Bias-Variance Tradeoff, Overfitting and the Double Descent Curve

An Honours Thesis

submitted in partial fulfilment of the requirements for the degree of

Bachelor of Science with Honours in Mathematics

presented to

the Department of Mathematics

Faculty of Science

National University of Singapore

Assistant Professor Subhroshekhar Ghosh, Supervisor

by

Ng Wei Le

March 5, 2021

Acknowledgement

Special thanks to my research supervisor, Prof. Subhroshekhar Ghosh, for guiding me in the direction to take the research, and for the time he spent teaching me the concepts.

Abstract

The bias-variance tradeoff is an important concept of classical Machine Learning practice, where the algorithm tries to balance the error from the bias term and variance term, typically by means of controlling the richness of the model to find a sweet spot between underfitting and overfitting. However, in many modern Machine Learning practices, the model is made to highly overfit the data, even up till the point of near interpolation. This is done even under the conditions of large amounts of data and noise.

The papers discussed in this thesis show that this happens not only in deep learning, but also kernel machines with close to 0 training error, and why current generalization bounds do not explain such phenomenon well. Additionally, they propose a possible performance curve to explain how overfitting and overparameterization could be beneficial, and some empirical evidence to support this.

Contents

List of Figures

1. Introduction

1.1 Supervised Learning

Given a set of (training) datapoints $\{(x_1, y_1), ..., (x_n, y_n)\}$ where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$ for $1 \leq i \leq n$, and assuming (x_i, y_i) were independent and identically distributed variables drawn from a probability distribution Q, we want to find a predictor function $h_n : \mathbb{R}^d \to \mathbb{R}$ that predicts y "well" given x that has not been seen.

To represent how well this function predicts, let l represent a loss function. Examples include the squared-loss function $l(y, \hat{y}) = (\hat{y} - y)^2$, and the 0-1 loss, usually used for classification: $l(\hat{y}, y) = 1_{\hat{y} \neq y}$.

The general goal of (supervised) machine learning to find a function h that minimizes the expected loss: $\mathbb{E}_{x,y}[l(h(x), y)]$.

In Empirical Risk Minimization (ERM), given datapoints $\{(x_1, y_1), ..., (x_n, y_n)\}$, the goal is to find a function h_n in some class of function $\mathcal{H}$ to minimize the empirical risk:

$$L_{emp}(h_n) = \frac{1}{n} \sum_{i=1}^{n} l(h_n(x_i), y_i).$$

Intuitively, if a function h_n does not predict training data well (has a high $L_{emp}(h_n)$), this would tell us that it will not predict other (x, y) samples drawn randomly as well.

The empirical risk minimizer, $\hat{h} : \mathbb{R}^d \to \mathbb{R}$ is then defined by:

$$\hat{h_n} = \arg \min_{h_n \in \mathcal{H}} L_{emp}(h_n).$$

Classically, there are reasons why one does not tend to choose a function h_n that reduces the empirical loss to near zero values, typically due to bounds on the generalization gap.

1.2 Bias Variance Tradeoff

We define the generalization error (or sometimes known as the generalization gap) as difference between the empirical and expected classifier loss, i.e. $|\mathbb{E}_{x,y}[l(\hat{h_n}(x), y)] - L_{emp}(\hat{h_n})|$. It would make sense to decrease this gap to as little as possible.

Many classical bounds have this generalization gap is a form of:

$$\mathbb{E}_{x,y}[l(\hat{h_n}(x), y)] \leq L_{emp}(\hat{h_n}) + O(\sqrt{c/n}) \tag{1.1}$$

where c is some measure of the complexity of $\mathcal{H}$, for example the fat-shattering dimension, VC-dimension, Rademacher complexity, etc. The general result being that, with a greater complexity of the function class $\mathcal{H}$, the greater the upper bound of this generalization gap. However, if $\mathcal{H}$ is not complex enough, we may not find any function $\hat{h_n} \in \mathcal{H}$ that reduces the empirical risk sufficiently (underfitting). Hence, classical algorithms try to find a balance between over and underfitting, decreasing the expected loss by controlling $\mathcal{H}$, either explicitly or implicitly. For example, to change the complexity of $\mathcal{H}$ explicitly, one might choose a simpler or more complicated architecture for a neural network. To reduce the complexity of $\mathcal{H}$ implicitly, one might use regularization to penalize coefficients to limit the model complexity, or simply stop the training algorithm prematurely (early stopping). This classical curve from Belkin et al. (2019) is shown in Figure 1.1.

