$ risk with respect to the squared loss. They further show that a data-dependent sampling can achieve a rate of $O(1/m^{\alpha})$, where $1/2 \leq \alpha \leq 1$, with even fewer features, when the regression function is sufficiently smooth and the spectrum of the kernel operator induced by the random features decays sufficiently fast. However, the method leading to these results

depends on the closed form of the linear least squares solution, and thus we cannot easily extend these results to non-smooth loss functions used in the classification case. [3] recently shows that for any given approximation accuracy, the number of random features required is determined by the leverage score, which is further upper bounded by the degrees of freedom of the kernel operator, when the random feature is optimized. This result is crucial for the sample complexity analysis of RFSVMs, though not many details are provided on this topic in Bach's work.

In Chapter II, we investigate the performance of RFSVMs formulated as a regularized optimization problem on classification tasks. In contrast to the slow learning rate in previous results by [29] and [3], we show, for the first time, that RFSVMs can achieve fast learning rate with far fewer features than the number of samples when the optimized features are available, and therefore the potential computational benefits of RFSVMs on classification tasks are justified.

1.3 Approximation Properties of Random Features Methods

Another question related to the random features methods is what feature maps and feature distributions can be gainfully used for supervised learning tasks. Random Fourier features and random binning features proposed in [29] all originate from such popular kernels as Gaussian or Laplacian. People are interested in these random features because the corresponding RKHSs are dense in the space of continuous functions under the supremum norm [25]. As a standalone learning method, we want flexibility in choosing the feature map and parameter distribution for such purposes as to lower computational cost or to inject prior knowledge. In Chapter III, we consider the random features using the following form of feature maps $\sigma(\omega \cdot x + b)$, where (ω, b) are parameters chosen randomly according to a certain distribution, and

$\sigma(\cdot)$ is a nonlinear function. We call this type of random features "neural network type", since they result in solutions to supervised learning tasks in the form of neural networks. And hence, random features of neural network type provide a foundation for the comparision between kernel methods and neural networks. We are particularly interested in using the rectified linear unit (ReLU) as the feature map, because ReLUs are widely used in deep neural networks as activation functions. Note that we do not evaluate the derivatives of feature maps in random features methods, so the advantage of ReLUs in preventing vanishing or exploding gradient does not play a role in our case. However, using ReLUs in the random features methods may still provide some computational advantages. The feature vector associated with ReLU feature map is always sparser than that of sigmoidal or sinusoidal nodes, and faster to evaluate, since each node either outputs 0 or identity.

When the feature map is chosen to be $\mathrm{ReLU}(\omega \cdot \mathbf{x} + b)$ where the weight vector (ω, b) obeys standard Gaussian over $\mathbb{R}^{d+1}$ or uniform distribution over $\mathbb{S}^d$, the induced kernel is one of the arccos kernels and can be written in a closed form [7]. When some other distributions or feature maps are considered, there may not be a good closed form for the induced kernels. In these cases, to justify the use of a random features method, we need to answer the question:

For a given feature map of the form $\sigma(\omega \cdot \mathbf{x} + b)$ and a given distribution of ω and b, under what conditions it is guaranteed that there exists a linear combination of randomly chosen $\sigma(\omega_i \cdot \mathbf{x} + b_i)$'s that is a good approximator to a target continuous function?

A related concept to this question is universality. A supervised learning algorithm or equivalently its hypothesis class is called universal, if the hypothesis class of this algorithm is dense in the space of continuous functions. From classic results

we know that neural networks for any non-polynomial activation functions are universal [23], but it requires an appropriate setup of weights for the neural networks to be good approximators. Hence, it does not imply that by randomly sampling inner weights, with high probability there exists a linear combination of chosen random nodes approximating the target continuous function well. [3] analyzed the approximation property of the arccos kernels. Proposition 3 in [3] implies the universality of arccos kernels by explicitly constructing functions in the RKHS of arccos kernels as approximators to Lipschitz functions. However this result heavily relies on the uniform parameter distribution over the unit sphere and only considers homogeneous non-decreasing feature maps.

In Chapter III, we answer the question we asked above. The first step is to establish the universality of the corresponding RKHS induced by random features. The general theory on the universality of RKHSs has been studied by [25], where the universality of Gaussian and Laplacian kernels are proved. we use tools from functional analysis to provide a set of sufficient conditions on the feature map σ and the distribution of (ω, b) for the universality of the corresponding kernels.

On top of the universality of the induced kernels, the approximation capability of random features methods can be characterized by [4], which shows how well the linear combination of random features approximates functions in the RKHS. [15] proved that for any continuous function, there exists a linear combination of randomly selected nodes approximating it, as long as σ is bounded and continuous, and the probability distribution over (ω, b) is absolutely continuous. Our results reproduce his result with a much simpler proof, and extend it to unbounded feature maps.

As an application of our results on the approximation properties of random fea-

tures, we show that the random ReLU features method is universally consistent. We compare the performance of random ReLU features to random Fourier features on several benchmark datasets. We observe that random ReLU features method can achieve an accuracy similar to random Fourier features method on the classification tasks but requires a shorter training and testing time.

As an extension of our interest in the RKHS induced by the random ReLU features, we further study the composition of functions in the RKHS. [12] shows that 3-layer ReLU networks with polynomial many activation nodes can express functions that can never be approximated by any 2-layer ReLU networks with polynomial many activation nodes. This phenomenon is called depth separation. It partly explains the advantage of deep networks over the shallow ones. [22] shows a similar depth separation result for functions in Barron's class and their compositions. Following these works, we show a depth separation result for functions in the RKHS induced by random ReLU features. This result shows that compositions of functions from the RKHSs induced by the random ReLU features are substantially more complicated than functions in the RKHS; see Section 3.6 for the statement. We further prove that compositions of functions from the RKHS can be approximated by multilayer ReLU networks, with all weights bounded by constants depending on the RKHS norm of the components of the target function. The depth separation result and the approximation result together show the potential limit of random ReLU features compared to deep ReLU networks.

We designed a synthetic dataset according to the construction in the depth separation result and use it to demonstrate the difference of performance of random ReLU features, 2-layer neural networks and 3-layer neural networks. The experiment clearly shows the limit of shallow models. And surprisingly the improvement

of optimized inner weights over the randomly chosen ones is not as significant as the improvement brought by the depth.

CHAPTER II

The Upper Bound on the Learning Rate of RFSVM

In this chapter, we study the learning rate of random features methods in two scenarios, the realizable case and the unrealizable case. In the realizable case, we assume that there exists a good approximator to the Bayes classifier in the RKHS of the random feature, or that the Bayes classifier itself is in it. In the unrealizable case, we take into consideration the risk gap between the Bayes classifier and the good approximator in the RKHS. In particular, we will show the following three results:

1. We prove that under Massart's low noise condition, with optimized random features, RFSVM can achieve the learning rate of $\tilde{O}(m^{-\frac{c_2}{1+c_2}})$ [1], with $\tilde{O}(m^{\frac{2}{2+c_2}})$ features when the Bayes classifier belongs to the RKHS of a kernel whose spectrum decays polynomially ($\lambda_i = O(i^{-c_2})$). When the spectrum of the kernel operator decays sub-exponentially, the learning rate can be improved to $\tilde{O}(1/m)$ with $\tilde{O}(\ln^d(m))$ features, where d is the dimension of raw features of data.
2. When the data-label distribution satisfies the separation condition; that is, when the support of the distribution of the two classes of points are apart by a positive distance, we prove that the RFSVM using optimized random features corresponding to the Gaussian kernel can achieve a learning rate of $\tilde{O}(1/m)$ with $\tilde{O}(\ln^{2d}(m))$ number of features.

[1] $\tilde{O}(n)$ represents a quantity less than $Cn\log^k(n)$ for some k.

3. Our theoretical analysis suggests reweighting random features before training. We confirm its benefit in our experiments over synthetic datasets.

We begin in Section 2.1 with a brief introduction to RKHS, random features and the formulation of support vector machines, and set up the notations we use throughout the chapter. In Section 2.2, we state the main assumptions on the data and feature distributions and related lemmas. The main theoretical results are presented in Section 2.3 and 2.4. And in Section 2.5, we verify the performance of RFSVM in experiments. In particular, we show the improvement brought by the reweighted feature selection algorithm.

2.1 Kernel Support Vector Machines and Random Features

Throughout this chapter, a labeled data point is a point (x, y) in $\mathcal{X} \times \{-1, 1\}$, where $\mathcal{X}$ is a bounded subset of $\mathbb{R}^d$. $\mathcal{X} \times \{-1, 1\}$ is equipped with a probability distribution $\mathbb{P}$. We use $\mathbb{P}_{\mathcal{X}}$ to denote the marginal distribution on the data space.

2.1.1 Kernels and Random Features

A positive definite kernel function $k(x, x')$ defined on $\mathcal{X} \times \mathcal{X}$ determines the unique corresponding reproducing kernel Hilbert space (RKHS), denoted by $(\mathcal{F}_k, \|\cdot\|_k)$. A map ϕ from the data space $\mathcal{X}$ to a Hilbert space H such that

$$\langle \phi(x), \phi(x') \rangle_H = k(x, x') \tag{2.1}$$

for any $x, x' \in \mathcal{X}$ is called a feature map of k and H is called a feature space. For any $f \in \mathcal{F}_k$, there exists an $h \in H$ such that $\langle h, \phi(x) \rangle_H = f(x)$, and the infimum of the norms of all such h's is equal to $\|f\|_k$. On the other hand, given any feature map ϕ into H, a kernel function is defined by Equation 2.1, and we call $\mathcal{F}_k$ the RKHS corresponding to ϕ, denoted by $\mathcal{F}_\phi$.

A common choice of feature space consists of L^2 functions of a probability space (Ω, ω, ν). For any probability density function $q(\omega)$ with respect to ν, $\phi(\omega; x)/\sqrt{q(\omega)}$, whenever $q(\omega) \neq 0$, with probability measure $q(\omega)\mathrm{d}\nu(\omega)$ defines the same kernel function with the feature map $\phi(\omega; x)$ under the distribution ν. (ϕ, ν) is called a random feature. One can sample the values of $\phi(\omega; x)$, the image of x under the feature map ϕ in the L^2 space, at points $\{\omega_1, \ldots, \omega_N\}$ i.i.d. according to the probability distribution ν to approximately represent $\phi(\cdot; x)$. Then the vector

$$\frac{1}{\sqrt{N}} \left(\phi(\omega_1; x), \ldots, \phi(\omega_N; x)\right)^\top$$

in $\mathbb{R}^N$ is called a random feature vector of x, denoted by $\phi_N(x)$. The corresponding kernel function determined by ϕ_N is denoted by k_N.

A well-known construction of random features is the random Fourier features proposed by [28]. The feature map is defined as follows,

$$\begin{aligned} \phi : \mathcal{X} &\to L^2(\mathbb{R}^d, \nu) \oplus L^2(\mathbb{R}^d, \nu) \\ x &\mapsto \left(\cos\left(\omega \cdot x\right), \sin\left(\omega \cdot x\right)\right) . \end{aligned}$$

And the corresponding random feature vector is

$$\phi_N(x) = \frac{1}{\sqrt{N}} \left(\cos\left(\omega_1 \cdot x\right), \cdots, \cos\left(\omega_N \cdot x\right), \sin\left(\omega_1 \cdot x\right), \cdots, \sin\left(\omega_N \cdot x\right)\right)^\top ,$$

where ω_i's are sampled according to ν. For the random Fourier feature, different choices of ν define different translation invariant kernels by Bochner's theorem [28]. For example, when ν is the normal distribution with mean 0 and variance γ^{-2}, the induced kernel function is the Gaussian kernel with bandwidth parameter γ,

$$k_\gamma(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\gamma^2}\right) .$$

Equivalently, we may consider the feature map $\phi_\gamma(\omega; x) \coloneqq \phi(\omega/\gamma; x)$ with ν being standard normal distribution.

A more general and more abstract feature map can be constructed using an orthonormal set of $L^2(\mathcal{X}, \mathbb{P}_\mathcal{X})$. Given the orthonormal set $\{e_i\}$ consisting of uniformly bounded functions, and a nonnegative sequence $(\lambda_i) \in \ell^1$, we can define a feature map

$$\phi(\omega; x) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} e_i(x) e_i(\omega) \,,$$

with feature space $L^2(\mathcal{X}, \omega, \mathbb{P}_\mathcal{X})$. The corresponding kernel is given by $k(x, x') = \sum_{i=1}^{\infty} \lambda_i e_i(x) e_i(x')$. The feature map and the kernel function are well defined because of the boundedness assumption on $\{e_i\}$. This representation can be obtained for any continuous kernel function on a compact set by Mercer's Theorem ([20]).

Every positive definite kernel function k satisfying $\int k(x, x) \, \mathrm{d}\mathbb{P}_\mathcal{X}(x) < \infty$ defines an integral operator on $L^2(\mathcal{X}, \mathbb{P}_\mathcal{X})$ by

$$\begin{aligned} \Sigma : L^2(\mathcal{X}, \mathbb{P}_\mathcal{X}) &\to L^2(\mathcal{X}, \mathbb{P}_\mathcal{X}) \\ f &\mapsto \int_\mathcal{X} k(x, t) f(t) \, \mathrm{d}\mathbb{P}_\mathcal{X}(t) \,. \end{aligned}$$

Σ is of trace class with trace norm $\int k(x, x) \, \mathrm{d}\mathbb{P}_\mathcal{X}(x)$. When the integral operator is determined by a random feature (ϕ, ν), we denote it by $\Sigma_{\phi,\nu}$, and the ith eigenvalue in a descending order by $\lambda_i(\Sigma_{\phi,\nu})$. Whenever the probability distribution ν is obvious in the context, we use Σ_ϕ for convenience. Note that the regularization parameter in the support vector machine is also denoted by λ but with no subscripts. The decay rate of the spectrum of Σ_ϕ plays an important role in the analysis of learning rate of random features method.

2.1.2 Supervised Learning and Support Vector Machines

Given m samples $\{(x_i, y_i)\}_{i=1}^m$ generated i.i.d. by $\mathbb{P}$ and a function $f : \mathcal{X} \to \mathbb{R}$, usually called a decision function in the machine learning context, the empirical and

the expected risks with respect to the loss function ℓ are defined by

$$R_m^\ell(f) \coloneqq \frac{1}{m}\sum_{i=1}^{m} \ell\left(y_i, f\left(x_i\right)\right) \quad \text{and} \quad R_{\mathbb{P}}^\ell(f) \coloneqq \mathbb{E}_{(x,y)\sim\mathbb{P}}\ell\left(y, f\left(x\right)\right) ,$$

respectively. A function f^* that minimizes $R_{\mathbb{P}}^\ell$ is called a Bayes decision function. A supervised learning task seeks for a good decision function $\hat{f}$ that achieves a small *excess risk* $R_{\mathbb{P}}^\ell(\hat{f}) - R_{\mathbb{P}}^\ell(f^*)$. According to whether the label y is ordinal or categorical, a supervised learning task is categorized into regression or classification, respectively. The 0-1 loss is commonly used to measure the performance of binary classifiers:

$$\ell^{0-1}(y, f(x)) = \begin{cases} 1 & \text{if} \quad yf(x) \leq 0; \\ 0 & \text{if} \quad yf(x) > 0. \end{cases}$$

For regression tasks, the most common loss function is the squared loss.

A Bayes classifier for the binary classification is given by

$$f_{\mathbb{P}}^*(x) \coloneqq \operatorname{sgn}\left(\mathbb{E}[y \mid x]\right) .$$

Note that the Bayes classifier may not be unique in this case. In particular, the values of the classifier at data points labeled by 1 or -1 with probability of $1/2$ do not affect its performance. And also the values of the classifier at points with marginal probability 0 do not affect its performance either.

The set of decision functions, $\mathcal{F}$, that is searched by a supervised learning algorithm is called the hypothesis class. And to find the good decision function with samples, the most common method is to minimize the empirical risk. However, for a binary classification problem, it is hard to find the global minimizer of the empirical risk because the loss function is discontinuous and non-convex. A popular surrogate loss function in practice is the hinge loss: $\ell^h(f) = \max(0, 1 - yf(x))$. Its clipped

version is useful in the analysis of the learning rate:

$$\ell^1(y, f(x)) = \begin{cases} 2 & \text{if} \quad yf(x) \leq -1; \\ 1 - yf(x) & \text{if} \quad -1 < yf(x) \leq 1; \\ 0 & \text{if} \quad 1 \leq yf(x). \end{cases}$$

And we have that $\inf_f R^1_{\mathbb{P}}(f) = R^1_{\mathbb{P}}(f^*_{\mathbb{P}}) = 2R^{0-1}_{\mathbb{P}}(f^*_{\mathbb{P}})$ and

$$R^1_{\mathbb{P}}(f) - \inf_f R^1_{\mathbb{P}}(f) \geq R^{0-1}_{\mathbb{P}}(f) - R^{0-1}_{\mathbb{P}}(f^*_{\mathbb{P}}) \,.$$

See [34] for more details.

Different supervised learning algorithms assume different underlying hypothesis classes. One commonly used hypothesis class is the class of linear functions on $\mathcal{X}$. Obviously we should not expect good performance from linear classifiers on $\mathcal{X}$ when the points of different classes are not separable by a hyperplane; for example, when the data distribution is as shown in Figure 2.2. The idea to solve linearly non-separable problems is to map the dataset to a higher dimensional space, actually an infinite dimensional space in most cases, so that the labels can be linearly separated in the new feature space. The new space is usually a Hilbert space, and the inner product can be encoded as the kernel function on $\mathcal{X} \times \mathcal{X}$. The structure among the features is thus fully characterized by the value of the kernel function evaluated at each pair of sample points. However, in most cases, for example when the kernel is universal, the capacity of the hypothesis class is so large that any configuration of labels can be linearly classified, assuming that no different labels are assigned to the same data point. Such a resultant classifier often has a poor performance on the test set, because it encodes all the noise or irrelevant factors from the training set. This is the overfitting phenomenon in statistics, and it can be quantified by the so-called generalization errors, $R^\ell_m(f) - R^\ell_{\mathbb{P}}(f)$.

To prevent overfitting, we need to constrain the complexity of the hypothesis class. A direct way to do this is to force the norm of functions we consider in the hypothsis class less than some constant chosen as a hyper-parameter. Equivalently, we can add a regularizer into the optimization objective with a scalar multiplier λ. There are many different ways to choose the norm constraint or the form of regularizer, served for different properties desired from the solution. Throughout this dissertation, we only consider the commonly used ℓ^2 regularization. Therefore, the solution of the binary classification problem is given by minimizing the following objective

$$R^h_{m,\lambda}(f) = R^h_m(f) + \frac{\lambda}{2} \|f\|^2_{\mathcal{F}} ,$$

over a hypothesis class $\mathcal{F}$.

When $\mathcal{F}$ is the RKHS of some kernel function, the algorithm described above is called kernel support vector machine. Note that for technical convenience, we do not include the bias term in the formulation of decision function so that all these functions are from the RKHS instead of the product space of RKHS and $\mathbb{R}$ (see Chapter 1 of [34] for more explanation of such a convention). Note that $R_{m,\lambda}$ is strongly convex and thus the infimum will be attained by some function in $\mathcal{F}$. We denote it by $f_{m,\lambda}$. By the representer theorem, we have that

$$f_{m,\lambda}(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x) .$$

Therefore, the optimization problem is reduced to a quadratic problem over $\mathbb{R}^m$ and the Gram matrix $G_k \coloneqq [k(x_i, x_j)]_{i,j}$ is required.

When random features ϕ_N and the corresponding RKHS are considered, we add N into the subscripts of the notations defined above to indicate the number of random features. For example $\mathcal{F}_N$ for the RKHS, $f_{N,m,\lambda}$ for the solution of the optimization problem.

2.2 Assumptions on the Data and Feature Distributions

In this section we state the main assumptions on the distribution of data and features, which are required for the results in the next sections. The first assumption is about the ambiguity of labels. Since we assume that the data-label pair (x, y) obeys a joint distribution $\mathbb{P}$, it is possible that different y's are assigned to the same x in a dataset. If a data point is assigned to two labels with $1/2$ probabilities for each, the classifier should ignore it in the training phase. However, since we only have access to samples instead of the true distribution, there is usually no means to identify noisy data points from clean ones. Therefore, noisy labels will be very hard to learn. The following property quantifies the level of noisiness and will affect the upper bound of learning rate.

Assumption II.1. *There exists $V \geq 2$ such that for all x,*

$$|\mathbb{E}_{(x,y)\sim\mathbb{P}}[y \mid x]| \geq 2/V\,.$$

This assumption is called Massart's low noise condition in many references (see for example [19]). When $V = 2$, almost all the data points have deterministic labels. Therefore it is easier to learn the true classifier based on observations. In the proof, Massart's low noise condition guarantees the variance condition ([34])

$$\mathbb{E}[(\ell^h(f(x)) - \ell^h(f^*_{\mathbb{P}}(x)))^2] \leq V(R^h(f) - R^h(f^*_{\mathbb{P}}))\,,$$

which is a common requirement for the fast learning rate results. Massart's condition is an extreme case of a more general low noise condition, called Tsybakov's condition, and our main results can be extended to Tsybakov's condition. However, for the simplicity of the statement of the results, we only consider Massart's condition.

The second assumption is about the quality of random features. [4] proved the

following useful theorem, which we will use to verify the finite approximatability of RKHSs induced by random features.

Theorem II.2 (Proposition 1 of [3]). *For $\mu > 0$ and a random feature (ϕ, ν), let*

$$d_{\max}(1, \mu) := \sup_{\omega \in \mathbb{R}^{d+1}} \|(\Sigma + \mu I)^{-1/2} \phi(\cdot; \omega)\|^2_{L_2(\mathbb{P})} ,$$

where $\Sigma : L_2(\mathbb{P}) \to L_2(\mathbb{P})$ is defined by

$$\Sigma f = \int k_{\phi,\nu}(x, y) f(y) \, \mathrm{d}\mathbb{P}(y) .$$

For $\delta > 0$, when

$$N \geq 5 d_{\max}(1, \mu) \log \left(\frac{16 d_{\max}(1, \mu)}{\delta} \right) , \tag{2.2}$$

there exists $\beta \in \mathbb{R}^N$ with norm less than 2, such that

$$\sup_{\|f\|_{\phi,\nu} \leq 1} \|f - \beta \cdot \phi_N(\cdot)\|_{L^2(\mathbb{P})} \leq 2\sqrt{\mu} , \tag{2.3}$$

with probability greater than $1 - \delta$.

Based on this theorem, we make the following definition.

Assumption II.3. *A feature map $\phi : \mathcal{X} \to L^2(\Omega, \omega, \nu))$ is called optimized if there exists a small constant μ_0 such that for any $\mu \leq \mu_0$,*

$$\sup_{\omega \in \Omega} \|(\Sigma + \mu I)^{-1/2} \phi(\omega; x)\|^2_{L^2(\mathbb{P})} = \mathrm{tr}(\Sigma(\Sigma + \mu I)^{-1}) = \sum_{i=1}^{\infty} \frac{\lambda_i(\Sigma)}{\lambda_i(\Sigma) + \mu} .$$

For any given μ, the quantity on the left hand side of the inequality is called the maximal leverage score with respect to μ, which is directly related to the number of features required to approximate a function in the RKHS of (ϕ, ν). The quantity on the right hand side is called degrees of freedom by [3] and effective dimension by [30], denoted by $d(\mu)$. As shown in the following examples, whatever the RKHS is, we can always construct an optimized feature map for it.

Assume that a feature map $\phi : (X) \to L^2(\Omega, \omega, \nu)$ satisfies that $\phi(\omega; x)$ is bounded for all ω and x. We can always convert it to an optimized feature map using the method proposed by [3]. We rephrase it using our notation as follows.

Denote

$$p(\omega) = \frac{\|(\Sigma + \mu I)^{-1/2}\phi(\cdot;\omega)\|^2_{L^2(\mathcal{X},\mathbb{P})}}{\int_\Omega \|(\Sigma + \mu I)^{-1/2}\phi(\cdot;\omega)\|^2_{L^2(\mathcal{X},\mathbb{P})}\, \mathrm{d}\nu(\omega)}\,. \tag{2.4}$$

Since ϕ is bounded, its L^2 norm is finite. The function p defined above is a probability density function with respect to ν. Then the new feature map is given by $\tilde{\phi}(\omega; x) = \phi(\omega; x)/\sqrt{p(\omega)}$ together with the measure $p(\omega)\mathrm{d}\nu(\omega)$. Note that this transformation of random features does not change the kernel and the corresponding integral operator Σ. With $\tilde{\phi}$, we have

$$\begin{aligned}\sup_{\omega\in\Omega} \left\|(\Sigma + \mu I)^{-1/2}\tilde{\phi}(\cdot;\omega)\right\|^2 &= \sup_{\omega\in\Omega} \frac{\left\|(\Sigma + \mu I)^{-1/2}\phi(\cdot;\omega)\right\|^2}{p(\omega)} \\ &= \int_\Omega \|(\Sigma + \mu I)^{-1/2}\phi(\cdot;\omega)\|^2_{L^2(\mathcal{X},\mathbb{P})}\, \mathrm{d}\nu(\omega) \\ &= \mathrm{tr}(\Sigma(\Sigma + \mu I)^{-1})\,.\end{aligned}$$

When the feature map is constructed mapping into $L^2(\mathcal{X}, \mathbb{P})$ as described in Section 2.1, it is also optimized. Indeed, we can compute

$$\begin{aligned}\sup_{\omega\in\mathcal{X}} \left\|(\Sigma + \mu I)^{-1/2}\phi(\cdot;\omega)\right\|^2 &= \sup_{\omega\in\mathcal{X}} \left\|\sum_{i=1}^{\infty} \frac{\sqrt{\lambda_i}}{\sqrt{\lambda_i + \mu}} e_i(\cdot)\right\|^2 \\ &= \sum_{i=1}^{\infty} \frac{\lambda_i}{\lambda_i + \mu}\,.\end{aligned}$$

When a feature map is optimized, it is easy to control its leverage score by the decay rate of the spectrum of Σ, as described below.

Definition II.4. We say that the spectrum of $\Sigma : L^2(\mathcal{X}, \mathbb{P}) \to L^2(\mathcal{X}, \mathbb{P})$ decays at a polynomial rate if there exist $c_1 > 0$ and $c_2 > 1$ such that

$$\lambda_i(\Sigma) \le c_1 i^{-c_2}\,.$$

We say that it decays sub-exponentially if there exist $c_3, c_4 > 0$ such that

$$\lambda_i(\Sigma) \leq c_3 \exp(-c_4 i^{1/d}) .$$

The decay rate of the spectrum of Σ characterizes the capacity of the RKHS to find a solution, which further determines the number of random features required in the learning process. Indeed, when the feature map is optimized, the number of features required to approximate a function in the RKHS with accuracy $O(\sqrt{\mu})$ is upper bounded by $O(d(\mu)\ln(d(\mu)))$. When the spectrum decays polynomially, the degrees of freedom $d(\mu)$ is $O(\mu^{-1/c_2})$, and when it decays sub-exponentially, $d(\mu)$ is $O(\ln^d(c_3/\mu))$. These are described in the following lemma.

Lemma II.5. *If $\lambda_i(\Sigma) \leq c_1 i^{-c_2}$, where $c_2 > 1$, we have*

$$d(\mu) \leq \frac{2c_2}{c_2 - 1} \left(\frac{c_1}{\mu}\right)^{1/c_2} , \tag{2.5}$$

for $\mu < c_1$.

If $\lambda_i(\Sigma) \leq c_3 \exp(-c_4 i^{1/d})$, we have

$$d(\mu) \leq 5 c_4^{-d} \ln^d(c_3/\mu) , \tag{2.6}$$

for $\mu < c_3 \exp\left(-\left(c_4 \vee \frac{1}{c_4}\right) d^2\right)$.

Proof. Both results make use the following observation:

$$d(\mu) = \sum_{i=1}^{\infty} \frac{\lambda_i}{\lambda_i + \mu} \leq m_\mu + \frac{1}{\mu} \sum_{m_\mu + 1}^{\infty} \lambda_i , \tag{2.7}$$

where $m_\mu = \max\{i : \lambda_i \leq \mu\}|$.

When $\lambda_i \leq c_1 i^{-c_2}$, denote $t_\mu = (c_1/\mu)^{1/c_2}$ and then $m_\mu = \lfloor t_\mu \rfloor$. For the tail part,

$$\begin{aligned} \frac{1}{\mu} \sum_{m_\mu + 1}^{\infty} \lambda_i &\leq 1 + \frac{1}{\mu} \int_{t_\mu}^{\infty} c_1 x^{-c_2} \, \mathrm{d}x \\ &\leq 1 + \frac{1}{c_2 - 1} \left(\frac{c_1}{\mu}\right)^{\frac{1}{c_2}} . \end{aligned}$$

Combining them together, when $c_1/\mu > 1$, the constant 1 can be absorbed by the second term with a coefficient 2.

When $\lambda_i \le c_3 \exp(-c_4 i^{1/d})$, denote $t_\mu = \frac{1}{c_4^d} \ln^d \left(\frac{c_3}{\mu}\right)$, and then $m_\mu = \lfloor t_\mu \rfloor$. For the tail part, we need to discuss different situations.

First, if $d = 1$, then we directly have

$$\begin{aligned} \frac{1}{\mu} \sum_{m_\mu+1}^{\infty} \frac{\lambda_i}{\lambda_i + \mu} &\le \frac{1}{\mu} \left(\mu + \int_{t_\mu}^{\infty} c_3 \exp(-c_4 x) \, \mathrm{d}x \right) \\ &= 1 + \frac{1}{c_4} \,. \end{aligned}$$

When $\mu < c_3 \exp(-(c_4 \vee \frac{1}{c_4}))$, we can combine these terms into $3t_\mu$.

Second, if $d \ge 2$, when $\mu \le c_3 \exp(-c_4 e)$, we have that

$$\exp(-c_4 x^{1/d}) \le \exp(-c_4 \frac{t_\mu^{1/d}}{\ln t_\mu} \ln x) = x^{-c_4 \frac{t_\mu^{1/d}}{\ln t_\mu}} \,. \tag{2.8}$$

Then,

$$\begin{aligned} \frac{1}{\mu} \sum_{m_\mu+1}^{\infty} \lambda_i &\le 1 + \frac{1}{\mu} \int_{t_\mu}^{\infty} c_3 \exp(-c_4 x^{-1/d}) \, \mathrm{d}x \\ &\le 1 + \frac{c_3}{\mu} \int_{t_\mu}^{\infty} x^{-c_4 \frac{t_\mu^{1/d}}{\ln t_\mu}} \\ &= 1 + \frac{t_\mu}{c_4 \frac{t_\mu^{1/d}}{\ln t_\mu} - 1} \,. \end{aligned}$$

When $c_4 \ge 1$, we may assume that $\mu \le c_3 \exp(-c_4 d^2)$, and then

$$c_4 \frac{t_\mu^{1/d}}{\ln t_\mu} - 1 \ge \frac{c_4 d^2}{2d \ln d} \ge \frac{4}{3} \,. \tag{2.9}$$

So the upper bound has the form $5t_\mu$.

When $c_4 < 1$, we may assume that $\mu \le c_3 \exp(-d^2/c_4)$, and then

$$c_4 \frac{t_\mu^{1/d}}{\ln t_\mu} - 1 \ge \frac{d^2/c_4}{2d \ln(d/c_4)} \ge \frac{4}{3} \,. \tag{2.10}$$

So the upper bound also has the form $5t_\mu$. □

Examples on the kernels with polynomial and sub-exponential spectrum decays can be found in [3]. In particular, we know the random Fourier feature with the Gaussian feature distribution and sub-Gaussian data distribution induces a kernel operator with sub-exponentially decaying spectrum; see Section 2.4. The random ReLU feature with the uniform feature distribution and the uniform data distribution over the sphere induces a kernel operator with polynomially decaying spectrum; see Section 3.4.

2.3 Learning Rate in Realizable Cases

With these preparations, we can state our first theorem.

Theorem II.6. *Assume that $\mathbb{P}$ satisfies Assumption II.1, and the random feature (ϕ, ν) satisfies Assumption II.3 and $|k_{\phi,\nu}| \leq 1$. Then for any $g \in \mathcal{F}_\phi$ with $\|g\|_\phi \leq R$ and $\|g - f_{\mathbb{P}}^*\|_{L^2(\mathbb{P})} \leq \epsilon$, when the spectrum of Σ_ϕ decays polynomially, by choosing*

$$\lambda = m^{-\frac{c_2}{2+c_2}}$$

$$N = 10 C_{c_1,c_2} m^{\frac{2}{2+c_2}} (\log(32 C_{c_1,c_2} m^{\frac{2}{2+c_2}}) + \log(1/\delta)) \,,$$

we have

$$R_{\mathbb{P}}^{0-1}(f_{N,m,\lambda}) - R_{\mathbb{P}}^{0-1}(f_{\mathbb{P}}^*) \leq C_{c_1,c_2,V,R} m^{-\frac{c_2}{2+c_2}} \left((\log^{3/2}(1/\delta) + \log(m)) \right) + 6\epsilon \,,$$

with probability $1 - 4\delta$.

When the spectrum of Σ_ϕ decays sub-exponentially, by choosing

$$\lambda = 1/m$$

$$N = 25 C_{d,c_4} \log^d(m) (\log(80 C_{d,c_4} \log^d(m)) + \log(1/\delta)) \,,$$

we have

$$R_{\mathbb{P}}^{0-1}(f_{N,m,\lambda}) - R_{\mathbb{P}}^{0-1}(f_{\mathbb{P}}^*) \leq C_{c_3,c_4,d,R,V} \frac{1}{m} \left(\log^{d+2}(m) + \log^{3/2}(1/\delta) \right) + 6\epsilon \,,$$

with probability $1 - 4\delta$ *when* $m \geq \exp((c_4 \vee \frac{1}{c_4})d^2/2)$.

When the Bayes classifier belongs to the RKHS of the feature map, we have $g = f_{\mathbb{P}}$ and $\epsilon = 0$, so this theorem characterizes the learning rate of RFSVM in realizable cases. The statement connects to the unrealizable case when $f_{\mathbb{P}}$ does not belong to the RKHS but an approximator g together with R and ϵ can be constructed.

For polynomially decaying spectrum, when $c_2 > 2$, we get a learning rate faster than $1/\sqrt{m}$. [30] obtained a similar fast learning rate for kernel ridge regression with random features, assuming polynomial decay of the spectrum of Σ_ϕ and the existence of a minimizer of the risk in $\mathcal{F}_\phi$. Theorem II.6 extends their result to classification problems and also the case when Σ_ϕ has an exponentially decaying spectrum. For RFKRR, the rate faster than $O(1/\sqrt{m})$ will be achieved whenever $c_2 > 1$, and the number of features required is only square root of our result. This is mainly caused by the fact that the loss function in the regression problem is the squared loss. For classification problems, we need calibration between the target 0-1 loss and the surrogate loss. The calibration function for the squared loss is also squared, while that for the hinge loss is linear. The result for the case where Σ_ϕ has a sub-exponentially decaying spectrum shows that RFSVM can achieve $\tilde{O}(1/m)$ with only $\tilde{O}(\log^d(m))$ features. The learning rate for sub-exponentially decaying spectrum has not been investigated for RFKRR. Note however that when d is large, the sub-exponential case requires a large number of samples, even possibly larger than the polynomial case. We therefore suspect that there is considerable room for improving our analysis of high dimensional data in the sub-exponential decay case. In particular, removing the exponential dependence on d under reasonable assumptions is an interesting direction for future work.

The proof of Theorem II.6 follows from the sample complexity analysis scheme

used by [34] and the approximation error result of [3]. The sample complexity analysis or the control of the generalization error relies on the analysis of the local Rademacher complexity.

The empirical Rademacher complexity for a class of functions $\mathcal{F}$ is defined as

$$\mathfrak{R}_{\mathcal{D}}(\mathcal{F}) \coloneqq \mathbb{E}\left[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \epsilon_i f(x_i)\middle|\mathcal{D}\right] ,$$

where $\mathcal{D}$ is a set of m samples drawn i.i.d. according to the data distribution $\mathbb{P}_{\mathcal{X}}$ and ϵ_i's are i.i.d. symmetric Bernoulli, also known as Rademacher, random variables. The (expected) Rademacher complexity $\mathfrak{R}_m(\mathcal{F})$ is defined as the expectation of $\mathfrak{R}_{\mathcal{D}}(\mathcal{F})$ over $\mathcal{D}$. The Rademacher complexity is a common tool in statistical learning theory. Similar to the VC-dimension, it quantifies the capacity of the hypothesis class. In a coarse analysis of the generalization error based on the Rademacher complexity, the class of functions considered is the ball of a fixed constant radius in the RKHS. To conduct a finer analysis, we need to consider the Rademacher complexity for functions with small risks. This leads to the definition of local Rademacher complexity, see the statement of Theorem II.7 for detailed description and [34] for more explanations.

The fast rate is achieved due to the fact that the Rademacher complexity of the RKHS of N random features with regularization parameter λ is only $O(\sqrt{N\log(1/\lambda)})$, while N and $1/\lambda$ need not be too large to control the approximation error when the optimized features are available.

The relation between the sample complexity and the local Rademacher complexity of the hypothesis class is described by Theorem II.7. In the theorem, ℓ^1 is the clipped hinge loss, and $R^* \coloneqq \inf_f R^h_{\mathbb{P}}(f)$. We also have that $R^* = R^h_{\mathbb{P}}(f^*_{\mathbb{P}})$.

Theorem II.7. *(Theorem 7.20 in [34]) For a RKHS $\mathcal{F}$, denote* $\inf_{f\in\mathcal{F}} R^1_{\mathbb{P},\lambda}(f) - R^*$

by r^*. For $r > r^*$, consider the following function classes

$$\mathcal{F}_r := \{f \in \mathcal{F} \mid R^1_{\mathbb{P},\lambda}(f) - R^* \leq r\}$$

and

$$\mathcal{H}_r := \{\ell^1 \circ f - \ell^1 \circ f^*_{\mathbb{P}} \mid f \in \mathcal{F}_r\}\,.$$

Assume that there exists $V \geq 1$ such that for any $f \in \mathcal{F}$,

$$\mathbb{E}_{\mathbb{P}}(\ell^1 \circ f - \ell^1 \circ f^*_{\mathbb{P}})^2 \leq V(R^1_{\mathbb{P}}(f) - R^*)\,.$$

If there is a function $\varphi_m : [0,\infty) \to [0,\infty)$ such that $\varphi_m(4r) \leq 2\varphi_m(r)$ and $\mathfrak{R}_m(\mathcal{H}_r) \leq \varphi_m(r)$ for all $r \geq r^*$, Then, for any $\delta \in (0,1]$, $f_0 \in \mathcal{F}$ with $\|\ell^{\text{hinge}} \circ f_0\|_\infty \leq B_0$, and

$$r > \max\left\{30\varphi_m(r), \frac{72V\ln(1/\delta)}{m}, \frac{5B_0\ln(1/\delta)}{m}, r^*\right\}\,,$$

we have

$$R^1_{\mathbb{P},\lambda}(f_{m,N,\lambda}) - R^* \leq 6\left(R^h_{\mathbb{P},\lambda}(f_0) - R^*\right) + 3r$$

with probability greater than $1 - 3\delta$.

To establish the fast rate of RFSVM using the theorem above, we must understand the local Rademacher complexity of RFSVM; that is, we must find a formula for $\varphi_m(r)$. The variance condition Equation 2.2 is satisfied under Assumption II.1. With this variance condition, we can upper bound the Rademacher complexity of RFSVM in terms of the number of features N and the regularization parameter λ. It is particularly important to have $1/\lambda$ inside the logarithm function.

First, we will need the summation version of Dudley's inequality using entropy number defined below, instead of covering number.

Definition II.8. For a semi-normed space $(E, \|\cdot\|)$, we define its (dyadic) entropy number by

$$e_n(E, \|\cdot\|) := \inf\left\{\varepsilon > 0 : \exists s_1, \ldots, s_{2^{n-1}} \in B^1 \text{ s.t. } B^1 \subset \bigcup_{i=1}^{2^{n-1}} B(s_i, \varepsilon)\right\}\,,$$

where B^1 is the unit ball in E and $B(a,r)$ is the ball with center at a and radius r.

Note that functions in $\mathcal{H}_r$ are basically compositions of functions in $\mathcal{F}_r$ and the loss function, so we have the following lemma. $\|\cdot\|_{L_2(D)}$ is the semi-norm defined by $\|\cdot\|_{L_2(D)} := (\frac{1}{m}\sum_i f^2(x_i))^{1/2}$.

Lemma II.9. $e_i(\mathcal{H}_r, \|\cdot\|_{L_2(D)}) \leq e_i(\mathcal{F}_r, \|\cdot\|_{L_2(D)})$

Proof. Assume that T is an ϵ-covering over $\mathcal{F}_r$ with $|T| = 2^i$. By definition $\epsilon \geq e_i(\mathcal{F}_r, \|\cdot\|_{L_2(D)})$. Then $T' = \ell^1 \circ T - \ell^1 \circ f^*_{\mathbb{P}}$ is a covering over $\mathcal{H}_r$. For any f and g in $\mathcal{F}_r$,

$$\left\|\ell^1 \circ f - \ell^1 \circ g\right\|_{L_2(D)} \leq 1 \cdot \|f - g\|_{L_2(D)} \ ,$$

because ℓ^1 is 1-Lipschitz. And hence the radius of the image of an ϵ-ball under ℓ^1 is less than ϵ. Therefore $\ell^1 \circ T - \ell^1 \circ f^*_{\mathbb{P}}$ is an ϵ-covering over $\mathcal{H}_r$ with cardinatily 2^i and $\epsilon \leq e_i(\mathcal{F}_r, \|\cdot\|_{L_2(D)})$. By taking infimum over the radius of all such T and T', the statement is proved. □

Now we need to give an upper bound for the entropy number of $\mathcal{F}_r$ with semi-norm $\|\cdot\|_{L_2(D)}$ using a volumetric estimate.

Lemma II.10. $e_i(\mathcal{F}_r, \|\cdot\|_{L_2(D)}) \leq 3(2r/\lambda)^{1/2} 2^{-i/2N}$.

Proof. Since $\mathcal{F}$ consists of functions

$$f(x) = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} w_i \phi(\omega_i; x) \, ,$$

under the semi-norm $\|\cdot\|_{L_2(D)}$ it is isometric with the N-dimensional subspace U of $\mathbb{R}^m$ spanned by the vectors

$$\{(\phi(\omega_1; x_i), \ldots, \phi(\omega_N; x_i))\}_{i=1}^m$$

for fixed m samples. For each $f \in \mathcal{F}_r$, we have $R^1_{\mathbb{P},\lambda}(f) - R^* \leq r$, which implies that $\|f\|_{\mathcal{F}} \leq (2r/\lambda)^{1/2}$. By the property of RKHS, we get

$$|f(x)| \leq \|f\|_{\mathcal{F}} \|k(x,\cdot)\|_{\mathcal{F}} \leq \left(\frac{2r}{\lambda}\right)^{1/2} \cdot 1 \,,$$

where we use the fact that $k(x,\cdot)$ is the evaluation functional in the RKHS.

Denote the isomorphism from $\mathcal{F}$ (modulo the equivalent class under the semi-norm) to U by I. Then we have

$$I(\mathcal{F}_r) \subset B^m_\infty \left(\left(\frac{2r}{m\lambda}\right)^{1/2} \right) \cap U \subset B^m_2 \left(\left(\frac{2r}{\lambda}\right)^{1/2} \right) \cap U \,.$$

The intersection region can be identified as a ball of radius $(2r/\lambda)^{1/2}$ in $\mathbb{R}^N$. Its entropy number by volumetric estimate is given by

$$e_i \left(B^N_2 \left(\left(\frac{2r}{\lambda}\right)^{1/2} \right), \|\cdot\|_2 \right) \leq 3 \left(\frac{2r}{\lambda}\right)^{1/2} 2^{-\frac{i}{N}} \,.$$

□

With the lemmas above, we can get an upper bound on the entropy number of $\mathcal{H}_r$. However, we should note that such an upper bound is not the best when i is small. Because the ramp loss ℓ^1 is bounded by 2, the radius of $\mathcal{H}_r$ with respect to $\|\cdot\|_{L_2(D)}$ is bounded by 1, which is irrelevant with r/λ. This observation will give us finer control on the Rademacher complexity.