1.3 Modern Machine Learning

In a paper by Zhang et al. (2016), examples were given where deep neural networks were trained to the point of little to no training error. Different architectures were tested on the CIFAR10 dataset (Krizhevsky & Hinton (2009)) and ImageNet dataset (Russakovsky et al. (2015)). However, very accurate predictions on the new data was given.

In Canziani et al. (2016), the architectures used on ImageNet have large amounts of parameters, multiple times bigger than the number of training datapoints.

1.4 Research Question

Indeed, it is still unanswered as to why these overparameterized data do not seem to cause high test loss due to overfitting. The papers discussed further show empirically that this property is not exclusive to deep learning, but also seems to appear in learning for kernel machines as well. This will be followed by a plausible explanation on how the classical bias-variance tradeoff graph can be reconciled with modern methods.

We first give a brief introduction on kernels and Reproducing Kernel Hilbert Spaces, which will then later be used in further sections.

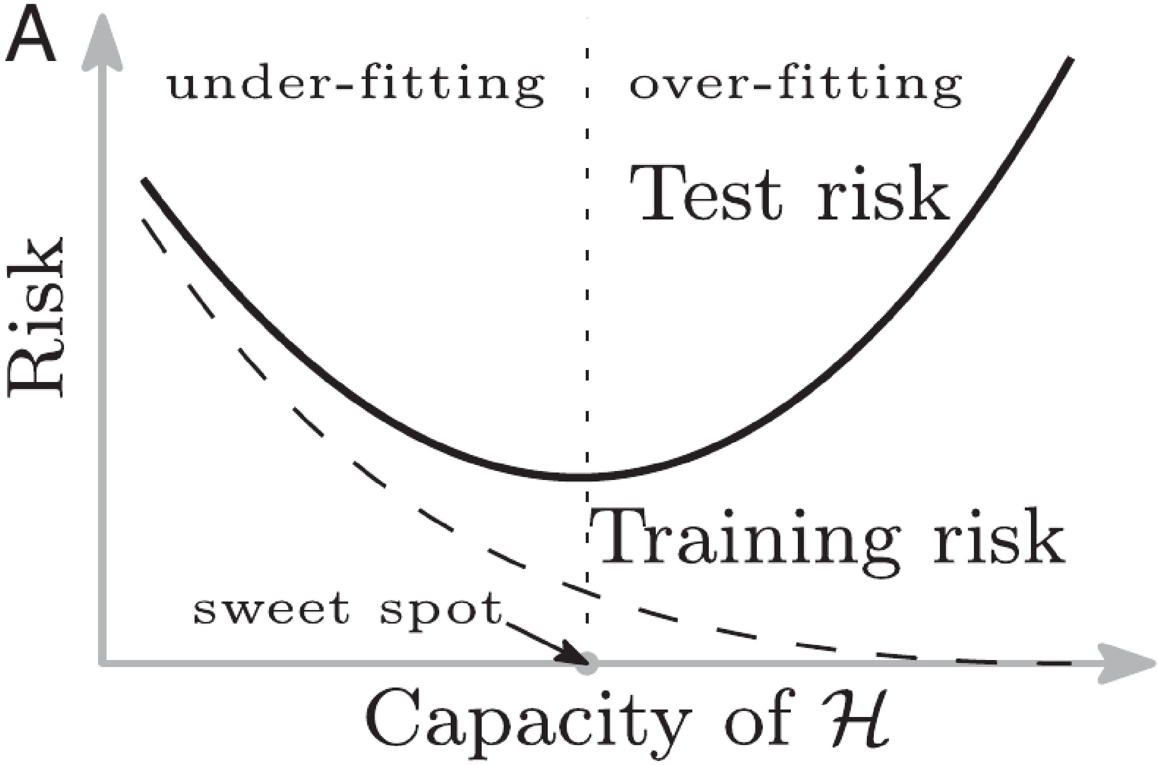


Figure 1.1: Classical U-shaped curve showing how the training and test risk changes with respect to the capacity of $\mathcal{H}$. The test risk results from bias-variance tradeoff, and the Capacity of $\mathcal{H}$ is selected at the sweet spot.