Lemma II.11. *Assume that $\lambda < 1/2$. Then*

$$\mathfrak{R}_D(\mathcal{H}_r) \leq \sqrt{\frac{(\ln 16) N \log_2 1/\lambda}{m}} \left(3\sqrt{2}\rho + 18\sqrt{r} \right) \,,$$

where $\rho = \sup_{h \in \mathcal{H}_r} \|h\|_{L_2(D)}$.

Proof. By Theorem 7.13 in [34], we have

$$\mathfrak{R}_D(\mathcal{H}_r) \leq \sqrt{\frac{\ln 16}{m}} \left(\sum_{i=1}^{\infty} 2^{i/2} e_{2^i}(\mathcal{H}_r \cup \{0\}, \|\cdot\|_{L_2(D)}) + \sup_{h \in \mathcal{H}_r} \|h\|_{L_2(D)} \right) \,.$$

It is easy to see that $e_i(\mathcal{H}_r \cup \{0\}) \leq e_{i-1}(\mathcal{H}_r)$ and $e_0(\mathcal{H}_r) \leq \sup_{h\in\mathcal{H}_r} \|h\|_{L_2(D)}$. Since $e_i(\mathcal{H}_r)$ is a decreasing sequence with respect to i, together with the lemma above, we know that

$$e_i(\mathcal{H}_r) \leq \min\left\{ \sup_{h\in\mathcal{H}_r} \|h\|_{L_2(D)}, 3\left(\frac{2r}{\lambda}\right)^{1/2} 2^{-\frac{i}{N}} \right\} .$$

Even though the second one decays exponentially, it may be much greater than the first term when $2r/\lambda$ is huge for small i's. To achieve the balance between these two bounds, we use the first one for first T terms in the sum and the second one for the tail. So

$$\mathfrak{R}_D(\mathcal{H}_r) \leq \sqrt{\frac{\ln 16}{m}} \left(\sup_{h\in\mathcal{H}_r} \|h\|_{L_2(D)} \sum_{i=0}^{T-1} 2^{i/2} + 3\left(\frac{2r}{\lambda}\right)^{1/2} \sum_{i=T}^{\infty} 2^{i/2} 2^{-\frac{2^i-1}{N}} \right) .$$

The first sum is $\frac{\sqrt{2}^T-1}{\sqrt{2}-1}$. When T is large enough, the second sum is upper bounded by the integral

$$\int_{T-1}^{\infty} 2^{x/2} 2^{-2^x-1/N} \,\mathrm{d}x \leq \frac{6N}{2^{T/2}} 2^{-\frac{2^T}{4N}} .$$

To make the form simpler, we bound $\frac{\sqrt{2}^T-1}{\sqrt{2}-1}$ by $3\cdot 2^{T/2}$, and denote $\sup_{h\in\mathcal{H}_r} \|h\|_{L_2(D)}$ by ρ. Taking T to be

$$\log_2\left(2N\log_2\left(\frac{1}{\lambda}\right)\right) .$$

we get the upper bound of the form

$$\mathfrak{R}_D(\mathcal{H}_r) \leq \sqrt{\frac{\ln 16}{m}} \left(3\rho\sqrt{2N\log_2\frac{1}{\lambda}} + \frac{18\sqrt{Nr}}{\log_2(1/\lambda)} \right) .$$

When $\lambda < 1/2$, $\log_2 1/\lambda > 1$, so we can further enlarge the upper bound to the form

$$\mathfrak{R}_D(\mathcal{H}_r) \leq \sqrt{\frac{(\ln 16)N\log_2 1/\lambda}{m}} \left(3\sqrt{2}\rho + 18\sqrt{r} \right) .$$

□

Next lemma analyzes the expected Rademacher complexity for $\mathcal{H}_r$.

Lemma II.12. *Assume $\lambda < 1/2$ and $\mathbb{E}h^2(x,y) \le V\mathbb{E}h(x,y)$. Then*

$$\mathfrak{R}_m(\mathcal{H}_r) \le C_1\sqrt{\frac{N(V+1)\log_2(1/\lambda)}{m}}\sqrt{r} + C_2\frac{N\log_2(1/\lambda)}{m}\,.$$

Proof. With Lemma II.11, we can directly compute the upper bound for $\mathfrak{R}_m(\mathcal{H}_r)$ by taking expectation over $D \sim \mathbb{P}^m$.

$$\begin{aligned}\mathfrak{R}_m(\mathcal{H}_r) &= \mathbb{E}_{D\sim\mathbb{P}^m}\mathfrak{R}_D(\mathcal{H}_r)\\ &\le \sqrt{\frac{(\ln 16)N\log_2 1/\lambda}{m}}\left(3\sqrt{2}\mathbb{E}\sup_{h\in\mathcal{H}_r}\|h\|_{L_2(D)} + 18\sqrt{r}\right)\,.\end{aligned}$$

By Jensen's inequality and A.8.5 in [34], we have

$$\begin{aligned}\mathbb{E}\sup_{h\in\mathcal{H}_r}\|h\|_{L_2(D)} &\le \left(\mathbb{E}\sup_{h\in\mathcal{H}_r}\|h\|^2_{L_2(D)}\right)^{1/2}\\ &\le \left(\mathbb{E}\sup_{h\in\mathcal{H}_r}\frac{1}{m}\sum_{i=1}^m h^2(x_i,y_i)\right)^{1/2}\\ &\le \left(\sigma^2 + 8\mathfrak{R}_m(\mathcal{H}_r)\right)^{1/2}\,,\end{aligned}$$

where $\sigma^2 := \mathbb{E}h^2$. When $\sigma^2 > \mathfrak{R}_m(\mathcal{H}_r)$, we have

$$\begin{aligned}\mathfrak{R}_m(\mathcal{H}_r) &\le \sqrt{\frac{(\ln 16)N\log_2(1/\lambda)}{m}}\left(9\sqrt{2}\sigma + 18\sqrt{r}\right)\\ &\le \sqrt{\frac{(\ln 16)N\log_2(1/\lambda)}{m}}\left(9\sqrt{2}\sqrt{Vr} + 18\sqrt{r}\right)\\ &\le 36\sqrt{\frac{2(\ln 16)N(V+1)\log_2(1/\lambda)}{m}}\sqrt{r}\,.\end{aligned}$$

The second inequality is because $\mathbb{E}h^2 \le V\mathbb{E}h$ and $\mathbb{E}h \le r$ for $h \in \mathcal{H}_r$.

When $\sigma^2 \le \mathfrak{R}_m(\mathcal{H}_r)$, we have

$$\begin{aligned}\mathfrak{R}_m(\mathcal{H}_r) &\le \sqrt{\frac{(\ln 16)N\log_2(1/\lambda)}{m}}\left(9\sqrt{2}\sqrt{\mathfrak{R}_m(\mathcal{H}_r)} + 18\sqrt{r}\right)\\ &\le 36\sqrt{\frac{(\ln 16)N\log_2(1/\lambda)}{m}}\sqrt{r} + 36^2\frac{(\ln 16)N\log_2(1/\lambda)}{m}\,.\end{aligned}$$

The last inequality can be obtained by dividing the formula into two cases, either $\mathfrak{R}_m(\mathcal{H}_r) < r$ or $\mathfrak{R}_m(\mathcal{H}_r) \ge r$ and then take the sum of the upper bounds of two cases.

Combining all these inequalities, we finally obtain an upper bound

$$\mathfrak{R}_m(\mathcal{H}_r) \leq C_1 \sqrt{\frac{(V+1)N\log_2(1/\lambda)}{m}}\sqrt{r} + C_2 \frac{N\log_2(1/\lambda)}{m}\,,$$

where C_1 and C_2 are two absolute constants. □

The last lemma gives the explicit formula of $\varphi_m(r)$, which will be finally plugged into Theorem II.7.

Lemma II.13. *When*

$$r = (900C_1^2 + 120C_2)N(V+1)\frac{\ln(1/\lambda)}{m} + (5B_0 + 72V)\frac{\ln(1/\delta)}{m} \tag{2.11}$$

we have

$$r \geq \max\{30\varphi_m(r), \frac{72V\ln(1/\delta)}{m}, \frac{5B_0\ln(1/\delta)}{m}\}.$$

It can be checked by simply plugging r into $\varphi_m(r)$.

Now with all these preparation, we can complete our proof of Theorem II.6

Proof. Under the assumption of Theorem II.6, $B_0 = 2\sqrt{N}R+1$ in Theorem II.7. We also have that $r^* \leq R^h_{\mathbb{P},\lambda}(f_0) - R^*$. By Lemma II.13, we can set

$$\begin{aligned} r = {} & (900C_1^2 + 120C_2)N(V+1)\frac{\ln(1/\lambda)}{m} \\ & + (10\sqrt{N}R + 5 + 72V)\frac{\ln(1/\delta)}{m} + R^h_{\mathbb{P},\lambda}(f_0) - R^*\,. \end{aligned}$$

By Theorem II.2, we have

$$\begin{aligned} R^h_{\mathbb{P},\lambda}(f_0) - R^* &= R^h_{\mathbb{P},\lambda}(f_0) - R^h_{\mathbb{P},\lambda}(g) + R^h_{\mathbb{P},\lambda}(g) - R^* \\ &\leq 2\sqrt{\mu}R + 4R^2\frac{\lambda}{2} + \epsilon\,, \end{aligned}$$

with probability $1-\delta$ when $N \geq 5d(\mu)\log\left(\frac{16d(\mu)}{\delta}\right)$.

When the spectrum of Σ decays polynomially,

$$d(\mu) \leq \frac{2c_2}{c_2 - 1}\left(\frac{c_1}{\mu}\right)^{1/c_2}\,.$$

Assume $m > c_1^{-(2+c_2)/(2c_2)}$. By choosing $\mu = c_1 m^{-\frac{2c_2}{2+c_2}} < c_1$ and $\lambda = m^{-c_2/(2+c_2)}$, we have

$$N = 10c_{1,2}m^{\frac{2}{2+c_2}}(\ln(32c_{1,2}m^{\frac{2}{2+c_2}}) + \ln(1/\delta))\,,$$

and

$$\begin{aligned}
&R^h_{\mathbb{P},\lambda}(f_{m,N,\lambda}) - R^* \\
&\quad \le \frac{18R}{m^{\frac{c_2}{2+c_2}}} + \frac{18R^2}{m^{\frac{c_2}{2+c_2}}} \\
&\quad + 30C_{1,2}c_{1,2}(\ln 32c_{1,2} + \frac{2}{2+c_2}\ln m + \ln(1/\delta))(V+1)\frac{c_2}{2+c_2}\frac{\ln m}{m^{\frac{c_2}{2+c_2}}} \\
&\quad + \frac{15(10c_{1,2}(\ln(32c_{1,2}m^{\frac{2}{2+c_2}}) + \ln(1/\delta)))^{1/2}Rm^{\frac{1}{2+c_2}} + 5 + 216V}{m}\ln(1/\delta)\,,
\end{aligned}$$

with probability $1 - 4\delta$, where

$$C_{1,2} = 900C_1^2 + 120C_2, \quad c_{1,2} = \frac{c_2 c_1^{1/c_2}}{c_2 - 1}\,.$$

When the spectrum of Σ decays sub-exponentially,

$$d(\mu) \le 5c_4^{-d}\ln^d(c_3/\mu)\,.$$

Assume that $m > \exp(-(c_4 \vee \frac{1}{c_4})d^2/2)$. By choosing $\mu = c_3/m^2$ and $\lambda = 1/m$, we have

$$N = 25c_{d,4}\ln^d(m)(\ln(80c_{d,4}\ln^d(m)) + \ln(1/\delta))\,,$$

and

$$\begin{aligned}
&R^h_{\mathbb{P},\lambda}(f_{m,N,\lambda}) - R^* \\
&\quad \le \frac{18R\sqrt{c_3}}{m} + \frac{18R^2}{m} \\
&\quad + 150C_{1,2}c_{d,4}(\ln 160c_{d,4} + d\ln\ln m + \ln(1/\delta))(V+1)\frac{\ln^{d+1} m}{m} \\
&\quad + \frac{150(c_{d,4}\ln^d(m)(\ln(80c_{d,4}\ln^d(m)) + \ln(1/\delta)))^{1/2}R + 5 + 216V}{m}\ln(1/\delta)\,,
\end{aligned}$$

with probability $1 - 4\delta$, where

$$C_{1,2} = 900C_1^2 + 120C_2, \quad c_{d,4} = \left(\frac{2}{c_4}\right)^d.$$

□

2.4 Learning Rate in Unrealizable Cases

To obtain the learning rate of RFSVM in unrealizable case, we need to find the approximator g in Theorem II.6 and give the estimate on $R_{\mathbb{P}}^h(g) - R^*$ and $\|g\|_\phi$. The construction of the approximator to the Bayes classifier relies on specific properties of the RKHS induced by the random feature. We focus on the RKHS corresponding to the Gaussian kernel in this section. Chapter III will discuss the construction for the RKHS induced by the random ReLU features. The general idea of the construction is to design an operator that maps the Bayes classifier to the RKHS. The approximation property of RKHS of Gaussian kernel has been studied in [34], where the margin noise exponent is defined to derive the risk gap. Here we introduce the simpler and stronger separation condition, which leads to a stronger result.

The points in $\mathcal{X}$ can be collected in to two sets according to their labels as follows,

$$\mathcal{X}_1 := \{x \in \mathcal{X} \mid \mathbb{E}(y \mid x) > 0\}$$
$$\mathcal{X}_{-1} := \{x \in \mathcal{X} \mid \mathbb{E}(y \mid x) < 0\}.$$

The distance of a point $x \in \mathcal{X}_i$ to the set $\mathcal{X}_{-i}$ is denoted by $\Delta(x)$.

Assumption II.14. *We say that the data distribution satisfies a separation condition if there exists $\tau > 0$ such that $\mathbb{P}_{\mathcal{X}}(\Delta(x) < \tau) = 0$.*

Intuitively, Assumption II.14 requires the two classes to be far apart from each other almost surely. This separation assumption is an extreme case when the margin noise exponent goes to infinity.

The separation condition characterizes a different aspect of the data distribution from Massart's low noise condition. Massart's low noise condition guarantees that the labels of data represent the distribution behind them accurately, while the separation condition guarantees the existence of a smooth, in the sense of small derivatives, function achieving the same risk with the Bayes classifier.

With both assumptions imposed on $\mathbb{P}$, we can get a fast learning rate of $\ln^{2d+1} m/m$ with only $\ln^{2d}(m)$ random features, as stated in the following theorem.

Theorem II.15. *Assume that $\mathcal{X}$ is bounded by radius ρ, and that the data distribution has density function upper bounded by a constant B and satisfies Assumption II.1 and II.14. Then by choosing*

$$\lambda = 1/m \quad \gamma = \tau/\sqrt{\ln m} \quad N = C_{\tau,d,\rho} \ln^{2d} m(\ln\ln m + \ln(1/\delta))\,,$$

the RFSVM using an optimized feature map corresponding to the Gaussian kernel with bandwidth γ achieves the learning rate

$$R^{0-1}_{\mathbb{P}}(f_{N,m,\lambda}) - R^{0-1}_{\mathbb{P}}(f^*_{\mathbb{P}}) \leq C_{\tau,V,d,\rho,B} \frac{\ln^{2d+1}(m)(\ln\ln(m) + \ln(1/\delta))}{m}\,,$$

with probability greater than $1 - 4\delta$ for $m \geq m_0$, where m_0 depends on τ, ρ, d.

This theorem only assumes that the data distribution satisfies low noise and separation conditions, and shows that with an optimized feature distribution, the learning rate of $\tilde{O}(1/m)$ can be achieved using only $\ln^{2d+1}(m) \ll m$ features. This justifies the benefit of using RFSVM in binary classification problems. The assumption of a bounded data set and a bounded distribution density function can be dropped if we assume that the probability density function is upper bounded by $C\exp(-c\|x\|^2/2)$, which suffices to provide the sub-exponential decay of spectrum of Σ_ϕ. But we prefer the simpler form of the results under current conditions. We speculate that the conclusion of Theorem II.15 can be generalized to all sub-Gaussian data.

Theorem II.15 requires a further analysis of the approximation error of RKHS to the Bayes classifier. This part adopts [34]'s idea of margin noise exponent. We say that the data distribution $\mathbb{P}$ has margin noise exponent $\beta > 0$ if there exists a positive constant c such that

$$\int_{\{x:\Delta(x)<t\}} |y| \mathrm{d}\mathbb{P}(x,y) \leq ct^{-\beta} \quad \forall t \in (0,1)\,.$$

Therefore, infinite β corresponds to Assumption II.14 with $\tau = 1$. However, the original proof of the approximation error that works with the margin noise exponent cannot be generalized to the case of infinite β, because the coefficient $\Gamma(d+\beta)/2^d$ will blow up (see Theorem 8.18 in [34]). This issue can be resolved by modifying the original proof, as shown below.

Lemma II.16. *Assume that there exists $\tau > 0$ such that*

$$\int_{\{x:\Delta(x)<t\}} |2\eta(x) - 1| \ \mathrm{d}\mathbb{P}_{\mathcal{X}}(x) = 0\,, \forall t < \tau\,,$$

where $\mathcal{X} \subset B^d(\rho)$ and $\eta(x)$ is a version of $\mathbb{P}(y=1|x)$. Then there exists a function f in the RKHS of the kernel

$$k_\gamma(x,x') = \exp\left(-\frac{\|x-x'\|^2}{2\gamma^2}\right)$$

where $\gamma < \tau/\sqrt{d-1}$ such that

$$R^h(f) - R^* < \frac{4\tau^{d-2}}{\Gamma(d/2)} \exp\left(-\frac{\tau^2}{\gamma^2}\right)\gamma^{d-2}\,,$$
$$\|f\|_{\mathcal{F}} \leq \frac{(\sqrt{\pi/2}\rho^2)^{d/2}}{\Gamma(d/2+1)}\gamma^{-d/2}$$

and

$$|f(x)| \leq 1\,.$$

Proof. First we define

$$\mathcal{X}_y := \{x : (2\eta(x)-1)y > 0\} \text{ for } y = \pm 1\,,$$

and $g(x) := (\sqrt{2\pi}\gamma)^{-d/2}\text{sign}(2\eta(x)-1)$. It is square integrable since $\eta(x) = 1/2$ for all $x \notin \mathcal{X}$. Then we map g onto the RKHS by the integral operator determined by k_γ,

$$f(x) := \int_{\mathbb{R}^d} \phi_\gamma(t;x)g(t)\ \mathrm{d}t\,,$$

where

$$\phi_\gamma(t;x) = \left(\frac{2}{\pi\gamma^2}\right)^{d/4} \exp\left(-\frac{\|x-t\|^2}{\gamma^2}\right)\,.$$

Note that it is a special property of Gaussian kernel that the feature map onto $L^2(\mathbb{R}^d)$ also has a Gaussian form. For other type of kernels, we may not have such a convenient characterization.

We know that

$$\|f\|_{\mathcal{H}} = \|g\|_{L^2} \le \frac{\sqrt{\text{Vol}(B^d(\rho))}}{(\sqrt{2\pi}\gamma)^{d/2}} = \frac{(\sqrt{\pi/2}\rho^2)^{d/2}}{\Gamma(d/2+1)}\gamma^{-d/2}\,.$$

Moreoever,

$$\begin{aligned}
|f(x)| &\le \int_{\mathbb{R}^d} \phi_\gamma(t;x)(\sqrt{2\pi}\gamma)^{-d/2}\ \mathrm{d}t \\
&= (\pi\gamma^2)^{-d/2}\int_{\mathbb{R}^d} \exp\left(-\frac{\|x-t\|^2}{\gamma^2}\right)\ \mathrm{d}t \\
&= 1\,.
\end{aligned}$$

Since f is uniformly bounded by 1, by Zhang's inequality [34], we have

$$R^h(f) - R^* = \mathbb{E}_{\mathbb{P}_\mathcal{X}}(|f(x) - \text{sign}(2\eta(x)-1)||2\eta(x)-1|)\,.$$

Now we give an upper bound on $|f(x) - \text{sign}(2\eta(x)-1)|$. Assume $x \in \mathcal{X}_1$. Then we

know that $f(x) \leq \text{sign}(2\eta(x) - 1) = 1$,

$$\begin{aligned}
1 - f(x) &= 1 - \left(\frac{1}{\pi\gamma^2}\right)^{d/2} \int_{\mathbb{R}^d} \exp\left(-\frac{\|x-t\|^2}{\gamma^2}\right) \text{sign}(2\eta(t) - 1)\ \mathrm{d}t \\
&= 1 - \left(\frac{1}{\pi\gamma^2}\right)^{d/2} \int_{\mathcal{X}_1} \exp\left(-\frac{\|x-t\|^2}{\gamma^2}\right)\ \mathrm{d}t \\
&+ \left(\frac{1}{\pi\gamma^2}\right)^{d/2} \int_{\mathcal{X}_{-1}} \exp\left(-\frac{\|x-t\|^2}{\gamma^2}\right)\ \mathrm{d}t \\
&\leq 2 - 2\left(\frac{1}{\pi\gamma^2}\right)^{d/2} \int_{B(x,\Delta(x))} \exp\left(-\frac{\|x-t\|^2}{\gamma^2}\right)\ \mathrm{d}t \\
&\leq 2 - 2\left(\frac{1}{\pi\gamma^2}\right)^{d/2} \int_{B(0,\Delta(x))} \exp\left(-\frac{\|t\|^2}{\gamma^2}\right)\ \mathrm{d}t \\
&= 2\left(\frac{1}{\pi\gamma^2}\right)^{d/2} \int_{\mathbb{R}^d \setminus B(0,\Delta(x))} \exp\left(-\frac{\|t\|^2}{\gamma^2}\right)\ \mathrm{d}t \\
&= \frac{4}{\Gamma(d/2)\gamma^d} \int_{\Delta(x)}^{\infty} \exp\left(-\frac{r^2}{\gamma^2}\right) r^{d-1}\ \mathrm{d}r\,.
\end{aligned}$$

Here the key is that $B(x, \Delta(x)) \subset \mathcal{X}_1$ when $x \in \mathcal{X}_1$. For $x \in \mathcal{X}_{-1}$, we have the same upper bound for $1 + f(x)$. Therefore, we have

$$\begin{aligned}
R^h(f) - R^* &\leq \frac{4}{\Gamma(d/2)\gamma^d} \int_{\mathcal{X}} \int_0^{\infty} \mathbf{1}_{(\Delta(x),\infty)}(r) \exp\left(-\frac{r^2}{\gamma^2}\right) r^{d-1} |2\eta(x) - 1|\ \mathrm{d}r\mathrm{d}\mathbb{P}_{\mathcal{X}}(x) \\
&= \frac{4}{\Gamma(d/2)\gamma^d} \int_0^{\infty} \int_{\mathcal{X}} \mathbf{1}_{(0,r)}(\Delta(x)) \exp\left(-\frac{r^2}{\gamma^2}\right) r^{d-1} |2\eta(x) - 1|\ \mathrm{d}\mathbb{P}_{\mathcal{X}}(x)\mathrm{d}r \\
&\leq \frac{4}{\Gamma(d/2)\gamma^d} \int_{\tau}^{\infty} \exp\left(-\frac{r^2}{\gamma^2}\right) r^{d-1}\ \mathrm{d}r
\end{aligned}$$

To get the last line, we apply the assumption on the expected label clarity. Now we only need to give an estimate of the integral.

$$\int_{\tau}^{\infty} \exp\left(-\frac{r^2}{\gamma^2}\right) r^{d-1}\ \mathrm{d}r \leq \int_{\tau}^{\infty} C \exp\left(-\alpha\frac{r^2}{\gamma^2}\right)\ \mathrm{d}r$$

where

$$C = \tau^{d-1} \exp(-(d-1)/2) \quad \alpha = 1 - 2\gamma^2\tau^{-2}(d-1)\,.$$

It is required that $\gamma < \sqrt{2}\tau/\sqrt{d-1}$ so that $\alpha > 0$. And then we can give an upper bound to the excess risk

$$R^h(f) - R^* \leq \frac{4\tau^d}{\Gamma(d/2)(2\tau^2 - (d-1)\gamma^2)} \exp\left(-\frac{\tau^2}{\gamma^2}\right) \gamma^{d-2}.$$

If we further require that $\gamma < \tau/\sqrt{d-1}$, then we have a simpler upper bound,

$$\frac{4\tau^{d-2}}{\Gamma(d/2)} \exp\left(-\frac{\tau^2}{\gamma^2}\right) \gamma^{d-2}.$$

□

Some remarks on this result:

1. The proof follows almost step by step the proof of [34]. The only difference occurs at where we apply Assumption II.14.
2. The approximation error is basically dominated by $\exp(-c/\gamma^2)$, and thus leaves us large room for balancing with the norm of the approximator.
3. The proof relies on the construction of the integral operator that maps $L^2(\mathbb{R}^d)$ function to the RKHS of the Gaussian kernel. A similar conclusion may hold for General RBF kernels using the fact that any RBF kernel can be expressed as an average of Gaussian kernel over different values of γ. A relevant reference is [31]. In Chapter III, we will see a similar construction for the RKHS induced by random ReLU features.

The last component for the proof of Theorem II.15 is the sub-exponential decay rate of the spectrum of Σ determined by the Gaussian kernel. The distribution of the spectrum of the convolution operator with respect to a distribution density function p has been studied by [38]. It shows that the number of eigenvalues of Σ greater than μ is asymptotic to $(2\pi)^{-d}$ times the volume of

$$\left\{(x, \xi) : p(x)\hat{k}(\xi) > \mu\right\},$$

where $\hat{k}$ is the Fourier transform of the kernel function k. By applying [38]'s work in our case, we have the following lemma. It is essentially Corollary 27 in [13], but our version explicitly shows the dependence on the band width β.

Lemma II.17. *Assume $\hat{k}(\xi) \leq \alpha \exp(-\beta\|\xi\|^2)$. If the density function $p(x)$ of probability distribution $\mathbb{P}_{\mathcal{X}}$ is bounded by B and $\mathcal{X}$ is a bounded subset of $\mathbb{R}^d$ with radius ρ, then*

$$\lambda_i(\Sigma) \leq C\alpha B \exp\left(-\beta\left(\frac{4\Gamma^{4/d}(d/2+1)}{\pi^{4/d}\rho^2}\right) i^{2/d}\right),$$

where $\lambda_1 \geq \lambda_2 \geq \cdots$ are eigenvalues of Σ in descending order.

Proof. Denote by E_t the set

$$\left\{(x,\xi) : \hat{k}(\xi)p(x) > t\right\}.$$

The volume, that is, the Lebesgue measure of E_t is denoted by $\mathrm{Vol}(E_t)$. By Theorem II of [38], the non-increasing function $\phi(\alpha)$ defined on $\mathbb{R}^+$ which is equi-measurable with $p(x)\hat{k}(\xi)$ describes the behaviour of λ_is. Indeed, $\lambda_i \leq C\phi((2\pi)^d i)$. By the volume formula of $2d$-dimensional ball we have the following estimate,

$$\begin{aligned}\sup\{s \in \mathbb{R}^+ : \phi(s) > t\} &= \mathrm{Vol}(E_t) \\ &\leq C_{d,\rho}\left(\frac{\ln(\alpha B/t)}{\beta}\right)^{d/2},\end{aligned}$$

where

$$C_{d,\rho} = \frac{\rho^d \pi^{d+2}}{\Gamma^2(d/2+1)}.$$

Solving for t, we know that

$$\phi(s) \leq \alpha B \exp\left(-\beta\left(\frac{s}{A}\right)^{2/d}\right).$$

Therefore, we have

$$\begin{aligned}\lambda_i(\Sigma) &\le C\alpha B \exp\left(-\beta\left(\frac{(2\pi)^d i}{A}\right)^{2/d}\right)\\ &= C\alpha B \exp\left(-\beta\left(\frac{4\Gamma^{4/d}(d/2+1)}{\pi^{4/d}\rho^2}\right) i^{2/d}\right).\end{aligned}$$

□

Now we can prove Theorem II.15.

Proof. Note that, by Lemma II.16, we can construct $g \in \mathcal{F}$ such that $R^h_{\mathbb{P},\lambda}(g) - R^*$ is controlled. And by Theorem II.2, we can find an $f_0 \in \mathcal{F}_N$ with similar risk to g. And this will be our f_0 as required by Theorem II.7. So we have

$$\begin{aligned}R^h_{\mathbb{P},\lambda}(f_0) - R^* \le{}& \frac{2(\sqrt{\pi/2}\rho^2)^d}{\Gamma^2(d/2+1)}\frac{\lambda}{\gamma^d} + \frac{2(\sqrt{\pi/2}\rho^2)^{d/2}}{\Gamma(d/2+1)}\gamma^{-d/2}\sqrt{\mu}\\ &+ \frac{4\tau^{d-2}}{\Gamma(d/2)}\exp\left(-\frac{\tau^2}{\gamma^2}\right)\gamma^{d-2},\end{aligned}$$

and with probability $1-\delta$, when $N = 5d(\mu)\ln(16d(\mu)/\delta)$. And

$$\|f_0\|_\infty \le C\sqrt{N}\gamma^{-d/2},$$

We choose $\gamma = \tau/\sqrt{\ln m}$ and $\lambda = 1/m$. Under the boundedness assumption on the density function and the property of Gaussian kernel, we know that by Lemma II.17,

$$\lambda_i(\Sigma) \le C\gamma B \exp\left(-\gamma^2\frac{4\Gamma^{4/d}(d/2+1)}{\pi^{4/d}\rho^2} i^{2/d}\right).$$

And similar to the second part of Theorem II.6, by identifying

$$c_3 = C\gamma B = CB\tau/\sqrt{\ln m} \quad c_4 = \frac{4\tau^2\Gamma^{4/d}(d/2+1)}{\pi^{4/d}\rho^2\ln m} := \frac{A}{\ln m},$$

and choosing $\mu = c_3/(m^{2d^2} \vee \exp(\frac{d^2}{c_4} \vee c_4 d^2))$, we have

$$d(\mu) \le 5d^{2d}(c_4^{-2d} \vee 1 \vee c_4^{-d}2^d \ln^d m).$$

Then when $m \geq \exp(A)$, we have $d(\mu) \leq 5(A^2 \wedge A/2)^{-d} \ln^{2d} m$, and

$$\begin{aligned} N &= 5d(\mu)(\ln(16d(\mu)) + \ln(1/\delta)) \\ &\leq 25(A^2 \wedge A/2)^{-d} \ln^{2d} m(\ln(80(A^2 \wedge A/2)^{-d}) + 2d \ln\ln m + \ln(1/\delta)) \,. \end{aligned}$$

Plug N and λ into Equation 2.11.

$$\begin{aligned} 3r = {} & 75C_{1,2}c_{d,\tau,\rho}(\ln(80c_{d,\tau,\rho}) + 2d \ln\ln m + \ln(1/\delta))(V+1)\frac{\ln^{2d+1} m}{m} \\ & + \left(150c_{d,\tau,\rho}^{1/2} \ln^{5d/4} m(\ln(80c_{d,\tau,\rho}) + 2d \ln\ln m + \ln(1/\delta))^{1/2} + 216V\right)\frac{\ln(1/\delta)}{m} \\ & + 3r^* \,, \end{aligned}$$

where

$$C_{1,2} = 900C_1^2 + 120C_2, \quad c_d = (A^2 \wedge A/2)^{-d} \,.$$

We can bound r^* by $R^h_{\mathbb{P},\lambda}(f_0) - R^*$. Therefore, the overall upper bound on the excess error is

$$\begin{aligned} & R^1_{\mathbb{P},\lambda}(f_{m,N,\lambda}) - R^* \\ & \quad \leq \frac{18(\sqrt{\pi/2}\rho^2)^d}{\Gamma^2(d/2+1)}\frac{\ln^{d/2} m}{\tau^d m} \\ & \quad + \frac{18(\sqrt{\pi/2}\rho^2)^{d/2}}{\Gamma(d/2+1)}\frac{\sqrt{CB\tau}\ln^{(d+1)/4} m}{\tau^{d/2} m^{d^2}} + \frac{36\tau^{d-2}}{\Gamma(d/2)}\frac{\tau^{d-2}}{m \ln^{d/2-1} m} \\ & \quad + 75C_{1,2}c_{d,\tau,\rho}\left(\ln(80c_{d,\tau,\rho}) + 2d \ln\ln m + \ln(1/\delta)\right)(V+1)\frac{\ln^{2d+1} m}{m} \\ & \quad + \left(150c_{d,\tau,\rho}^{1/2} \ln^{5d/4} m(\ln(80c_{d,\tau,\rho}) + 2d \ln\ln m + \ln(1/\delta))^{1/2} + 216V\right)\frac{\ln(1/\delta)}{m} \,. \end{aligned}$$

□

The analysis in both cases requires access to optimized features. If we drop the assumption of optimized feature map, only weak results can be obtained for the learning rate and the number of features required.

As shown by [30], RFKRR can achieve excess risk of $O(1/\sqrt{m})$ using $O(\sqrt{m}\log(m))$ features without the optimized feature assumption. However, it is inappropriate to directly compare this result with the learning rate in classification scenario. Because as surrogate loss functions, least square loss has a different calibration function with hinge loss. Basically, $O(\epsilon)$ risk under square loss only implies $O(\sqrt{\epsilon})$ risk under $0-1$ loss, while $O(\epsilon)$ risk under hinge loss implies $O(\epsilon)$ risk under $0-1$ loss. Therefore, [30]'s analysis only implies an excess risk of $O(m^{-1/4})$ in classification problems with $\tilde{O}(\sqrt{m})$ features.

For RFSVM, we expect a similar result. Without assuming an optimized feature map, the leverage score can only be upper bounded by κ^2/μ, where κ is the upper bound on the function $\phi(\omega; x)$ for all ω, x. Substituting κ^2/μ for $d(\mu)$ in the proofs of learning rates, we need to balance $\sqrt{\mu}$ with $1/(\mu m)$ to achieve the optimal rate. This balance is not affected by the spectrum of Σ or whether $f^*_{\mathbb{P}}$ belongs to $\mathcal{F}$. Obviously, setting $\mu = m^{-2/3}$, we get a learning rate of $m^{-1/3}$, with $\tilde{O}(m^{2/3})$ random features. Even though this result is also new for RFSVM in regularized formulation, the gap to previous analysis like [28] is too large. Considering that the random features used in practice that are not optimized also have quite good performance, we need further analysis on RFSVM without optimized feature map. In particular, we can only show that $1/\epsilon^2$ random features are sufficient to guarantee the learning rate less than ϵ when $1/\epsilon^3$ samples are available. Though not helpful for justifying the computational benefit of random features method, this result matches the parallel result for RFKRR in [30] and the approximation result in [33].

[30] also compared the performance of RFKRR with Nystrom method, which is the other popular method to scale kernel ridge regression to large data sets.We do not find any theoretical guarantees on the fast learning rate of SVM with Nystrom

method on classification problems in the literature, though there are several works on its approximation quality to the accurate model and its empirical performance (see [39, 40]). The tools used in this paper should also work for learning rate analysis of SVM using Nystrom method. We leave this analysis to the future.

2.5 Experimental Results

The main limit of our two theorems is the assumption of an optimized feature distribution, which is not clear how to obtain in practice yet. Developing a data-dependent feature selection method is therefore an important problem for future work on RFSVM. [3] proposed an algorithm to approximate the optimized feature map from any feature map. Adapted to our setup, the reweighted feature selection algorithm is described as follows.

Algorithm 1: Reweighted Feature Selection

input : L uniform subsamples of data $\{x_i\}_{i=1}^{L}$,

M feature vectors $\{\omega_i\}_{i=1}^{M}$,

regularization hyperparameter λ;

output: N optimized features;

Generate the matrix Φ with columns $\phi_M(x_i)/\sqrt{L}$;

Compute $\{r_i\}_{i=1}^{M}$, the diagonal of $\Phi\Phi^\intercal(\Phi\Phi^\intercal + \mu I)^{-1}$;

Resample N features from $\{\omega_i\}_{i=1}^{M}$ according to the probability distribution

$p_i = r_i / \sum r_i$;

The theoretical guarantees of this algorithm have not been discussed in the literature. A result in this direction will be extremely useful for guiding practioners. Instead, here we implement it in our experiment and empirically compare the performance of RFSVM using this reweighted feature selection method to the performance of RFSVM without this preprocessing step[2].

[2]The source code is available at `https://github.com/syitong/randfourier`.

The sample points shown in Figure 2.2 are generated from either the inner circle or outer annulus uniformly with equal probability, where the radius of the inner circle is 0.9, and the radius of the outer annulus ranges from 1.1 to 2. The points from the inner circle are labeled by -1 with probability 0.9, while the points from the outer annulus are labeled by 1 with probability 0.9. In such a simple case, the unit circle describes the boundary of the Bayes classifier.

First, we compared the performance of RFSVM with that of KSVM on the training set with 1000 samples, over a large range of regularization parameter ($-7 \leq \log \lambda \leq 1$). The bandwidth parameter γ is fixed to be an estimate of the average distance among the training samples. After training, models are tested on a large testing set ($> 10^5$). For RFSVM, we considered the effect of the number of features by setting N to be $1, 3, 5, 10$ and 20, respectively. Moreover, both feature selection methods, simple random feature selection (labeled by 'unif' in the figures), which does not apply any preprocess on drawing features, and reweighted feature selection (labeled by 'opt' in the figures) are inspected. For the reweighted method, we set $M = 100N$ and $L = 0.3m$ to compute the weight of each feature. Every RFSVM is run 10 times, and the average accuracy and standard deviation are presented.

The results of KSVM, RFSVMs with 1 and 20 features are shown in Figure 2.5. The performance of RFSVM is slightly worse than the KSVM, but improves as the number of features increases. It also performs better when the reweighted method is applied to generate features.

To further compare the performance of simple feature selection and reweighted feature selection methods, we plot the learning rate of RFSVM with $O(\ln^2(m))$ features and the best λs for each sample size m. KSVM is not included here since it is too slow on training sets of size larger than 10^4 in our experiment compared to

RFSVM. The error rate in Figure 2.3 is the excess risk between learned classifiers and the Bayes classifier. We can see that the excess risk decays as m increases, and the RFSVM using reweighted feature selection method outperforms the simple feature selection.

According to Theorem II.15, the benefit brought by the optimized random feature, that is, the fast learning rate, will show up when the sample size is greater than $O(\exp(d))$. The number of random features required also depends on d, the dimension of data. For data of small dimension and large sample size, as in our experiment, it is not a problem. However, in applications of image recognition, the dimension of the data is usually very large and it is hard for our theorem to explain the performance of RFSVM. On the other hand, if we do not pursue the fast learning rate, the analysis for general feature maps, not necessarily optimized, gives a learning rate of $O(m^{-1/3})$ with $O(m^{2/3})$ random features, which does not depend on the dimension of data. Actually, for high dimensional data, there is barely any improvement in the performance of RFSVM by using reweighted feature selection method; see Figure 2.5). It is important to understand the role of d to fully understand the power of random features method.

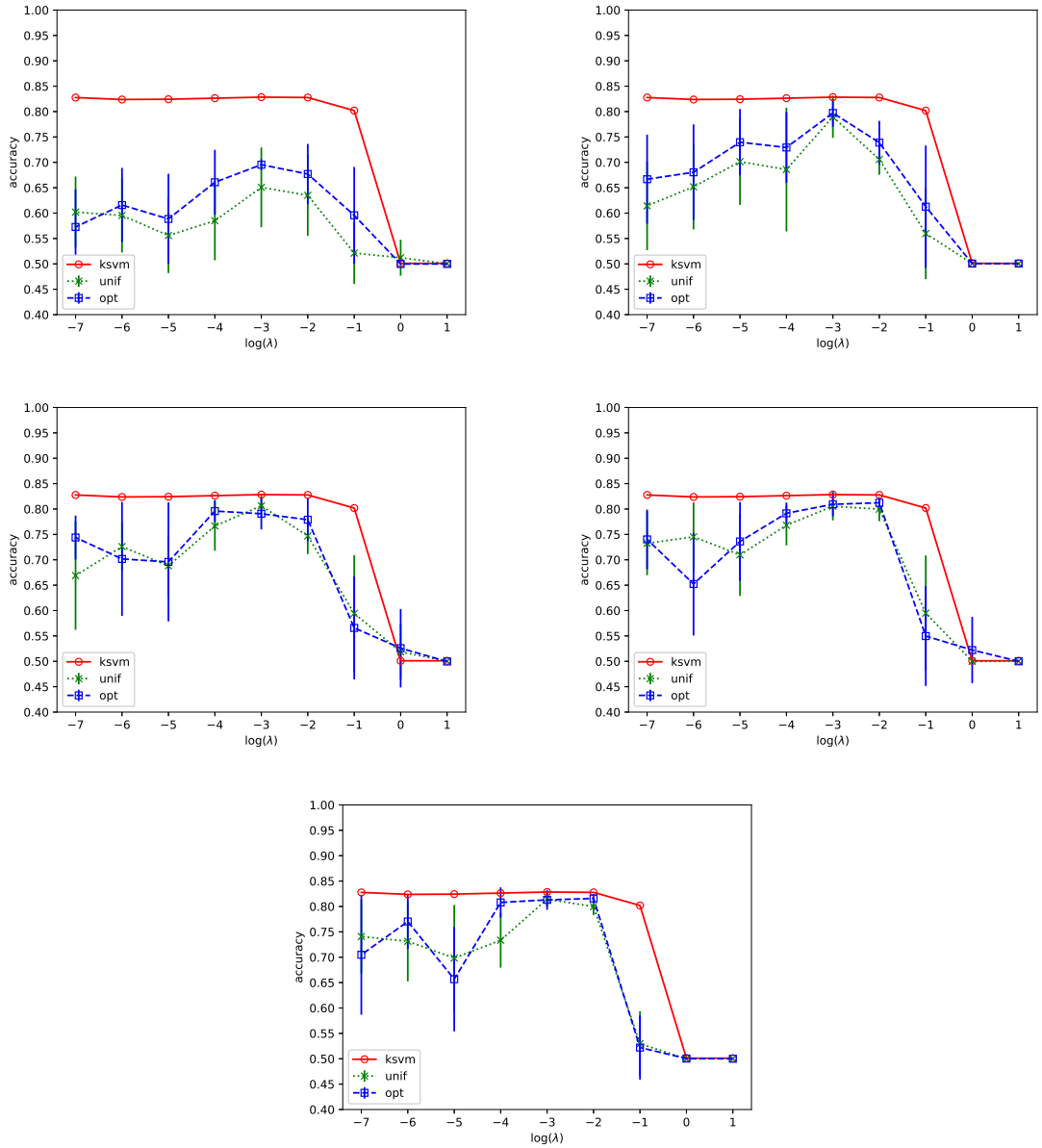


Figure 2.1: Comparison between RFSVMs with KSVM Using Gaussian Kernel.
Top left: $N = 1$; top right: $N = 3$; middle left: $N = 5$; middle right: $N = 10$; bottom: $N = 20$. "ksvm" is for KSVM with Gaussian kernel, "unif" is for RFSVM with direct feature sampling, and "opt" is for RFSVM with reweighted feature sampling. Error bars represent standard deviation over 10 runs. Each sub-figure shows the performance of RFSVM with different number of features.

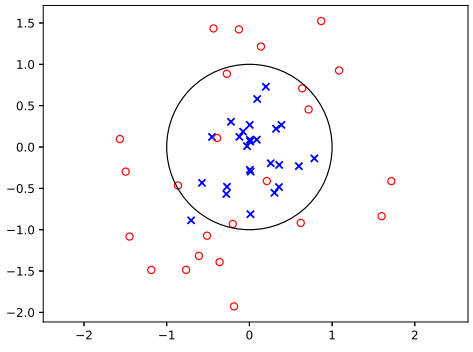


Figure 2.2: Distribution of Training Samples.

50 points are shown in the graph. Blue crosses represent the points labeled by -1, and red circles the points labeled by 1. The unit circle is one of the best classifier for these data with 90% accuracy.