2. Kernels

2.1 Notation

We use the symbol $\mathbb{K}$ when it can refer to both $\mathbb{R}$ or $\mathbb{C}$. Also, let z^* or $(z)^*$ denote the conjugate of z for any $z \in \mathbb{C}$. The sections covering Kernels and reproducing kernel Hilbert spaces are mainly referenced using Steinwart & Christmann (2008).

2.2 Definition and Properties

Definition 1. For a non-empty set X, let $k : X \times X \to \mathbb{K}$ be known as a kernel if there exists a function $\phi : X \to \mathcal{H}$ (known as a feature map of k) where $\mathcal{H}$ is a $\mathbb{K}$-Hilbert space (known as a feature space of k) such that

$$k(x_1, x_2) = \langle \phi(x_2), \phi(x_1) \rangle_{\mathcal{H}}. \tag{2.1}$$

Lemma 1. *For any kernel k on X, $k(x_1, x_2) = k(x_2, x_1)^*$.*

From the properties of the inner product, we know that $k(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle^* = k(x_2, x_1)^*$. Therefore, for kernels on $\mathbb{R}$, the symmetric property: $k(x_1, x_2) = k(x_2, x_1)$ holds.

Lemma 2. *Let k_1, k_2 be kernels on a non-empty set X. Then $k_1 + k_2$ and $ak_1, a \in \mathbb{R}^+ \cup \{0\}$ are kernels.*

Below, we define the Gaussian RBF kernel:

Definition 2. Let the complex Gaussian RBF kernel (on $\mathbb{C}^d$)be defined as:

$$k_{\gamma,\mathbb{C}^d}(z, z') := e^{-\gamma^{-2} \sum_{i=1}^{d} (z_i - z_i'^*)^2}$$

We then define the real Gaussian RBF kernel (or simply the Gaussian RBF kernel for short) acting on $\mathbb{R}^d$ as:

$$k_\gamma(x, x') = e^{-\gamma^{-2} \|x - x'\|_2^2}.$$

Definition 3. The Laplacian (or exponential) acting on $\mathbb{R}^d$ is defined as:

$$k_\gamma(x,x') = e^{-\gamma^{-1}\|x-x'\|_2}$$

It can be shown (Steinwart & Christmann (2008)) that the complex and real Gaussian RBF kernels and the Laplacian kernel are kernels.

Definition 4. For a non-empty set X, a function $k : X \times X \to \mathbb{R}$ is said to be a positive definite if, for any $m \in \mathbb{Z}^+ \cup \{0\}$ and for all $x_1, ..., x_n \in X$, we have the following matrix (called the Gram matrix) being positive semi-definite:

$$K := (k(x_i, x_j))_{i,j}.$$

Equivalently: for all $a_1, ..., a_n \in \mathbb{R}$, we have:

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

Definition 5. The positive definite function $k : X \times X \to \mathbb{R}$ is said to be symmetric if $k(x_1, x_2) = k(x_2, x_1)$ for all $x_1, x_2 \in X$

Theorem 1. *A real function $k : X \times X \to \mathbb{R}$ is a kernel if and only if k is a positive definite symmetric function (also known as a positive definite kernel).*

Proof. Suppose k is a kernel. Then there exists some feature map $\Phi : X \to \mathcal{H}$.

$$\begin{aligned}\sum_{j=1}^{n}\sum_{i=1}^{n} a_j a_i k(x_j, x_i) &= \sum_{j=1}^{n}\sum_{i=1}^{n} a_j a_i \langle \phi(x_i), \phi(x_j) \rangle_{\mathcal{H}} \\ &= \langle \sum_{i=1}^{n} a_i \phi(x_i), \sum_{j=1}^{n} a_j \phi(x_j) \rangle_{\mathcal{H}} \\ &= \| \sum_{i=1}^{n} a_i \phi(x_i) \| \\ &\geq 0.\end{aligned}$$

Also, from Lemma 1, we know that the real kernel k is symmetric, proving one side of the theorem. To prove the other side:

Given $k : X \times X \to \mathbb{R}$ a positive definite symmetric function, we shall prove that $\Phi : X \to H$

with mapping $x \mapsto k(\cdot, x)$ is a valid feature map for some feature space H. First, we define

$$\hat{\mathcal{H}} := \sum_{i=1}^{n} a_i k(\cdot, x_i), n \in \mathbb{Z}^+ \cup \{0\}, a_i \in \mathbb{R} \text{ for all } i, x_i \in X \text{ for all } i.$$

For $f, g \in \hat{\mathcal{H}}$ where $f = \sum_{i=1}^{n} a_i k(\cdot, x_i)$ and $g = \sum_{j=1}^{m} b_j k(\cdot, y_j)$, we define the inner product as such:

$$\begin{aligned} \langle f, g \rangle &:= \sum_{i=1}^{n} \sum_{j=1}^{m} a_i b_j k(y_j, x_i) \\ &= \sum_{j=1}^{m} b_j f(y_j) \\ &= \sum_{i=1}^{n} a_i g(x_i) \end{aligned} \tag{2.2}$$

This definition is bilinear and symmetric.

We also have: $\langle f, f \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j k(x_j, x_i) \geq 0$ since k is a positive definite function. It can be shown that $\langle \cdot, \cdot \rangle$ follows Cauhy-Schwarz Inequality (Steinwart & Christmann (2008)), hence we have:

$$\begin{aligned} |f(x)|^2 &= |\sum_{i=1}^{n} a_i k(x, x_i)|^2 \\ &= |\langle f, k(\cdot, x) \rangle|^2 \ (\because (2.2) \text{ with } g = \sum_{j=1}^{m} b_j k(\cdot, y_j) = k(\cdot, x) \text{ with } m = 1, b_1 = 1, y_1 = x) \\ &\leq \langle k(\cdot, x), k(\cdot, x) \rangle \ \langle f, f \rangle. \end{aligned}$$

Therefore, if $\langle f, f \rangle = 0$, then $f = 0$, hence showing that $\langle f, f \rangle > 0$ if and only if $f \neq 0$. Hence, $\langle \cdot, \cdot \rangle$ defines a proper inner product in $\hat{\mathcal{H}}$.

Let $\mathcal{H}$ be the completion of $\hat{\mathcal{H}}$ and the map $U : \hat{\mathcal{H}} \to \mathcal{H}$ be the map where $\langle Ux, Uy \rangle_{\mathcal{H}} = \langle x, y \rangle_{\hat{\mathcal{H}}}$ for all $x, y \in \hat{\mathcal{H}}$. Then we have, for all $x, x' \in X$:

$$k(x, x') = \langle k(\cdot, x'), k(\cdot, x) \rangle_{\hat{\mathcal{H}}} = \langle Uk(\cdot, x'), Uk(\cdot, x) \rangle_{\mathcal{H}}.$$

Thus we find a feature map of k, proving that k is a kernel. □

2.3 Reproducing Kernel Hilbert Spaces

Initially introduced by Stanislaw Zaremba, reproducing kernel Hilbert spaces have many applications in the fields such as Statistical Learning and complex analysis. An RKHS is a $\mathbb{K}$-Hilbert function space where point evaluation is continuous linear funcitonal.

Definition 6. (RKHS). Let $\mathcal{H}$ be a $\mathbb{K}$-Hilbert space of functions over a non-empty set X. $\mathcal{H}$ is called an RKHS over X if the Dirac function $\delta_x : \mathcal{H} \to \mathbb{K}$ defined as:

$$\delta_x(f) := f(x),\ x \in X,\ f \in \mathcal{H}$$

is continuous. Equivalently, there exists $0 < M_x < \infty$ such that

$$\delta_x(f) \leq M_x \|f\|_{\mathcal{H}},\ \text{for all} f \in \mathcal{H}.$$

δ_x is called a bounded operator on $\mathcal{H}$.

This is not easy to put into practice, hence the reproducing kernel is defined.

Definition 7. (Reproducing Kernel). For a non-empty set X and a function $k : X \times X \to \mathbb{K}$ k is called a reproducing kernel of $\mathcal{H}$ if $k(\cdot, x) \in \mathcal{H}$ for all $x \in X$ and the following property hold for all $x \in X$ and $f \in \mathcal{H}$:

$$f(x) = \langle f, k(\cdot, x) \rangle \tag{2.3}$$

The condition in equation (2.3) is also known as the reproducing property.