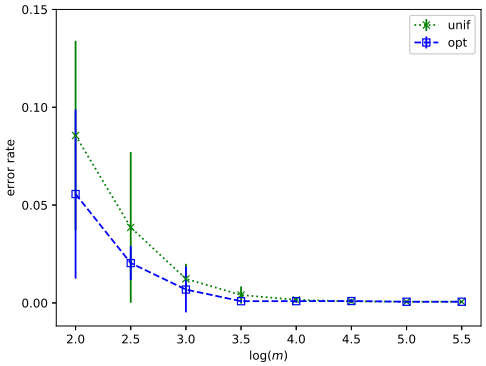
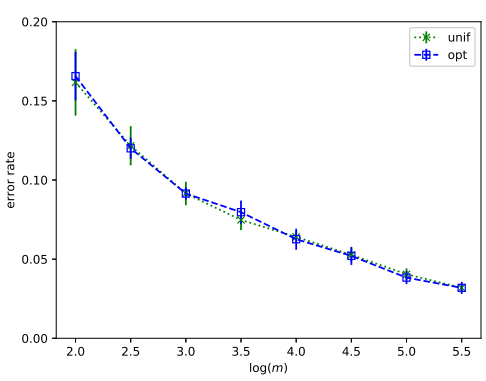


Figure 2.3: Learning Rate of RFSVMs.

The excess risks of RFSVMs with the simple random feature selection ("unif") and the reweighted feature selection ("opt") are shown for different sample sizes. The error rate is the excess risk. The error bars represent the standard deviation over 10 runs.



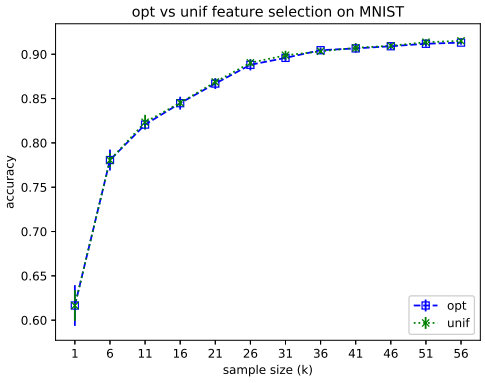


Figure 2.4: The performance of RFSVMs on 10 dimensional data and MNIST.
The simple random feature selection ("unif") and the reweighted feature selection ("opt") are shown for different size of sample data. Left: The data labeled with probability 0.9 to be -1 are within the 10 dimensional ball centered at the origin and radius 0.9, and the data labeled with probability 0.9 to be 1 are within the shell of radius 1.1 to 2. The error rate is the excess risk. The error bars represent the standard deviation over 10 runs. Right: The accuracy of RFSVM with two different feature selection methods on MNIST.

CHAPTER III

The Approximation Properties of Random ReLU Features

In the analysis of the learning rate of random Fourier features in unrealizable cases, the core is to understand the approximation property of the RKHS induced by random Fourier features. In this chapter, we will consider the RKHS induced by a class of random features inspired by neural networks. We are particularly interested in the random features with ReLU as feature maps.

The results in this chapter are summarized as follows.

1. We establish sufficient conditions for the universality of RKHSs derived from random features of neural network type (Theorem III.4). And based on this result we are able to prove the universality of random features methods in admissible cases (Corollary III.9 and III.8).
2. We describe the random ReLU features method (Algorithm 2) and prove its universal consistency (Proposition III.11). We compare the performance of random ReLU features to random Fourier features and confirm the advantages of random ReLU features on computational cost.
3. We prove that compositions of functions in the RKHS induced by the random ReLU feature can be efficiently approximated by multilayer ReLU networks (Proposition III.15). We also prove that the composition of functions in

the RKHS generates functions more complicated than functions in the RKHS (Proposition III.16). Beyond the proof of existence of the depth separation, we designed synthetic data and experiments showing that the good multi-layer approximator can be found by stochastic gradient descent while shallow models have poor performance.

In Section 2.1, we review some basic functional analysis and probability results that are useful for obtaining the approximation results, and define the notations used in this chapter. The universality of the random ReLU features is given in Section 3.3. We describe a simple random ReLU features method and prove an upper bound on its generalization error in Section 3.4, which implies the universal consistency of the algorithm. The approximation result by multilayer ReLU networks and the depth separation result of RKHSs induced by the random ReLU feature are presented in Section 3.5 and 3.6. The performance of random ReLU features in experiments and comparison with random Fourier features and neural networks are discussed in Section 3.7.

3.1 Universality and Random Features of Neural Network Type

Throughout the chapter, we assume that $\mathcal{X}$ is a closed subset contained in the d dimensional ball centered at the origin with radius r. Let $C(\mathcal{X})$ denote the space of all continuous functions on $\mathcal{X}$ equipped with the supremum norm. When a subset of $C(\mathcal{X})$ is dense, we call it universal. To show that a subset is dense in a Banach space, we need only consider its annihilator as described by the following lemma.

Lemma III.1. *For a Banach space $\mathcal{B}$ and its subset U, the linear span of U is dense in $\mathcal{B}$ if and only if $U^{\perp}$, the annihilator of U, is $\{0\}$.*

The proof can be easily derived from Theorem 8 in Chapter 8 of [20]. It is a

consequence of Hahn-Banach theorem. Since $\mathcal{X}$ is compact, the dual space of $C(\mathcal{X})$ is the space of all signed measures equipped with the total variation norm, denoted by $M(\mathcal{X})$ (see Theorem 14 in Chapter 8 of [20]). As the consequence of Lemma III.1 and the duality between $C(\mathcal{X})$ and $M(\mathcal{X})$, [25] uses the following useful criteria for justifying the dense subset of $C(\mathcal{X})$.

Lemma III.2. *$\mathcal{F} \subset C(\mathcal{X})$ is universal if and only if for any signed measure ν,*

$$\int_{\mathcal{X}} f(x)\, \mathrm{d}\nu(x) = 0 \quad \forall f \in \mathcal{F} \implies \nu = 0\,.$$

The definition of random features and the relation between random features and kernels have been introduced in Section 2.1. We only remind readers that the RKHS induced by the random feature (ϕ, μ) is denoted by $(\mathcal{H}_{\phi,\mu}, \|\cdot\|_{\phi,\mu})$. The feature map we consider in this work is of the form $\phi(x;\omega,b) = \sigma(\omega \cdot x + b)$ where σ is a non-linear function on $\mathbb{R}$. The random features models using this feature map, as we pointed out in Section I, are described by the same formula as that of two-layer neural networks using σ as activation node. We will see in Section 3.3 that the universality of these two models are also closely connected. To simplify our notations, we denote by $(x, 1)$ the concatenation of the d-dimensional vector x and a scalar 1. Since we do not treat the bias variable b different from the coefficients ω, we view all the inner weights as a $d+1$ dimensional vector and call it ω. Hence, the functions expressed by N random features appear as

$$f_N(x) = \sum_{i=1}^{N} c_i \sigma(\omega \cdot (x, 1))\,. \tag{3.1}$$

We are most interested in the case where $\sigma(z) = \max(0, z)$, which is the ReLU node.

Other notations include: $\mathbb{P}$ denotes the data distribution over $\mathcal{X}$; $|\cdot|$ means the Euclidean norm when the operand is a vector in $\mathbb{R}^d$, but the total variation norm

when the operand is a measure; $f_{i:j}$ represents the composition of functions $f_j \circ \cdots \circ f_i$; $\mathbf{R}^\ell$ denotes the expected risk with respect to the loss ℓ; and τ_d denotes the uniform probability distribution over $\mathbb{S}^d$.

3.2 Barron's Class and Maurey's Sparsification Lemma

The seminal paper [5] considered the function class described by the following formula,

$$f(x) = f_0 + \int \left(e^{i\omega \cdot x} - 1\right) \, \mathrm{d}\rho(\omega) \,,$$

where $f_0 \in \mathbb{R}$ and ρ is a complex-valued measure on $\mathbb{R}^d$. This class of functions is called Barron's class, denoted by $\Gamma(\mathcal{X})$. Let $|\omega|_{\mathcal{X}} = \sup_{x \in \mathcal{X}} |\omega \cdot x|$. The subset of Γ in which ρ satisfies

$$\int |\omega|_{\mathcal{X}} \, \mathrm{d}|\rho|(\omega) < R$$

for some constant R is denoted by Γ_R. When σ in f_N is a sigmoidal function, Barron in his work gave upper bounds on the number N and the scale of the outer coefficients c_i to approximate functions in Γ_R. As for the inner weights ω_i's, Barron proved that they may increase to infinity as the approximation error goes to 0 for sigmoidal activation nodes except for the unit step function.

Barron's class provides a way to constrain the complexity of functions to enable quantitative approximation analysis. At the same time, the class remains larger than the simple hypothesis class Equation 3.1 so that it includes more functions of interest. Barron's class can also be used as a building block to construct a larger function class by composition. [22] extended Barron's result to multi-layer neural networks and showed that they approximate compositions of Barron's functions.

The proof of Barron's result relies on the famous Maurey's sparsification lemma[27].

Lemma III.3. *Assume that $X_1, \ldots, X_n$ are n i.i.d. random variables with values in the unit ball of a Hilbert space. Then with probability greater than $1 - \delta$, we have*

$$\left\| \frac{1}{n} \sum_{i=1}^{n} X_i - \mathbb{E}X_1 \right\| \leq \frac{1}{\sqrt{n}} \left(1 + \sqrt{2 \log \frac{1}{\delta}} \right) . \tag{3.2}$$

Furthermore, there exists $x_i, \ldots, x_n$ in the unit ball such that

$$\left\| \frac{1}{n} \sum_{i=1}^{n} x_i - \mathbb{E}X_1 \right\| \leq \frac{1}{\sqrt{n}} . \tag{3.3}$$

We will use this lemma to prove a result for random ReLU features similar to Barron's result in Section 3.5.

3.3 Universality of Random Features

To study the approximation capability of random ReLU features models, we start with the universality of the RKHS induced by random features of neural network type.

Theorem III.4. *Assume that $|\sigma(z)| \leq K|z|^k + M$ for some $M, K \geq 0$ and $k \in \mathbb{N}_0$. Let μ be a probability distribution whose support is dense in $\mathbb{R}^{d+1}$ with $\int |\omega|^{2k} \, \mathrm{d}\mu(\omega) \leq M_2$. If σ is not a polynomial, the RKHS $\mathcal{H}_{\sigma,\mu}$ is universal.*

Proof. First, it is easy to see that $\sigma(\omega \cdot (x, 1)) \in L^2(\mathbb{R}^{d+1}, \mu)$. Indeed, since $|\sigma(z)| \leq K|z|^k + M$, we have

$$\begin{aligned} \int \sigma^2(\omega \cdot (x,1)) \, \mathrm{d}\mu(\omega) &\leq 2 \int K^2 (\omega \cdot (x,1))^{2k} \, \mathrm{d}\mu(\omega) + 2M^2 \\ &\leq 2K^2 (|x|^2 + 1)^k \int |\omega|^{2k} \, \mathrm{d}\mu(\omega) + 2M^2 \\ &\leq 2K^2 (r^2 + 1)^k M_2 + 2M^2 , \end{aligned}$$

where r is the radius of $\mathcal{X}$. Next, we show that the functions in $\mathcal{H}_{\sigma,\mu}$ are all continuous. For $f \in \mathcal{H}_{\sigma,\mu}$, assume that

$$f(x) = \int_{\Omega} \sigma(\omega \cdot (x,1)) g(\omega) \, \mathrm{d}\mu(\omega)$$

for some $g \in L^2(\mu)$. To show that f is continuous at a given point x, we want to show that for any $\epsilon > 0$, there exists $\delta > 0$ such that $|f(x) - f(y)| < \epsilon$ whenever $|x - y| < \delta$. Denote

$$I_1(R) \coloneqq \left| \int_{|\omega|>R} (\sigma(\omega \cdot (x,1)) - \sigma(\omega \cdot (y,1)))g(\omega) \, \mathrm{d}\mu(\omega) \right| ,$$

and

$$I_2(R) \coloneqq \left| \int_{|\omega|\leq R} (\sigma(\omega \cdot (x,1)) - \sigma(\omega \cdot (y,1)))g(\omega) \, \mathrm{d}\mu(\omega) \right| .$$

Then since

$$\begin{aligned} I_1(R) &\leq \int_{|\omega|>R} \left(K(|\omega \cdot (x,1)|^k + |\omega \cdot (y,1)|^k) + 2M\right) |g(\omega)| \, \mathrm{d}\mu(\omega) \\ &\leq 2\sqrt{2}\|g\|_{L^2} \left(\int_{|\omega|>R} K^2 |\omega|^{2k} (r^2+1)^k + M^2 \, \mathrm{d}\mu(\omega) \right) , \end{aligned}$$

we know that $I_1(R) \to 0$ as $R \to \infty$. In particular, for sufficiently large R, we have $I_1(R) < \epsilon/2$. On the other hand, since σ is continuous, there exists $\delta_1 > 0$ such that $|\omega \cdot (x,1) - \omega \cdot (y,1)| < \delta_1$ implies that

$$\begin{aligned} I_2(R) &\leq \int_{|\omega|\leq R} \frac{\epsilon}{2\|g\|_{L^2}} |g(\omega)| \, \mathrm{d}\mu(\omega) \\ &\leq \frac{\epsilon}{2} . \end{aligned}$$

So if we set $\delta = \delta_1/R$, we will have $|f(x) - f(y)| < \epsilon$.

We just proved that all the functions in the $\mathcal{H}_{\sigma,\mu}$ are all continuous. So we can use Lemma III.2 to justify the universality. For a signed measure ν with finite total variation, assume that

$$\int_{\mathcal{X}} \int_{\mathbb{R}^{d+1}} \sigma(\omega \cdot (x,1)) g(\omega) \, \mathrm{d}\mu(\omega) \mathrm{d}\nu(x) = 0 ,$$

for all $g \in L^2(\mathbb{R}^{d+1}, \mu)$. We want to show that ν must be the 0 measure. Since the function $\sigma(\omega \cdot (x,1))g(\omega)$ is integrable over $\mu \times \nu$, by Fubini's theorem we have

$$\int_{\mathbb{R}^{d+1}} \left(\int_{\mathcal{X}} \sigma(\omega \cdot (x,1)) \, \mathrm{d}\nu(x) \right) g(\omega) \, \mathrm{d}\mu(\omega) ,$$

equals 0 for all $g \in L^2(\mathbb{R}^{d+1}, \mu)$. Then

$$\int_{\mathbb{R}^{d+1}} \sigma(\omega \cdot (x, 1))\, \mathrm{d}\nu(x) = 0 \ \mu\text{-a.e.} \tag{3.4}$$

Indeed the function of ω defined on the left hand side of Equation 3.4 has to be 0 everywhere because of continuity. Since σ is not a polynomial, by [23], we know that ν must be a 0 measure. If it is not, then there exists f in $C(\mathcal{X})$ such that $\int f\, \mathrm{d}\nu = \epsilon$ where $\epsilon \geq 0$. Because the linear span of $\sigma(\omega \cdot (x, 1))$ is dense in $C(\mathcal{X})$, there must exist c_is and ω_is such that

$$\int \sum_{i=1}^{k} c_i \sigma(\omega_i \cdot (x, 1))\, \mathrm{d}\nu(x) \geq \frac{\epsilon}{2}\,.$$

This contradicts Equation 3.4. □

It is interesting to note that in the proof the universality of the RKHS induced by a random feature can be derived from that of the corresponding neural networks.

Theorem III.4 shows that the RKHSs induced by a broad class of random features of neural network type are dense in the space of continuous functions. And hence these random features methods have sufficient capacity for supervised learning tasks.

The general theorem requires the feature distribution to be supported almost everywhere on $\mathbb{R}^{d+1}$. However, when the feature map is ReLU, this requirement can be relaxed due to the homogeneity. The following proposition provides us a sufficient condition for the universality of $\mathcal{H}_{\mathrm{ReLU},\mu}$.

Proposition III.5. *Let μ be a probability distribution supported on a dense subset of an d-dimensional ellipsoid centered at the origin. Then the RKHS $\mathcal{H}_{\mathrm{ReLU},\mu}$ is universal.*

Proof. The proof is the same up to Eq. 3.4. We claim that

$$\int_{\mathcal{X}} \sigma(\omega \cdot (x, 1))\, \mathrm{d}\nu(x) = 0$$

for all $\omega \in \mathbb{R}^{d+1}$. If not, there exists $\beta \neq 0$ such that

$$\int_{\mathcal{X}} \sigma(\beta \cdot (x,1))\ \mathrm{d}\nu(x) = \epsilon > 0..$$

Then there exists a positive constant c such that $c\beta$ belongs to the support of μ because of the shape of the feature space. By homogeneity, we have

$$\int_{\mathcal{X}} \sigma\left(c\beta \cdot (x,1)\right)\ \mathrm{d}\nu(x) = c\epsilon > 0\,.$$

The function

$$g : \omega \mapsto \int_{\mathcal{X}} \sigma(\omega \cdot (x,1))\ \mathrm{d}\nu(x)$$

is continuous over the ellipsoidal feature space Ω, and thus there exists $\delta > 0$ such that $g(\omega)$ is greater than $\epsilon/2$ for all $\omega \in B^{d+1}(c\beta, \delta) \cap \Omega$. This contradicts the fact that the support of μ is dense over Ω. Then by the same argument at the end of the proof of Theorem III.4, we complete the proof of the proposition. □

This proposition is only stated for the regular feature space like $\mathbb{S}^d$ or ellipsoids around the origin for simplicity of the statement. The proof actually works for any set Ω satisfying the following property: for any $\omega \neq 0$ in $\in \mathbb{R}^{d+1}$, there exists a scalar $c > 0$ such that $c\omega \in \Omega$. This proposition allows us to restrict the ReLU feature map to be a bounded function when the data set is bounded. And this will be important for us when we prove the approximation capability of random ReLU features methods.

As we mentioned in the introduction, the universality of $\mathcal{H}_{\mathrm{ReLU},\tau_d}$, where τ_d is the uniform distribution over $\mathbb{S}^d$, has been shown implicitly in [3]. However, due to the non-constructive approach we take, it is much easier for us to prove the universality of the RKHS induced by any homogeneous random feature σ with μ supported over various domains described in the above paragraph.

In the next, we study the approximation properties of random ReLU features. As we mentioned in the introduction, since random features methods are not deterministic, the approximation property has to be stated with probability. Therefore we first give the following definition.

Definition III.6. Given a random feature (σ, μ), if for any f in $C(\mathcal{X})$ and $\delta, \epsilon > 0$, there exist a positive integer N such that with probability greater than $1 - \delta$, we can find coefficients $\{c_i\}_{i=1}^N$ such that

$$\|f_N - f\|_{L^2(\mathbb{P})} > \epsilon \,,$$

we say that the random feature is universal.

If we've known that the RKHS induced by the random feature (σ, μ) is universal, to show that it is universal in the sense of Definition III.6, we only need to verify that any function $f \in \mathcal{H}_{\sigma,\mu}$ can always be approximated by the linear combination of finitely many random features, f_N, with high probability of random choice of ω_i's. This has been studied in [4]. To apply Bach's result, we make the following definition.

Definition III.7. A random feature (σ, μ) is called admissible if for any $\lambda > 0$,

$$\sup_{\omega \in \Omega} \langle \sigma_\omega, (\Sigma + \lambda I)^{-1} \sigma_\omega \rangle < \infty \,,$$

where $\Sigma : L_2(\mathbb{P}) \to L_2(\mathbb{P})$ is defined by

$$\Sigma f = \int k_{\sigma,\mu}(x, y) f(y) \, \mathrm{d}\mathbb{P}(y) \,,$$

and $\sigma_\omega(x) = \sigma(\omega \cdot (x, 1))$.

This property is used in [35, 30] to efficiently obtain random features in their learning rate analysis. It is a data-dependent property except for special cases such as bounded feature maps. How to design the admissible random features remains an

open problem. Though it is data-dependent, it does not depend on the labels or target functions. And hence we may gain advantages in supervised learning tasks if we can design random features according to the characteristics of the data distribution.

With this admissibility assumption, we have the following complete description of universality of random features.

Corollary III.8. *Assume that* $|\sigma(z)| \leq K|z|^k + M$ *for some* $M, K \geq 0$ *and* $k \in \mathbb{N}_0$. *Let* μ *be a probability distribution whose support is dense in* $\mathbb{R}^{d+1}$ *with* $\int |\omega|^{2k} \, \mathrm{d}\mu(\omega) \leq M_2$. *If* σ *is not a polynomial and* (σ, μ) *is admissible with respect to the data distribution* $\mathbb{P}$, *the random feature* (σ, μ) *is universal.*

Proof. The condition guarantees that $\mathcal{H}_{\sigma,\mu}$ is dense in $C(\mathcal{X})$. For any f in $C(\mathcal{X})$ and $\epsilon > 0$, there exists $\tilde{f} \in \mathcal{H}_{\sigma,\mu}$ such that $\sup_x |f(x) - \tilde{f}(x)| < \epsilon$, which implies that $\|f - \tilde{f}\|_{L_2(\mathbb{P})} < \epsilon$. Assume that

$$\tilde{f}(x) = \int_\Omega \sigma(\omega \cdot (x, 1)) g(\omega) \, \mathrm{d}\mu(\omega) \,,$$

where $\|g\|_{L_2(\mu)} = G$. Since (σ, μ) is admissible, there exists $\lambda \leq \epsilon^2/(4G^2)$ such that

$$d_{\max}(1, \lambda) = \sup_{\omega \in \Omega} \langle \sigma_\omega, (\Sigma + \lambda I)^{-1} \sigma_\omega \rangle < \infty \,.$$

So by Theorem II.2, there exists $N \in \mathbb{N}$ such that

$$\|\tilde{f} - f_N\|_{L^2(\mathbb{P})} \leq 2G\sqrt{\lambda} \leq \epsilon \,,$$

with probability over the random $\omega_i \sim \mu$ greater than $1 - \delta$. □

It is easy to see that the random feature (σ, μ) is admissible when σ is bounded. Therefore, we have the following corollary.

Corollary III.9. *When* σ *is bounded and* $\mathcal{H}_{\sigma,\mu}$ *is universal. The random feature* (σ, μ) *is universal in the sense of Definition III.6.*

Proof. We need only check that (σ, μ) is admissible. Assume $|\sigma| \leq \kappa$. Then, for any $\lambda > 0$,

$$\sup_{\omega \in \Omega} \langle \sigma_\omega, (\Sigma + \lambda I)^{-1} \sigma_\omega \rangle \leq \lambda^{-1} \|\sigma_\omega\|^2_{L^2(\mathbb{P})} \leq \frac{\kappa^2}{\lambda}.$$

□

[15] proved that under the assumptions of Corollary III.9, for any continuous function f and randomly generated ω_i's, with probability 1, there exist c_i's such that f_N converges to f under $L^2(\mathbb{P})$-norm as N goes to infinity. Note that Definition III.6 only requires the convergence in probability instead of almost surely, and thus Corollary III.9 seems to be weaker than Huang's result. They are actually equivalent. To see this, first note that the statement is on the existence of the approximator in the linear space of n random bases. Denote by $E(\{\omega_i\}_{i=1}^n)$ the space spanned by $\{\sigma(\omega_i \cdot (x, 1))\}_{i=1}^n$. The convergence in probability stated in Corollary III.9 implies that there exists a subsequence $\{n_k\}_{k=1}^\infty$ such that with probability 1, the infinite sequence of $\{\omega_i\}_{i=1}^\infty$ sampled randomly satisfies that $E(\{\omega_i\}_{i=1}^{n_k})$ contains an approximator converging to the target function as k goes to infinity. Since $E(\{\omega_i\}_{i=1}^n) \subset E(\{\omega_i\}_{i=1}^m)$ whenever $n \leq m$, the almost sure convergence holds for all n.

The above corollary shows that random ReLU features with parameter distribution μ supported on the ellipsoid around the origin can approximate any continuous function with high probability over the random choice of parameter ω_is. We will give a detailed description of random ReLU features method in the supervised learning context and discuss its performance in Section 3.4 and Section 3.7.

3.4 Learning Rate of the Random ReLU Features Method

In this section, we describe a supervised learning algorithm using random ReLU features, and give a loose upper bound on the number of nodes and sample size required to achieve an excess risk of ϵ. The purpose is to show that random ReLU features method is a universally consistent supervised learning algorithm.

Algorithm 2: Random ReLU features method.

input : $\{(x_i, y_i)\}_{i=1}^m$, γ, R, N;

output: $f_N(x) = \sum_{j=1}^N c_j \mathrm{ReLU}(\omega_j \cdot (x_i, 1/\gamma))$;

Generate $\{\omega_j\}_{j=1}^N \subset \mathbb{S}^d$ according to the uniform distribution over $\mathbb{S}^d$;

Choose appropriate loss function ℓ according to the type of tasks, and solve the optimization problem:

$$\underset{\sum_{j=1}^N c_j^2 \leq R^2}{\text{minimize}} \ \frac{1}{m} \sum_{i=1}^m \ell \left(\sum_{j=1}^N c_j \mathrm{ReLU}(\omega_j \cdot (x_i, 1/\gamma)), y_i \right)$$

The first hyper-parameter γ plays a similar role of the bandwidth parameter in random Fourier features method. Since the numerical ranges of features in different data sets may be large, introducing the bandwidth parameter can help normalize the data.

The second hyper-parameter R is the constraint on the 2-norm of the outer weights during the training. Considering the constrained instead of regularized form of optimization simplifies the generalization error analysis, and it is also practical based on our experiment results. In fact, since for a fixed number of random features, the capacity of the hypothesis class is limited, random features methods are less vulnerable than traditional kernel method. This has been confirmed that carefully choosing the batch size and step size can avoid overfitting without use of regularizer or norm constraint in random features methods in regression tasks [6].

When the target function belongs to $\mathcal{H}_{\mathrm{ReLU},\tau_d}$, the standard statistical learning theory guarantees that Algorithm 2 will return a solution with excess risk sufficiently small if the sample size m and the number of features N are large enough and R is chosen to be greater than $2\|f_0\|_{\mathrm{ReLU},\tau_d}$. Then we have the following theorem.

Proposition III.10. *Let ℓ be the hinge or logistic loss, and $f_0 \in \mathcal{H}_{\mathrm{ReLU},\tau_d}$ with norm less than R_{f_0}. If we choose m samples and N random features, where*

$$m \geq \left[\left(4 + 2\sqrt{2\ln\frac{1}{\delta}}\right)\frac{R(\sqrt{r^2+1}+1)}{\epsilon}\right]^2 ,$$

and

$$N \geq \frac{5(r^2+1)}{\epsilon^2}\ln\left(\frac{16(r^2+1)}{\epsilon^2\delta}\right) ,$$

and set $\gamma = 1$ and $R = 2R_{f_0}$ in Algorithm 2. Then, with probability greater than $1-2\delta$, we have

$$\mathbf{R}^\ell(f_N) - \mathbf{R}^\ell(f_0) \leq 3\epsilon ,$$

where f_N is the solution returned by Algorithm 2.

Proof. Recall that the radius of $\mathcal{X}$ is r. Since $f_0 \in \mathcal{H}_{\mathrm{ReLU},\tau_d}$ and its norm is less than R_{f_0}, by Theorem II.2, there exists

$$g : x \mapsto \sum_{i=1}^{N} c_i \mathrm{ReLU}(\omega_i \cdot (x, 1)) ,$$

with $\sum c_i^2 \leq 4R_{f_0}^2$, such that with probability greater than $1-\delta$ over the features $\{\omega_i\}_{i=1}^N$,

$$\|g - f_0\|_{L^2(\mathbb{P})} \leq \epsilon , \tag{3.5}$$

given that

$$N \geq \frac{20(r^2+1)}{\epsilon^2}\ln\left(\frac{64(r^2+1)}{\epsilon^2\delta}\right) .$$

This implies that

$$\mathbf{R}^{\ell}(g) - \mathbf{R}^{\ell}(f_0) \leq \epsilon \,. \tag{3.6}$$

Next, we want to bound the excess risk between the solution f_N returned by the algorithm and g. Denote the RKHS induced by ReLU and the empirical distribution over $\{\omega_i\}_{i=1}^N$ by $\mathcal{H}_N$. Then both f_N and g belongs to $B(2R_{f_0})$, the ball centered at 0 with radius $2R_{f_0}$ in $\mathcal{H}_N$. By the statistical theory ([26]) we know that

$$\mathbf{R}^{\ell}(f_N) - \mathbf{R}^{\ell}(g) \leq 2 \sup_{f \in B(2R_{f_0})} \left| \mathbb{E}\ell(f(x), y) - \frac{1}{m} \sum_{i=1}^{m} \ell(f(x_i), y_i) \right| \tag{3.7}$$

$$\leq 2\mathfrak{R}_m(\ell \circ B(2R_{f_0})) + \sqrt{2 \ln \frac{1}{\delta}} \frac{2R_{f_0}(\sqrt{r^2+1}+1)}{\sqrt{m}} \tag{3.8}$$

$$\leq \frac{4R_{f_0}\sqrt{r^2+1}}{\sqrt{m}} + \sqrt{2 \ln \frac{1}{\delta}} \frac{2R_{f_0}(\sqrt{r^2+1}+1)}{\sqrt{m}} \,, \tag{3.9}$$

(3.10)

with probability greater than $1 - \delta$. Set

$$m = \left(4R_{f_0} + 2R_{f_0}\sqrt{2 \ln \frac{1}{\delta}} \right) \frac{(\sqrt{r^2+1}+1)}{\epsilon^2} \,.$$

Taking the union bound over the two inequalities, the proof is complete. □

By Proposition III.5, we know that for any continuous function f^* and $\epsilon > 0$, there exists $f_0 \in \mathcal{H}_{\mathrm{ReLU},\tau_d}$ such that its RKHS norm is less than R and

$$\|f_0 - f^*\|_{L^2(\mathbb{P})} \leq \epsilon \,.$$

This proves the following statement.

Proposition III.11. *The random ReLU features method (Algorithm 2) is universally consistent.*

To obtain a meaningful learning rate of random ReLU methods beyond the universal consistency of the algorithm. We need to obtain tight upper bounds on the generalization error and approximation error. The generalization error bound tighter than Proposition III.10 can be obtained by using local Rademacher complexity as in Chapter II.

For the approximation error, if a function f does not belong to the RKHS of uniform ReLU features, we want to find a function g in the RKHS to approximate it. In particular, to study the learning rate of random ReLU features support vector machines, we need to know how to construct an approximator in the RKHS for the Bayes classifier $\operatorname{sign}(2\eta(x)-1)$. Here we may adopt the Assumption II.14 that $\Delta x \geq \tau$ for any $x \in \mathcal{X}_1 \cup \mathcal{X}_{-1}$ and $\eta(x) > \frac{1}{2}$ for all x. Again, we consider that $\mathcal{X} \subset \mathbb{S}^d$. Then following the proof of Proposition 3 of [3], we can construct the approximator of the Bayes classifier by

$$g(x) = \int_{\mathbb{S}^d} \operatorname{sign}(2\eta(y)-1) \frac{1-r^2}{(1+r^2-2r(x\cdot y))^{(d+1)/2}} \, d\sigma_d(y) \, .$$

Then by choosing $1-r = C(d)\delta^{-2/(d+1)}$, we have $\|g\|_{RKHS} \leq \delta$ and

$$\sup_{x\in\mathcal{X}} |\operatorname{sign}(2\eta(x)-1) - g(x)| \leq C(d,\tau)\delta^{-2/(d+1)} \, .$$

And we also have $|g(x)| \leq 1$ for all x. If we further have that the spectrum of the kernel operator against the data distribution decays in the rate $\lambda_i = O(i^{-d/2})$, then we can prove that the learning rate of random ReLU with optimized feature sampling will achieve $O(m^{-1/8})$ with m samples and $\sqrt{m}$ random features. If the spectrum of the kernel operator against the data distribution decays exponentially, the learning rate can be largely improved to $O(1/m)$ with $\log m$ features. However, it is not clear how to characterize the data distribution satisfying this requirement on $\mathbb{S}^d$, if such a data distribution exists.

We also test the performance of random ReLU features method on several datasets and make a comparison to random Fourier features in Section 3.7.

3.5 Multi-layer Approximation

In this section, we will prove that the composition of functions in $\mathcal{H}_{\mathrm{ReLU},\tau_d}$ can be approximated by multi-layer ReLU networks with bounded weights.

Barron showed that any function f in Γ_R (see Section 3.2) can be approximated by a single-hidden-layer sigmoidal neural network as described by Equation 3.1, with sigmoidal σ, outer weights $\sum |c_i| < R$ and $L^2(\mathbb{P})$ approximation error less than $R/\sqrt{N}$.

To prove a similar quantitative approximation result for ReLU networks, but with bounds on the inner weights, we consider the function class Λ_R whose members have the representation described by Equation 3.11.

$$f(x) = \int_\Omega \mathrm{ReLU}(\omega \cdot (x, 1))\ \mathrm{d}\mu(\omega)\,, \tag{3.11}$$

where μ is a signed measure satisfying $\int_\Omega |\omega|\ \mathrm{d}|\mu|(\omega) < \infty$. We further denote $\Lambda_R(\mathcal{X})$ to be the subset of $\Lambda(\mathcal{X})$ such that $\int_\Omega |\omega|\ \mathrm{d}|\mu|(\omega) \leq R$.

Compared to Barron's class, our definition of Λ_R uses the Euclidean norm of ω instead of $|\cdot|_\mathcal{X}$. For compact $\mathcal{X}$ with radius r, $|\omega|_\mathcal{X} \leq r|\omega|$. On the other hand, as long as the convex hull of $\mathcal{X} \cup \{0\}$ has non-empty interior, there always exists a constant c such that $|\omega|_\mathcal{X} \geq c|\omega|$. So in most cases $|\cdot|$ and $|\cdot|_\mathcal{X}$ only differ by a constant factor. We should note that the integral representation of f in Equation 3.11 is not unique. There exist cases where the same f can expressed by several different μ's, but only some of them satisfy the constraint of $\Lambda_R(\mathcal{X})$. The original Barron's class exhibits a similar situation. For example, in the case where the Fourier transform of an extension of f to the entire space is integrable, it provides a complex measure

that generates f, and such an extension is not unique.

Some properties of $\Lambda_R(\mathcal{X})$ and $\Lambda(\mathcal{X})$ are given below.

Proposition III.12.

1. *$\Lambda(\mathcal{X})$ consists of continuous functions.*
2. *Functions in $\Lambda_R(\mathcal{X})$ are R-Lipschitz.*
3. *Assume that μ is a probability measure on Ω with $\int_\Omega |\omega|^2 \, \mathrm{d}\mu(\omega) = M_2$. Then $\{f : \|f\|_{\mathcal{H}_{\mathrm{ReLU},\mu}} \leq R\} \subset \Lambda_{R\sqrt{M_2}}(\mathcal{X})$.*
4. *$\Lambda(\mathcal{X})$ is universal.*

Proof.

1. This is implied by the second statement.
2. ReLU is 1-Lipschitz. For f in $\Lambda_C(\mathcal{X})$ and x_1, x_2 in $\mathcal{X}$,

$$\begin{aligned} |f(x_1) - f(x_2)| &= \left| \int (\mathrm{ReLU}(\omega \cdot (x_1, 1)) - \mathrm{ReLU}(\omega \cdot (x_2, 1)) \, \mathrm{d}\rho(\omega) \right| \\ &\leq \int |\mathrm{ReLU}(\omega \cdot (x_1, 1) - \mathrm{ReLU}(\omega \cdot (x_2, 1)| \, \mathrm{d}|\rho|(\omega) \\ &\leq \int |\omega \cdot (x_1 - x_2, 0)| \, \mathrm{d}|\rho|(\omega) \\ &\leq R|x_1 - x_2| \, . \end{aligned}$$

3. For any f in the ball of radius R of the RKHS $\mathcal{H}_{\mathrm{ReLU},\mu}$, we have that

$$f(x) = \int \sigma(\omega \cdot (x, 1)) g(\omega) \, \mathrm{d}\mu(\omega) \quad \text{and} \quad \int g^2(\omega) \, \mathrm{d}\mu(\omega) \leq R^2 \, .$$

Then

$$\begin{aligned} \int |\omega||g(\omega)| \, \mathrm{d}\mu(\omega) &\leq \left(\int \omega \cdot \omega \, \mathrm{d}\mu(\omega) \int g^2(\omega) \, \mathrm{d}\mu(\omega) \right)^{1/2} \\ &\leq \sqrt{M_2} R \, . \end{aligned}$$

4. Since

$$\Lambda(\mathcal{X}) \supset \bigcup_{R>0} \left\{ f \in \mathcal{H}_{\mathrm{ReLU},\mu} \mid \|f\|_{\mathcal{H}_{\mathrm{ReLU},\mu}} \leq R \right\}$$
$$= \mathcal{H}_{\mathrm{ReLU},\mu}\,,$$

$\Lambda(\mathcal{X})$ is universal, by Proposition III.4. This can also be shown by the fact that Equation 3.1 belongs to Λ and the set of neural networks with ReLUs as activation nodes is universal.

□

Later we will directly extend to the multi-layer ReLU network approximation results for Λ to $\mathcal{H}_{\mathrm{ReLU},\tau_d}$ using the third property in Proposition III.12. Usually $\mathcal{H}_{\mathrm{ReLU},\mu}$ is strictly included in Λ and so is Λ in $C(\mathcal{X})$. Indeed, when μ is absolutely continuous with respect to Lebesgue measure, for any fixed ω, $\mathrm{ReLU}(\omega\cdot(x,1))$ belongs to $\Lambda(\mathcal{X})$, but not $\mathcal{H}_{\mathrm{ReLU},\mu}$. And $\Lambda(\mathcal{X})$ only contains Lipschitz functions, but there clearly exist continuous functions over $\mathcal{X}$ that are not Lipschitz.

For the functions in $\Lambda_R(\mathcal{X})$, we can approximate them by ReLU networks. Stronger than Barron's approximation theorem, our theorem provides $O(1)$ upper bounds for both inner and outer weights of f_N.

Theorem III.13. *Assume that $\mathcal{X}$ is bounded by a ball of radius r. $\mathbb{P}$ is a probability measure on $\mathcal{X}$. For any $f \in \Lambda_R(\mathcal{X})$, there exists $g(x) = \sum_{i=1}^{N} c_i \mathrm{ReLU}(\tilde{\omega}_i \cdot (x,1))$ with $|c_i| = \tilde{R}/N \leq R/N$, and $\|\tilde{\omega}_i\| = 1$ for all i, such that $\|f-g\|_{L^2(\mathbb{P})} \leq R\sqrt{r^2+1}/\sqrt{N}$.*

It is not surprising that as with Barron's work, the key step of the proof is the Maurey's sparsification lemma (see Section 3.1). With this lemma, we can prove our main theorem.

Proof. If f is constantly 0, we only need to choose c_i to be 0 for all $i \in [N]$. Now assume that f is not constantly 0. Then we can always restrict the feature space to $S = \mathbb{R}^{d+1} \backslash \{0\}$ and have $0 < \int_S |\omega| \, \mathrm{d}|\rho|(\omega) = \tilde{R} \leq R$. f can be written into the following form

$$f(x) = \int_S \tilde{R}\tau(\omega)\mathrm{ReLU}\left(\frac{\omega \cdot (x,1)}{|\omega|}\right) \frac{|\omega|}{\tilde{R}} \, \mathrm{d}|\rho|(\omega)\,,$$

where $\tau(\omega)$ equals 1 when ω belongs to the positive set of ρ and -1 when it belongs to the negative set of ρ. Since $\frac{|\omega|}{\tilde{R}} \, \mathrm{d}|\rho|(\omega)$ is a probability measure and

$$\left\| \tilde{R}\tau(\omega)\mathrm{ReLU}\left(\frac{\omega \cdot (x,1)}{|\omega|}\right) \right\|_{L^2(\mathbb{P})} \leq \tilde{R}\sqrt{r^2+1}\,,$$

we can apply Lemma III.3 and get the conclusion that there exists ω_is such that

$$\left\| f - \frac{1}{N}\sum_{i=1}^{N} \tilde{R}\tau(\omega_i)\mathrm{ReLU}\left(\frac{\omega_i \cdot (x,1)}{|\omega_i|}\right) \right\|_{L^2(\mathbb{P})} \leq \frac{\tilde{R}\sqrt{r^2+1}}{\sqrt{N}}\,.$$

By setting $\tilde{\omega}_i = \omega_i/|\omega_i|$ and $c_i = \tilde{R}\tau(\omega_i)/N$, the statement is proved. □

Our theorem shows that for ReLU networks to approximate the functions in $\Lambda_R(\mathcal{X})$ within an error of ϵ, only $O(C/\epsilon^2)$ nodes are required. Moreover, every outer weights in front of the nodes are bounded by $O(\epsilon^2)$ and only differ by signs, and the inner weights are of unit length. Compared with the theorem in [5], where outer weights are bounded by a constant under ℓ^1 norm, but bound on inner weights depends inversely on the approximation error, our extra bounds on inner weights largely shrink the search area for approximators. This improvement comes from the fact that functions in $\Lambda_R(\mathcal{X})$ are defined by the transformation induced by ReLU functions, which is homogeneous of degree 1.

Inspired by [22]'s work, we also extend our result to the composite of functions in Λ_R. First, we need to define the class $\Lambda_R(\mathcal{X})$ for vector-valued functions.

Definition III.14. For a map f from $\mathbb{R}^m$ to $\mathbb{R}^n$, we say that it belongs to the class $\Lambda_R(\mathcal{X})$ if each component $(f)_i \in \Lambda_{R_i}(\mathcal{X})$ for $1 \le i \le n$ and $\sum_i R_i^2 \le R^2$.

Note that Proposition III.12 (2) still holds for the vector valued function class Λ_R.

The following theorem, as an analogy to Theorem 3.5 in [22], shows that the composition of functions in Λ_R can be approximated by a multi-layer ReLU network to approximate it. And all the weights in the neural network can be controlled by constants related to R and dimensions of functions. This result justifies what type of functions can be learned when the weight matrices of multi-layer ReLU networks are constrained by some constants.

Proposition III.15. *Assume that for all $1 \le i \le L+1$, K_i is a compact set with radius r in $\mathbb{R}^{m_i}$, among which $K_1 = \mathcal{X}$. Let B_{m_i} denote the unit ball in $\mathbb{R}^{m_i}$. $\mathbb{P}$ is a probability measure on K_1. $f_i : \mathbb{R}^{m_i} \to \mathbb{R}^{m_{i+1}}$ belongs to $\Lambda_{R_i}(K_i + sB_{m_i})$ for an $s > 0$ and any $1 \le i \le L$. Then for $\epsilon > 0$, there exists a set $S_L \subset K_1$ with*

$$\mathbb{P}(S_L) > 1 - \frac{\epsilon^2}{s^2} \sum_{i=1}^{L-1} \frac{i^2}{\prod_{j=i+1}^{L} R_j^2},$$

and an L-layer neural networks $g_{1:L}$ where

$$g_i : \mathbb{R}^{m_i} \to \mathbb{R}^{m_{i+1}}$$
$$(g_i(x))_j = \sum_{k=1}^{N_i} c_{i,j,k} \mathrm{ReLU}(\omega_{i,j,k} \cdot (x, 1))$$
$$N_i = \frac{\prod_{j=i}^{L} R_j^2((r+s)^2+1)}{\epsilon^2}$$

such that

$$\left(\int_{S_L} \|f_{1:L} - g_{1:L}\|^2 \, \mathrm{d}\mathbb{P} \right)^{1/2} \le L\epsilon.$$

Moreover, the ith layer of the neural network $g_{1:\ell}$ contains $m_{i+1}N_i$ nodes.

Let $W_{i\to i+1}$ denote the weight matrix from layer i to layer $i+1$ for $0 \le i \le L$, where $W_{0\to 1}$ denotes the weights from input x to the first layer of nodes and $W_{L\to L+1}$ denotes the weights from the last layer to the output. Then we have

$$\|W_{0\to 1}\|_F \le \prod_{i=1}^{L} R_i \sqrt{m_2((r+s)^2+1)}\,,$$

$$\|W_{i\to i+1}\|_F \le \sqrt{m_{i+2}} \text{ for } 1 \le i \le L-1\,,$$

$$\|W_{L\to L+1}\|_F \le \sqrt{(r+s)^2+1}\,.$$

Each bias term is bounded by 1.