Definition 8. (Canonical Feature Maps). Let $\mathcal{H}$ be an RKHS over X with reproducing kernel k. Let the function $\Phi : X \to \mathcal{H}$ be defined such that for all $x \in X$,

$$\Phi(x) = k(\cdot, x).$$

We call Φ the canonical feature map of k.

Lemma 3. ***(A reproducing kernel of an RKHS is a kernel).*** *Let $\mathcal{H}$ be an RKHS over X with reproducing kernel k. Then k is a kernel.*

Proof. We simply proof that Φ is a feature map of k.

$$\begin{aligned}\langle \Phi(x_2), \Phi(x_1) \rangle &= \langle k(\cdot, x_2), k(\cdot, x_1) \rangle \\ &= k(x_1, x_2) \ (\because \text{Reproducing Property (2.3)})\end{aligned}$$

So $\mathcal{H}$ is also a feature space of k. □

Lemma 4. *Let $\mathcal{H}$ be an $\mathbb{K}$-Hilbert functional RKHS over X with reproducing kernel k. Then H is a Reproducing Kernel Hilbert Space.*

Proof. Recall the Dirac functional $\delta_x : H \to \mathbb{K}$ where:

$$\delta_x(f) = f(x),\ x \in X,\ f \in H.$$

Then we have:

$$\begin{aligned}|\delta_x(f)| &= |f(x)| \\ &= |\langle f, k(\cdot, x) \rangle| \ (\because \text{Reproducing Property (2.3)}) \\ &\leq \|k(\cdot, x)\|_{\mathcal{H}} \|f\|_{\mathcal{H}} \ (\because \text{Cauchy-Schwarz Inequality})\end{aligned}$$

This shows that the Dirac functionals are continuous. □

2.3.1 Representer Theorem

Representer Theorem ensures that the argmin of an empirical risk expression involving a function over an RKHS can be expressed as a linear combination of kernels applied on the training data points as proven in Schölkopf et al. (2001).

Theorem 2. ***(Representer Theorem).*** *Given a non-empty set X, training data $(x_1, y_1), ...(x_n, y_n) \in X \times \mathbb{R}$, and RKHS $\mathcal{H}$ a $\mathbb{R}$-Hilbert function space over X with reproducing kernel $k : X \times X \to \mathbb{R}$. Let g be a strictly increasing function $g : [0, \infty] \to \mathbb{R}$, and $l : (X \times \mathbb{R}^2)^n \to \mathbb{R} \cup \{\infty\}$ be an arbitrary loss function.*
We want to minimize the following empirical risk term:

$$L_{emp}(f, (x_1, y_1), ..., (x_n, y_n)) \ := \ l((x_1, y_1, f(x_1)), ..., (x_n, y_n, f(x_n))) + g(\|f\|).$$

For $\hat{f} = \arg\min_{f \in \mathcal{H}} L_{emp}(f, (x_1, y_1), ..., (x_n, y_n))$, $\hat{f}$ *can be represented in the form:*

$$\hat{f}(\cdot) = \sum_{i=1}^{n} a_i k(\cdot, x_i)$$

with $a_i \in \mathbb{R}$ *for* $1 \leq i \leq n$.

Proof. First we let $\Phi : X \to \mathcal{H}$ be the canonical feature map of k as defined in (8). Recall that $\Phi(x)(\cdot) = k(\cdot, x)$. Due to the reproducing property where $\Phi(x)(x') = \langle \Phi(x), k(\cdot, x') \rangle$, we have:

$$\begin{aligned} \Phi(x)(x') &= k(x', x) \\ &= \langle \Phi(x), k(\cdot, x') \rangle \\ &= \langle \Phi(x), \Phi(x') \rangle. \end{aligned}$$

So Φ is a feature space of k. Using orthogonal decomposition, we decompose $f \in \mathcal{H}$ into a component projected onto the span of $\Phi(x_1), ..., \Phi(x_n)$, and the other component orthogonal to this span. We will then prove this orthogonal component is 0 for any f that reduces the empirical risk term, hence completing the proof.

$$f = \sum_{i=1}^{n} a_i \Phi(x_i) + \gamma,$$

where $\gamma \in \mathcal{H}$, $\langle \Phi(x_i), \gamma \rangle = 0$ for all $1 \leq i \leq n$.