Proof. For $\ell = 1$, we construct the approximation g_1 for f_1 by applying Theorem III.13 to each component of f_1. First, set $S_1 = K_1$.

$$\begin{aligned}\int_{S_1} \|g_1(x) - f_1(x)\|^2 \,\mathrm{d}\mathbb{P}(x) &= \sum_{i=1}^{m_2} \int_{S_1} (f_1(x) - g_1(x))_i^2 \,\mathrm{d}\mathbb{P}(x) \\ &\le \sum_{i=1}^{m_2} \frac{R_{1,i}^2((r+s)^2+1)}{N_1} \\ &\le \frac{R_1^2((r+s)^2+1)}{N_1}\,.\end{aligned}$$

Set

$$N_1 = \frac{\prod_{i=1}^{L} R_i^2((r+s)^2+1)}{\epsilon^2}$$

and the conclusion holds for $\ell = 1$. Assume that there exist $g_{1:\ell}$ and S_ℓ as described in the theorem. Define

$$S_{\ell+1} = S_\ell \cap \{x \in K_1 : \|g_{1:\ell}(x) - f_{1:\ell}(x)\| \le s\}\,.$$

Then by Markov's inequality and induction assumption,

$$\mathbb{P}(S_{\ell+1}) \ge 1 - \frac{\epsilon^2}{s^2} \sum_{i=1}^{\ell-1} \frac{i^2}{\prod_{j=i+1}^{L} R_j^2} - \frac{\epsilon^2 \ell^2}{s^2 \prod_{i=\ell+1}^{L} R_i^2}\,.$$

Then we want to construct $g_{\ell+1}$ on $g_{1:\ell}(S_{\ell+1})$ to approximate $f_{\ell+1}$, again by applying Theorem III.13 to each component of $f_{\ell+1}$. Note that the measure we consider here

is the push-forward of $\mathbb{P}$ by $g_{1:\ell}$, which is a positive measure with total measure less than 1.

$$\int_{g_{1:\ell}(S_{\ell+1})} \|g_{\ell+1}(x) - f_{\ell+1}(x)\|^2 \, \mathrm{d}g_{1:\ell}(\mathbb{P})(x) \leq \frac{R_{\ell+1}^2((r+s)^2+1)}{N_{\ell+1}} \,.$$

And by triangle inequality,

$$\begin{aligned}
&\left(\int_{S_{\ell+1}} \|g_{1:\ell+1}(x) - f_{1:\ell+1}(x)\|^2 \, \mathrm{d}\mathbb{P}(x)\right)^{1/2} \\
&\leq \left(\int_{S_{\ell+1}} \|g_{\ell+1} \circ g_{1:\ell}(x) - f_{\ell+1} \circ g_{1:\ell}(x)\|^2 \, \mathrm{d}\mathbb{P}(x)\right)^{1/2} \\
&+ \left(\int_{S_{\ell+1}} \|f_{\ell+1} \circ g_{1:\ell}(x) - f_{\ell+1} \circ f_{1:\ell}(x)\|^2 \, \mathrm{d}\mathbb{P}(x)\right)^{1/2} \\
&\leq \frac{R_{\ell+1}\sqrt{(r+s)^2+1}}{\sqrt{N_{\ell+1}}} + R_{\ell+1}\frac{\ell\epsilon}{\prod_{i=\ell+1}^{L} R_i} \,.
\end{aligned}$$

Set

$$N_{\ell+1} = \frac{((r+s)^2+1)\prod_{i=\ell+1}^{L} R_i^2}{\epsilon^2} \,,$$

and we get the upper bound

$$\left(\int_{S_{\ell+1}} \|g_{1:\ell+1}(x) - f_{1:\ell+1}(x)\|^2 \, \mathrm{d}\mathbb{P}(x)\right)^{1/2} \leq \frac{(\ell+1)\epsilon}{\prod_{i=\ell+2}^{L} R_i} \,.$$

Now let's examine the weight matrix and the bias term of the neural network $g_{1:\ell}$. Note that the weight matrix from ith layer to the $(i+1)$th layer of $g_{1:\ell}$ consists of $c_{i,j,k}$ from g_i and $\omega_{i+1,j,k}$ from g_{i+1}. For $1 \leq i \leq L-1$, the exact form of the weight

matrices is given by the matrix product $W_{i\to i+1} = AB$, where

$$A_{i\to i+1} = \begin{pmatrix} (\omega_{i+1,1,1})_1 & \cdots & (\omega_{i+1,1,1})_{m_{i+1}} \\ \vdots & \ddots & \vdots \\ (\omega_{i+1,1,N_{i+1}})_1 & \cdots & (\omega_{i+1,1,N_{i+1}})_{m_{i+1}} \\ (\omega_{i+1,2,1})_1 & \cdots & (\omega_{i+1,2,1})_{m_{i+1}} \\ \vdots & \ddots & \vdots \\ (\omega_{i+1,2,N_{i+1}})_1 & \cdots & (\omega_{i+1,2,N_{i+1}})_{m_{i+1}} \\ \vdots & \ddots & \vdots \\ (\omega_{i+1,m_{i+2},1})_1 & \cdots & (\omega_{i+1,m_{i+2},1})_{m_{i+1}} \\ \vdots & \ddots & \vdots \\ (\omega_{i+1,m_{i+2},N_{i+1}})_1 & \cdots & (\omega_{i+1,m_{i+2},N_{i+1}})_{m_{i+1}} \end{pmatrix}$$

$$B_{i\to i+1} \begin{pmatrix} \frac{\tilde{R}_{i,1}\tau_{i,1,1}}{N_i} \cdots \frac{\tilde{R}_{i,1}\tau_{i,1,N_i}}{N_i} & 0\cdots 0 & 0\cdots 0 \\ 0\cdots 0 & \frac{\tilde{R}_{i,2}\tau_{i,2,1}}{N_i} \cdots \frac{\tilde{R}_{i,2}\tau_{i,2,N_i}}{N_i} & 0\cdots 0 \\ \vdots & \ddots & \vdots \\ 0\cdots 0 & 0\cdots 0 & \frac{\tilde{R}_{i,m_{i+1}}\tau_{i,m_{i+1},1}}{N_i} \cdots \frac{\tilde{R}_{i,m_{i+1}}\tau_{i,m_{i+1},N_i}}{N_i} \end{pmatrix}.$$

$\tau_{i,j,k}$s are all ± 1. Since $|\omega_{i,j,k}| \le 1$, each components are also bounded by 1. For any row of $W_{i\to i+1}$,

$$\begin{aligned} \|(W_{i\to i+1})_k\|^2 &= \sum_{j=1}^{m_{i+1}} N_i \frac{\tilde{R}_{i,j}^2}{N_i^2} \\ &\le \frac{R_i^2}{N_i}. \end{aligned}$$

Hence $\|W_{i\to i+1}\|_F^2 \le R_i^2 N_{i+1} m_{i+2}/N_i$. Plugging into the expressions of N_i and N_{i+1} and taking square root, we get the upper bound m_{i+2} as in the statement. For the bias term, we simply use the fact that it is a coordinate of ω and thus bounded by 1.

For the bottom layer the weight matrix is just given by $A_{0\to 1}$, its Frobenius norm can only bounded by $(N_1 m_2)^{1/2}$. On the other hand, the weight matrix of the top

layer is given by $B_{L\to L+1}$, whose Frobenius norm is bounded by $R_L/\sqrt{N_L}$. Since N_i is of the scale $1/\epsilon^2$, $\|A_{0\to 1}\|_F$ is of the scale ϵ while $\|B_{L\to L+1}$ is of the scale $1/\epsilon$. To resolve this issue, we can rescale $A_{0\to 1}$ down by a factor of ϵ. Because the ReLU nodes are homogeneous of degree 1, for the function $g_{1:L}$ remains unchanged, we need only scale all the bias terms in the intermediate layers down by the same factor ϵ, and scale up $B_{L\to L+1}$ by the factor of $1/\epsilon$. For $\epsilon < 1$, this rescaling keeps the weights matrices in the intermediate layers unchanged, the bias terms are still all less than 1. And

$$\|\epsilon A_{0\to 1}\|_F \le \prod_{i=1}^{L} R_i \sqrt{m_2((r+s)^2+1)}\,,$$

$$\|B_{L\to L+1}/\epsilon\|_F \le \sqrt{(r+s)^2+1}\,.$$

□

Let's define the RKHS norm for a vector valued function as the 2-norm of the RKHS norm of its component; that is, for $f : \mathbb{R}^n \to \mathbb{R}^m$ and a RKHS $\mathcal{H}$ on $\mathbb{R}^n$, $\|f\|_{\mathcal{H}}^2 \coloneqq \sum_{i=1}^m \|(f)_i\|_{\mathcal{H}}^2$.

If we consider the random ReLU features whose feature distribution μ is supported on $\mathbb{S}^d$, by Proposition III.12 (3), we have that for any f with $\|f\|_{\mathrm{ReLU},\mu} \le R$, $f \in \Lambda_R(\mathcal{X})$. So for composition of functions in $\mathcal{H}_{\mathrm{ReLU},\mu}$ we can also construct multi-layer ReLU networks to approximate it. The depth separation result in the next section shows that this is a specific advantage of deep networks compared to shallow ones.

3.6 Depth Separation for $\mathcal{H}_{\mathrm{ReLU},\tau_d}$

In this section, we prove that the depth separation result similar to [12] and [22] also holds for functions in $\mathcal{H}_{\mathrm{ReLU},\tau_d}$; that is, functions with poly(d) RKHS norm in $\mathcal{H}_{\mathrm{ReLU},\tau_d}$ cannot approximate compositions of such functions.

Proposition III.16. *There exist universal constants c, C such that for any $d > C$, there exist a probability measure $\mathbb{P}$ on $\mathbb{R}^d$, and two functions $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R} \to \mathbb{R}$ with $\|f\|_{\mathrm{ReLU},\tau_d}$ and $\|g\|_{\mathrm{ReLU},\tau_d}$ less than $\mathrm{poly}(d)$ so that the following holds. For every two-layer ReLU networks h,*

$$\|h - f \circ g\|_{L^2(\mu)} \geq c\,.$$

The proof is based on the construction of [12], but we need to modify the functions considered in their work so that they belong to $\mathcal{H}_{\mathrm{ReLU},\tau_d}$ with $\mathrm{poly}(d)$ norm. This result shows that the composition of functions in $\mathcal{H}_{\mathrm{ReLU},\tau_d}$ generates substantially more complicated functions than those in $\mathcal{H}_{\mathrm{ReLU},\tau_d}$.

Before the proof of the depth separation, we first show how to upper bound the RKHS norm $\|\cdot\|_{\mathrm{ReLU},\tau_d}$ of the function $\sqrt{|x|^2+1}$. To get the upper bound, we start with the polynomials of projection $f(\tilde{x}) = \alpha(\beta \cdot \tilde{x})^p$ defined on $\mathbb{S}^d$, and its RKHS norm in the space induced by $\mathrm{ReLU}(\omega \cdot \tilde{x})$ and τ_d. We denote this RKHS by $\tilde{\mathcal{H}}$. And we use $\tilde{x}$ to emphasize that these are points on $\mathbb{S}^d$, while x represents a point in $\mathbb{R}^d$.

A sufficient condition for the membership of RKHS induced by the random feature (ReLU, τ_d) is provided in [4]. It shows that functions with bounded $\frac{d}{2}+\frac{3}{2}$ derivatives are in $\mathcal{H}_{\mathrm{ReLU},\tau_d}$, and the RKHS norm is upper bounded by the $\lceil \frac{d}{2}+\frac{3}{2} \rceil$th derivative. For the polynomials of projection, we provide a different way to evaluate a tighter upper bound of its norm.

The tool we use here is the spherical harmonics. The kernel derived from the uniform ReLU feature (ReLU, τ_d) can be explicitly written as follows [7],

$$\int_{\mathbb{S}^d} \mathrm{ReLU}(\omega \cdot \tilde{x})\mathrm{ReLU}(\omega \cdot \tilde{x}')\, d\tau_d(\omega) \coloneqq \frac{\sqrt{1-(\tilde{x}\cdot\tilde{x}')^2} + (\pi - \arccos(\tilde{x}\cdot\tilde{x}'))\,\tilde{x}\cdot\tilde{x}'}{2(d+1)\pi}\,,$$

Note that we rewrite the kernel function in a different but equivalent form compared to the formula in Cho's work to emphasize that it is of the form $k(\tilde{x} \cdot \tilde{y})$, so

called dot product kernels. The kernel function $k(s)$ has the Taylor expansion

$$\frac{1}{2(d+1)}\left(\frac{1}{\pi}+\frac{1}{2}s+\frac{1}{2\pi}s^2+\frac{1}{\pi}\sum_{k=2}^{\infty}\frac{(2k-3)!!}{(2k)!!}\frac{1}{2k-1}s^{2k}\right)=\sum_{j=0}^{\infty}a_j s^j\,.$$

Then by [2], we know that $k(\tilde{x}\cdot\tilde{y})$ has the Mercer type expansion

$$k(\tilde{x}\cdot\tilde{y})=\sum_{i=1}^{\infty}\sum_{j=1}^{N(i,d)}\lambda_i Y_{i,j}^d(\tilde{x})Y_{i,j}^d(\tilde{y})\,,$$

where $\{Y_{i,j}^d\}$ are the spherical harmonics on $\mathbb{S}^d$, $N(i,d)$ is the dimension of the eigenspace corresponding to ith eigenvalue and

$$\begin{aligned}\lambda_i &= |\mathbb{S}^{d-1}|\int_{-1}^{1}k(s)P_i^d(s)(1-s^2)^{\frac{d-2}{2}}\,ds\\ &= |\mathbb{S}^{d-1}|\int_{-1}^{1}\sum_{j=0}^{\infty}a_j s^j P_i^d(s)(1-s^2)^{\frac{d-2}{2}}\,ds\\ &= \sum_{j=0}^{\infty}a_j c(i,j,d)\,.\end{aligned}$$

$c(i,j,d)>0$ when $i\le j$ and $i\equiv j \mod 2$ and $c(i,j,d)=0$ otherwise. Now consider the function $f(\tilde{x})=\alpha(\beta\cdot\tilde{x})^p$. Without loss of generality, we assume that $\beta\in\mathbb{S}^d$. Its projection onto the subspace spanned by $Y_{i,j}^d$ is given by the Funk-Hecke formula

$$\int_{\mathbb{S}^d}\alpha(\beta\cdot\tilde{x})^p Y_{i,j}^d(\tilde{x})\,d\sigma_d(\tilde{x})=\alpha c(i,p,d)Y_{i,j}^d(\beta)\,.$$

In other words, for even number p or $p=1$,

$$\alpha(\beta\cdot\tilde{x})^p=\sum_{i=1}^{p}\sum_{j}\alpha c(i,p,d)Y_{i,j}^d(\beta)Y_{i,j}^d(\tilde{x})\,.$$

Then the RKHS norm of f is given by

$$\|f\|_{\tilde{\mathcal{H}}}^2=\alpha^2\sum_{i=1}^{p}\sum_{j}\frac{c^2(i,p,d)}{\lambda_i}\left(Y_{i,j}^d(\beta)\right)^2\,.$$

Note that $\lambda_i\ge a_p c(i,p,d)$. So we have

$$\begin{aligned}\|f\|_{\tilde{\mathcal{H}}}^2 &\le \alpha^2\sum_{i=1}^{p}\sum_{j}a_p^{-1}c(i,p,d)\left(Y_{i,j}^d(\beta)\right)^2\\ &= a_p^{-1}\alpha^2(\beta\cdot\beta)^p\\ &= a_p^{-1}\alpha^2\,.\end{aligned}$$

Since $a_p \geq \frac{1}{2(d+1)\pi p^2}$, for $p = 1$ or even p, we have the following lemma.

Lemma III.17. $\|\alpha(\beta \cdot \tilde{x})^p\|_{\tilde{\mathcal{H}}} \leq \sqrt{2(d+1)\pi}\alpha p$.

Using this result, we can further justify whether a function belongs to the $\tilde{\mathcal{H}}$ based on its Taylor expansion.

Next, we show that the function $g(x) = \alpha(\beta \cdot (x, 1))^p(1 + \|x\|^2)^{(1-p)/2}$ defined on $\mathbb{R}^d$ belongs to $\mathcal{H}_{\mathrm{ReLU},\tau_d}$. Assume that $f(\tilde{x}) = \alpha(\beta \cdot \tilde{x})^p$, then

$$\begin{aligned}\alpha(\beta \cdot (x, 1))^p(1 + x^2)^{(1-p)/2} &= \sqrt{\|x\|^2 + 1} f\left(\frac{(x, 1)}{\sqrt{\|x\|^2 + 1}}\right) \\ &= \int_{\mathbb{S}^d} h(\omega)\mathrm{ReLU}(\omega \cdot (x, 1))\, \mathrm{d}\tau_d(\omega)\,,\end{aligned}$$

for some $h \in L^2(\mathbb{S}^d, \tau_d)$. So

$$\|g|_{\mathrm{ReLU},\tau_d} \leq \|h\|_{L^2(\tau_d)} = \|f\|_{\tilde{\mathcal{H}}} \leq \sqrt{2(d+1)\pi}\alpha p\,.$$

When $p = 2, \alpha = 1$ and $\beta = \vec{e}_j$, we have

$$\left\| \frac{x_j^2}{\sqrt{1 + \|x\|^2}} \right\|_{\mathrm{ReLU},\tau_d} \leq 2\sqrt{2(d+1)\pi}\,.$$

And

$$\begin{aligned}\left\| \sqrt{1 + \|x\|^2} \right\|_{\mathrm{ReLU},\tau_d} &\leq \sum_{j=1}^{d} \left\| \frac{x_j^2}{\sqrt{1 + \|x\|^2}} \right\|_{\mathrm{ReLU},\tau_d} \\ &\leq 2\sqrt{2\pi}(d+1)^{3/2}\,. \qquad (3.12)\end{aligned}$$

Now we prove Proposition III.16.

Proof. For any $\omega_i \in \mathbb{S}^d$ and $c_i \in \mathbb{R}$, we define

$$f_i(t) = c_i\mathrm{ReLU}(t(1 - (\omega_i)_{d+1}^2)^{1/2} + (\omega_i)_{d+1})\,.$$

Then

$$f_i\left(x \cdot \frac{(\omega_i)_{1:d}}{\|(\omega_i)_{1:d}\|}\right) = c_i\mathrm{ReLU}(\omega_i \cdot (x, 1))\,,$$

where we use the convention that $0/\|0\| = 0$. Then by Proposition 13 of [12], we know for any ω_i and c_i, there exists a function $\tilde{g}$ and data distribution $\mathbb{P}$ such that

$$\left\| \sum_{i=1}^{N} f_i \left(x \cdot \frac{(\omega_i)_{1:d}}{\|(\omega_i)_{1:d}\|} \right) - \tilde{g}(x) \right\|_{L^2(\mathbb{P})} \geq \frac{\delta}{\alpha},$$

where δ and α are two constants. Now we construct $\tilde{g}$ as a composite function $g \circ f$. We choose $f(x) = \sqrt{\|x\|^2 + 1}$. By Equation 3.12, $\|f\|_{\mathcal{H}} \leq 2\sqrt{2\pi}(d+1)^{3/2}$. The data distribution is also chosen to be the squared Fourier transform of the indicator function of unit volume ball, the same with [12]. Then we construct g based on the function $\tilde{g}$ in Proposition 13 in [12]. By Lemma 12 in Eldan and Shamir's work, we know that there exists N-Lipschitz function f_1 where $N = c(\alpha d)^{3/2}$ supported on $[\alpha\sqrt{d}, 2\alpha\sqrt{d}]$ with range in $[-1, 1]$ such that

$$\|f_1(\|x\|) - \tilde{g}(\|x\|)\|^2_{L^2(\mathbb{P})} \leq \frac{3}{\alpha^2\sqrt{d}}.$$

Now we define $f_2(t) = f_1(\sqrt{t^2 - 1})$. It is supported on $[\sqrt{\alpha^2 d + 1}, \sqrt{4\alpha^2 d + 1}]$ and $3N$-Lipschitz. And we have $f_2(\sqrt{\|x\|^2 + 1}) = f_1(\|x\|)$. The last step is to find an approximator from $\mathcal{H}_{\mathrm{ReLU},\tau_d}$ for f_2. This can be done be applying Proposition 6 in [4], which implies in our setup that there exists a function f with $\|f\|_{\mathcal{H}} \leq M$ and

$$\sup_{|t| \leq \sqrt{4\alpha^2 d+1}} |f(t) - f_2(t)| \leq CN\sqrt{4\alpha^2 d + 1} \left(\frac{M}{3N\sqrt{4\alpha^2 d + 1}} \right)^{-1/2},$$

where C is a universal constant. By setting $M = (3N\sqrt{4\alpha^2 d + 1})^{3/2}\alpha^2\sqrt{d}$, we have

$$\|f \circ g - \tilde{g}\|_{L^2(\mathbb{P})} \leq O\left(\frac{1}{\alpha d^{1/4}} \right).$$

Therefore, we proved that

$$\left\| \sum_{i=1}^{N} f_i \left(x \cdot \frac{(\omega_i)_{1:d}}{\|(\omega_i)_{1:d}\|} \right) - f \circ g(x) \right\|_{L^2(\mathbb{P})} \geq \frac{\delta}{2\alpha} \coloneqq c,$$

□

3.7 Experiments

We compared the performance of random ReLU features method to the popular random Fourier features with Gaussian feature distribution on four synthetic data sets (see Figure 3.7) and three real data sets: MNIST [21], adult and covtype [10]. Our purpose is to show that in practice, random ReLU features method display comparable performance with random Fourier features models and have several advantages over random Fourier features with respect to computational efficiency.

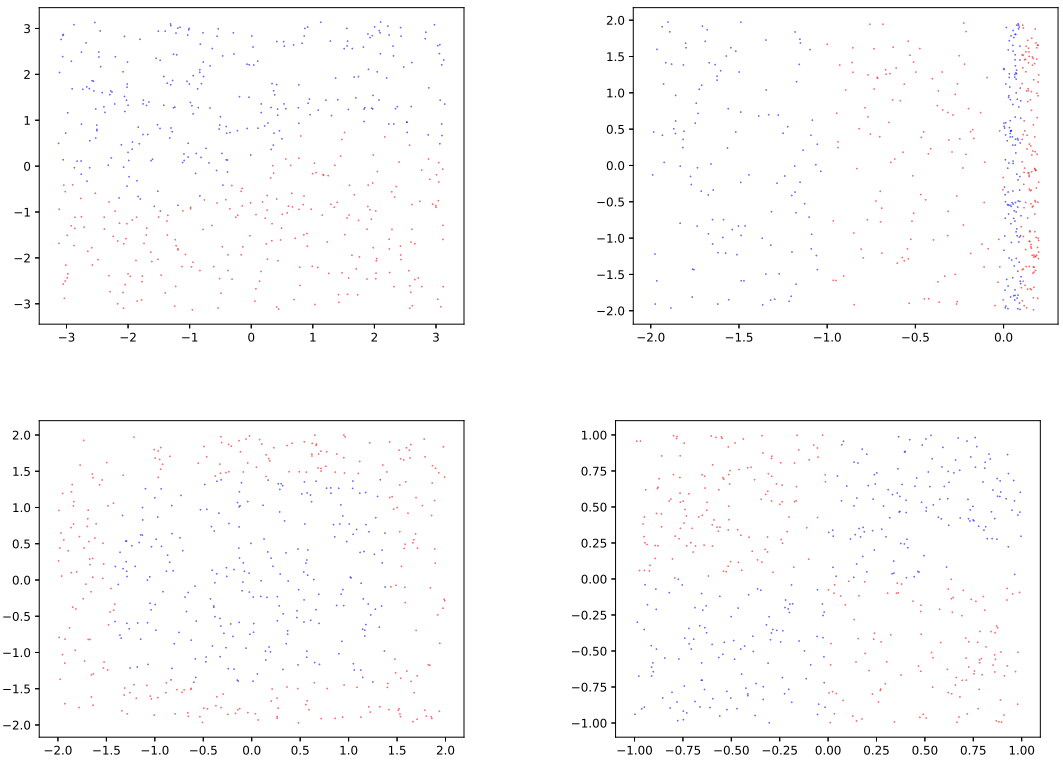


Figure 3.1: Illustration of distributions of synthetic data. Top left: sine. Top right: strips. Bottom left: square. Bottom right: checkboard.

For all four synthetic data sets, we used 20 random features for each method; for real data sets we used 2000 random features. For the binary classification tasks, we used hinge loss. For the multi-class classification tasks like MNIST and covtype, we

chose logistic loss. Even though adding a regularization term is popular in practice, we chose to constrain the 2-norm of the outer weights by a large constant (1000 for synthetic data sets and 10000 for real data sets) as described in Section 3.4. The optimization method was the plain stochastic gradient descent and the model was implemented using Tensorflow [24]. The learning rate and bandwidth were screened carefully for both models through grid search.

In Figure 3.2, we present the dependence of two methods on the bandwidth parameters in the screening step. Each point displays the best 5-fold cross validation accuracy among all learning rates. We can see that the performance of random Fourier features with Gaussian distribution is more sensitive to the choice of bandwidth than random ReLU features.

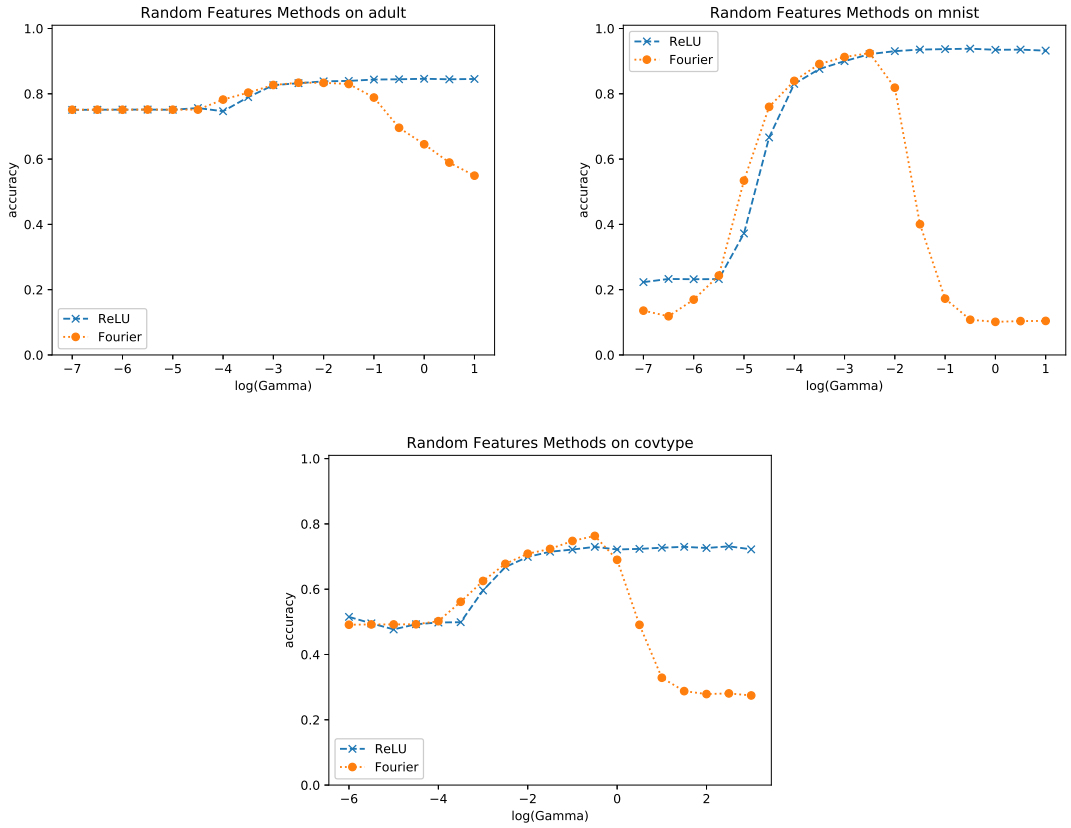


Figure 3.2: Cross validation accuracy of random Fourier features and random ReLU features. Top left: adult. Top right: mnist. Bottom: covtype.

We list the accuracy and training time for two methods in Table 3.1. We can see that on all the data sets, random ReLU features method requires shorter training time. It outperforms random Fourier features with higher accuracy on adult and MNIST data sets. Its performance is similar to random Fourier features on sine, checkboard and square. However, its performance on strips and covtype data set is significantly worse.

The training time and testing time (not listed) of random ReLU features are always shorter than random Fourier features. This is mainly because half of random ReLU feature vectors coordinates are zero. For random Fourier features, we cannot expect that.

Table 3.1: Left: accuracy of random ReLU features versus random Fourier features. Right: training time of random ReLU features versus random Fourier features; Time is shown in seconds.

	Fourier	ReLU
sine	0.993(0.007)	0.984(0.005)
strips	0.834(0.084)	0.732(0.006)
square	0.948(0.038)	0.934(0.015)
checkboard	0.716(0.045)	0.743(0.027)
adult	0.838(0.002)	0.846(0.002)
mnist	0.937(0.001)	0.951(0.001)
covtype	0.816(0.001)	0.769(0.002)

	Fourier	ReLU
sine	1.597(0.050)	1.564(0.052)
strips	1.598(0.056)	1.565(0.052)
square	1.769(0.061)	1.743(0.057)
checkboard	1.581(0.078)	1.545(0.073)
adult	6.648(0.181)	5.849(0.216)
mnist	70.438(0.321)	69.229(1.080)
covtype	125.719(0.356)	112.613(1.558)

The depth separation and multi-layer approximation results in Section 3.5 only prove the existence of the advantage of deeper models. It is not clear whether we can find good multi-layer approximators with significant better performance than shallow models. Therefore, we designed a synthetic data set that is supposed to hard to learn by shallow models according to the depth separation result to test depth separation phenomenon. The data distribution is rotation invariant over $\mathbb{R}^4$ with rapid oscillating density. The classification target function g is ± 1 at the high density region and 0 otherwise. The regression target function $\tilde{g}$ is obtained by mollifying g with the smooth bump function. See Figure 3.7 for the illustration of

the data distribution and the target functions.

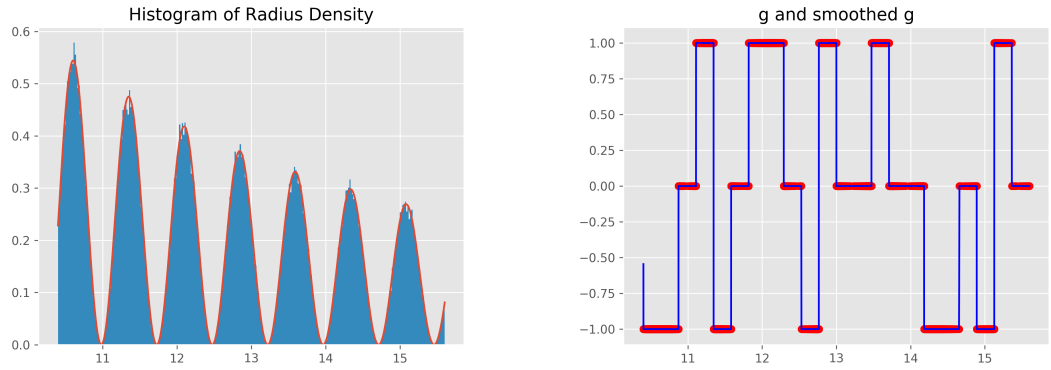


Figure 3.3: Illustration of distributions of synthetic data for depth separation experiment. Left: radial density of data. Right: Smoothed (blue) and unsmoothed (red) target labels.

For both classification and regression tasks, we trained three models: random ReLU models, 2-layer neural nets, and 3-layer neural nets. The random ReLU models and 2-layer neural nets have exactly the same structure. The only difference is whether the inner weights are randomly chosen or trained together with top layer. The performance is examined for two models with 20 nodes up to 5120 nodes. The 3-layer neural nets are simply fully connected with the equal width in each layer. To make a fair comparison, we maintain the total number of parameters of the 3-layer neural nets to be the same with the shallow models at each level.

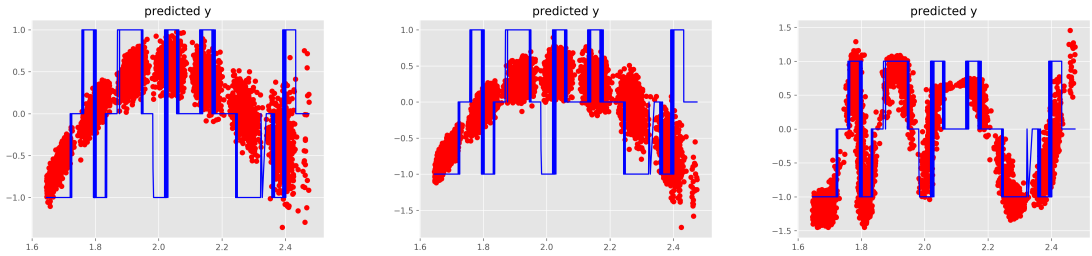


Figure 3.4: Performance of the deep and shallow models in the regression task. Left: the predicted labels (red) by random ReLU models compared to the true labels (blue) plotted against normalized radius of data. Middle: the predicted labels (red) by 2-layer neural nets compared to the true labels (blue) plotted against normalized radius of data. Right: the predicted labels (red) by 3-layer neural nets compared to the true labels (blue).

From Figure 3.7 we can see that 3-layer neural nets consistently achieve significantly better performance and the gap between 2-layer neural nets and random ReLU features is not as obvious. By plotting the predicted labels on tested data in Figure 3.4, we can see that 3-layer neural nets indeed learned a function closer to the true function. The random ReLU models, however, learned a more regular function and did not adapt to the rapid oscillation of the true labels. It is surprising that 2-layer neural nets' performance is not significantly better than random ReLU and the learned function is also quite similar to the one learned by random ReLU. This may imply that the structure of the model has a more important impact on the performance than the number of adjustable parameters. The performance of the three models in classification tasks is consistent with that in regression tasks.[1]

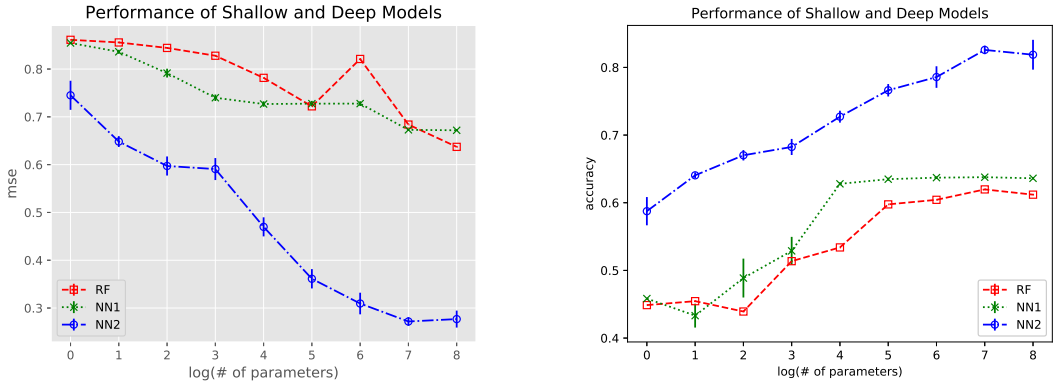


Figure 3.5: Left: mean squared loss of the deep and shallow models in the regression task. Right: classification accuracy of the deep and shallow models in the classification task.

[1] All the code can be downloaded from `https://github.com/syitong/randrelu`

CHAPTER IV

Open Problems and Future Works

This chapter collects the problems that are important for understanding random features method but have not been solved yet, and discusses how the knowledge of random features can help us understand the performance of neural networks better.

4.1 Open Problems for Random Features

How to obtain optimized random features? We have seen that the optimized random feature plays an important role in the approximation error analysis, and experiments show that using reweighted feature selection (Algorithm 1) indeed improves the performance of the algorithm in classification tasks. However, there are two issues to be resolved. First, the reweighted feature selection algorithm can only return approximate optimized random features. We do not understand how the hyper-parameters in reweighted feature selection affect the learning rate of the algorithm if approximate random features are used. Second, since the optimized random feature only depends on the data distribution $\mathbb{P}_{\mathcal{X}}$, in the cases where $\mathbb{P}_{\mathcal{X}}$ is known, for example in a controlled experiment, how to design the random feature is also a valuable question.

Dependence on the dimension of raw features. The fast learning rate of RFSVM in the unrealizable case depends on d, the dimension of $\mathcal{X}$, exponentially. This is caused by the fact that the approximation error between the approximator in the RKHS of the Gaussian kernel and the target Bayes classifier depends on d. Similarly for the random ReLU feature, we also have the approximation error depending on d. Considering that in practice, the dimension of raw data is usually very high, such an exponential dependence on d is not acceptable. Our experiments on high dimensional synthetic data and MNIST seems to confirm the impact of the dimension. We do not know the optimality of the dependence on the dimension yet, and neither how to make use of the fact that many high dimensional data in practice actually have a small intrinsic dimension.

Decay rate of the spectrum of $\Sigma_{\mathrm{ReLU},\tau_d}$. To obtain the learning rate of random features methods, we need to understand the decay rate of the operator induced by the random ReLU feature. Currently, we can only characterize the decay rate in the case where $\mathbb{P}_{\mathcal{X}}$ is the uniform distribution over $\mathbb{S}^d$ by using the spherical harmonics. In this case, the decay rate is polynomial and thus the learning rate is not useful for explaining the practical performance. For some other common data distribution such as the sub-Gaussian or bounded data distribution, we do not know how to characterize the decay rate of the operator yet. If we can prove exponential decay rate for some data distribution, then it is possible to prove the fast learning rate of the random ReLU feature.

4.2 Random Features and Neural Networks

Historically, one important motivation of random features is to obtain a neural network model without solving a non-convex optimization problem. Nowadays, using

stochastic gradient descent to solve non-convex optimization problems seems to be a standard way to train neural networks, and it works surprisingly well in practice. However, we still do not fully understand why this happens. The performance of random features methods provides some lower bound for that of neural networks, because random features methods return partially trained neural networks as solutions. If we further optimize the inner weights of a random feature solution with some norm constraint over inner weights, we can obtain a learning rate guarantee for the resultant neural network.

The recent progress on the achievability of zero training error, and thus the global optimality, of training neural networks via gradient descent provides a different way to understand the connection between random features and neural networks [1, 11]. [1] shows that when an over-parameterized 2-layer neural network is used, 0 training error is achievable with high probablity over the random initialization of weights. And the resultant neural network is not far from the randomly initialized one, so the generalization error is controlled. Then the target function in the RKHS of the random feature corresponding to the neural network can be efficiently learned by running gradient descent on the non-convex optimization problem. Meanwhile, with the over-parameterization assumption, simply optimizing the outer weights with the inners being fixed after random initialization can also result in a random features model with 0 training error and bounded generalization error.

All these comparisons between random features and neural networks do not provide support for the advantage of 2-layer neural networks over the random features models. Recalling the experiment in Section 3.7, it is actually not clear yet whether such an advantage exists. On the other hand, it seems that the deep neural networks indeed have advantage over shallow models based on the evidence from experiments

and the theoretically confirmed depth separation phenomenon. But there is no theory yet on how and when the gradient descent can find the good multi-layer network approximator.

Bibliography

[1] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. *arXiv:1901.08584 [cs, stat]*, January 2019.

[2] D. Azevedo and V. A. Menegatto. Sharp estimates for eigenvalues of integral operators generated by dot product kernels on the sphere. *Journal of Approximation Theory*, 177:57–68, January 2014.

[3] Francis Bach. Breaking the Curse of Dimensionality with Convex Neural Networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017.

[4] Francis Bach. On the Equivalence between Kernel Quadrature Rules and Random Feature Expansions. *Journal of Machine Learning Research*, 18(21):1–38, 2017.

[5] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[6] Luigi Carratino, Alessandro Rudi, and Lorenzo Rosasco. Learning with SGD and Random Features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10192–10203. Curran Associates, Inc., 2018.

[7] Youngmin Cho and Lawrence K. Saul. Kernel Methods for Deep Learning. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 342–350. Curran Associates, Inc., 2009.

[8] Corinna Cortes, Mehryar Mohri, and Ameet Talwalkar. On the impact of kernel approximation on learning accuracy. *Journal of Machine Learning Research*, 9:113–120, 2010.

[9] Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj, Maria-Florina F Balcan, and Le Song. Scalable Kernel Methods via Doubly Stochastic Gradients. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3041–3049. Curran Associates, Inc., 2014.

[10] Dua Dheeru and Efi Karra Taniskidou. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences, 2017.

[11] Simon S. Du, Jason D. Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient Descent Finds Global Minima of Deep Neural Networks. *arXiv e-prints*, page arXiv:1811.03804, November 2018.

[12] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, pages 907–940, 2016.

[13] Z. Harchaoui, F. Bach, and E. Moulines. Testing for Homogeneity with Kernel Fisher Discriminant Analysis. *ArXiv e-prints*, April 2008.

[14] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A Dual Coordinate Descent Method for Large-scale Linear SVM. In

Proceedings of the 25th International Conference on Machine Learning, ICML '08, pages 408–415, Helsinki, Finland, 2008. ACM.

[15] Guang-Bin Huang, Lei Chen, Chee Kheong Siew, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Networks*, 17(4):879–892, 2006.

[16] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: Theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

[17] P. S. Huang, H. Avron, T. N. Sainath, V. Sindhwani, and B. Ramabhadran. Kernel methods match Deep Neural Networks on TIMIT. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 205–209, May 2014.

[18] B. Igelnik and Yoh-Han Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE Transactions on Neural Networks*, 6(6):1320–1329, November 1995.

[19] Vladimir. Koltchinskii, SpringerLink (Online service), and École d'Été de Probabilités de Saint-Flour. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems École d'Été de Probabilités de Saint-Flour XXXVIII-2008*. Lecture Notes in Mathematics,0075-8434 ;2033. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2011.

[20] P.D. Lax. *Functional Analysis*. Pure and Applied Mathematics. Wiley, 2002.

[21] Yann Lecun and Corinna Cortes. *The MNIST Database of Handwritten Digits*.

[22] Holden Lee, Rong Ge, Tengyu Ma, Andrej Risteski, and Sanjeev Arora. On the Ability of Neural Nets to Express Distributions. In Satyen Kale and Ohad

Shamir, editors, *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pages 1271–1296, Amsterdam, Netherlands, July 2017. PMLR.

[23] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861 – 867, 1993.

[24] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015.

[25] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.

[26] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[27] Gilles Pisier. Remarques sur un résultat non publié de B. Maurey. *Séminaire Analyse fonctionnelle (dit "Maurey-Schwartz")*, pages 1–12, 1980.

[28] Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances*

in Neural Information Processing Systems 20, pages 1177–1184. Curran Associates, Inc., 2008.

[29] Ali Rahimi and Benjamin Recht. Weighted Sums of Random Kitchen Sinks: Replacing minimization with randomization in learning. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1313–1320. Curran Associates, Inc., 2009.

[30] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3218–3228, 2017.

[31] Clint Scovel, Don Hush, Ingo Steinwart, and James Theiler. Radial kernels and their reproducing kernel Hilbert spaces. *Journal of Complexity*, 26(6):641–660, 2010.

[32] S. Shalev-Shwartz and T. Zhang. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *ArXiv e-prints*, September 2012.

[33] Bharath Sriperumbudur and Zoltan Szabo. Optimal Rates for Random Fourier Features. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1144–1152. Curran Associates, Inc., 2015.

[34] I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer New York, 2008.

[35] Yitong Sun, Anna Gilbert, and Ambuj Tewari. But How Does It Work in Theory? Linear SVM with Random Features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances*

in Neural Information Processing Systems 31, pages 3383–3392. Curran Associates, Inc., 2018.

[36] Dougal J. Sutherland and Jeff G. Schneider. On the Error of Random Fourier Features. *CoRR*, abs/1506.02785, 2015.

[37] Vladimir Naumovich. Vapnik. *Statistical Learning Theory*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. Wiley, New York, 1998.

[38] Harold Widom. Asymptotic Behavior of the Eigenvalues of Certain Integral Equations. *Transactions of the American Mathematical Society*, 109(2):278–295, 1963.

[39] Tianbao Yang, Yu-feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström Method vs Random Fourier Features: A Theoretical and Empirical Comparison. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 476–484. Curran Associates, Inc., 2012.

[40] Kai Zhang, Liang Lan, Zhuang Wang, and Fabian Moerchen. Scaling up Kernel SVM on Limited Resources: A Low-rank Linearization Approach. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 1425–1434, La Palma, Canary Islands, April 2012. PMLR.