Next, applying the reproducing property again,

$$\begin{aligned} f(x_j) &= \langle f, k(\cdot, x_j) \rangle \\ &= \langle \sum_{i=1}^{n} a_i \Phi(x_i) + \gamma, \Phi(x_j) \rangle \\ &= \langle \sum_{i=1}^{n} a_i \Phi(x_i), \Phi(x_j) \rangle + \langle \gamma, \Phi(x_j) \rangle \\ &= \sum_{i=1}^{n} a_i \langle \Phi(x_i), \Phi(x_j) \rangle. \end{aligned}$$

Now, consider:

$$\begin{aligned}
\|f\|^2 &= \|\sum_{i=1}^{n} a_i \Phi(x_i) + \gamma\|^2 \ (\because \text{orthogonality}) \\
&= \|\sum_{i=1}^{n} a_i \Phi(x_i)\|^2 + \|\gamma\|^2 \\
&\geq \|\sum_{i=1}^{n} a_i \Phi(x_i)\|^2 \\
\Longrightarrow g(\|f\|) &\geq g(\|\sum_{i=1}^{n} a_i \Phi(x_i)\|)
\end{aligned}$$

Therefore, if we have $\gamma = 0$, since $f(x_i)$ is unaffected by this for all $1 \leq i \leq n$, and $l((x_1, y_1, f(x_1)), ..., (x_n, y_n, f(x_n)))$ is also unaffected by γ. For the term $g(\|f\|)$, it decreases if we have $\gamma = 0$. Hence, $\hat{f} = \arg\min_{f \in \mathcal{H}} L_{emp}(f, (x_1, y_1), ..., (x_n, y_n))$, $\hat{f}$ must have $\gamma = 0$, and

$$\begin{aligned}
\hat{f} &= \sum_{i=1}^{n} a_i \Phi(x_i) + 0 \\
&= \sum_{i=1}^{n} a_i k(\cdot, x_i)
\end{aligned}$$

□

3. Overfitted and Interpolated Kernel Classifiers

We shall give experimental results performed in Belkin et al. (2018b) that show the strong generalization performance also appears in kernel classifiers.

3.1 Rationale

The aim is to have interpolated solutions which fits the data perfectly, thus having no regularization. A form of inductive bias is to be introduced to choose the solution with some special properties. Though no finite number of functions in the RKHS are able to fit the training data, minimum norm RKHS solutions can be obtained using Representer Theorem. Some rationale behind choosing the minimum norm solution as the inductive bias is introduced in section 5.3.

3.2 Training

We have a set of training datapoints $(x_1.y_1), ..., (x_n, y_n) \in \mathbb{R}^d \times \mathbb{R}$. Let $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a kernel. Let $\mathcal{H}$ denote the RKHS corresponding to the kernel k. Let f^* denote the minimum norm interpolant given the datapoints, i.e.

$$f^* = \arg \min_{f \in \mathcal{H}, f(x_i) = y_i, 1 \le i \le n} (\|f\|_{\mathcal{H}}).$$

We know that by representer theorem (2),f^* can be represented in form of:

$$f^*(\cdot) = \sum_{i=1}^{n} a_i^* k(x_i, \cdot).$$

Due to its interpolating properties, we know that for $a^* := (a_1^*, ..., a_n^*)^{\mathrm{T}}$,

$$a^* = K^{-1}(y_1, ..., y_n)^{\mathrm{T}} \tag{3.1}$$

where $K \in \mathbb{R}^{d,d}$ is the matrix where $K_{i,j} = k(x_i, x_j)$. We also know that:

$$\|f^*\|_{\mathcal{H}} = \sum_{i,j=1}^{n} a_i^* a_j^* k(x_i, x_j) = a^{*\mathrm{T}} K a^*.$$

Due to the time complexity required to solve a^*, iterative methods were used in the paper, where a^* is solved by an unconstrained minimization problem:

$$a^* = \arg\min_a \sum_i l((\sum_j a_i k(x_j, x_i)), y_i)$$

for loss function l.

The Gaussian RBF kernel and Laplacian RBF kernels were used for k.

3.3 Results

The models were trained till mean square loss on the training dataset approaches zero. It is noted that the benefits of early stopping regularization were little to none in terms of test regression or classification error.

Regardless of using iterative or direct (3.1) methods, the interpolated solution performed close to optimal, as seen in the Figure 3.1.

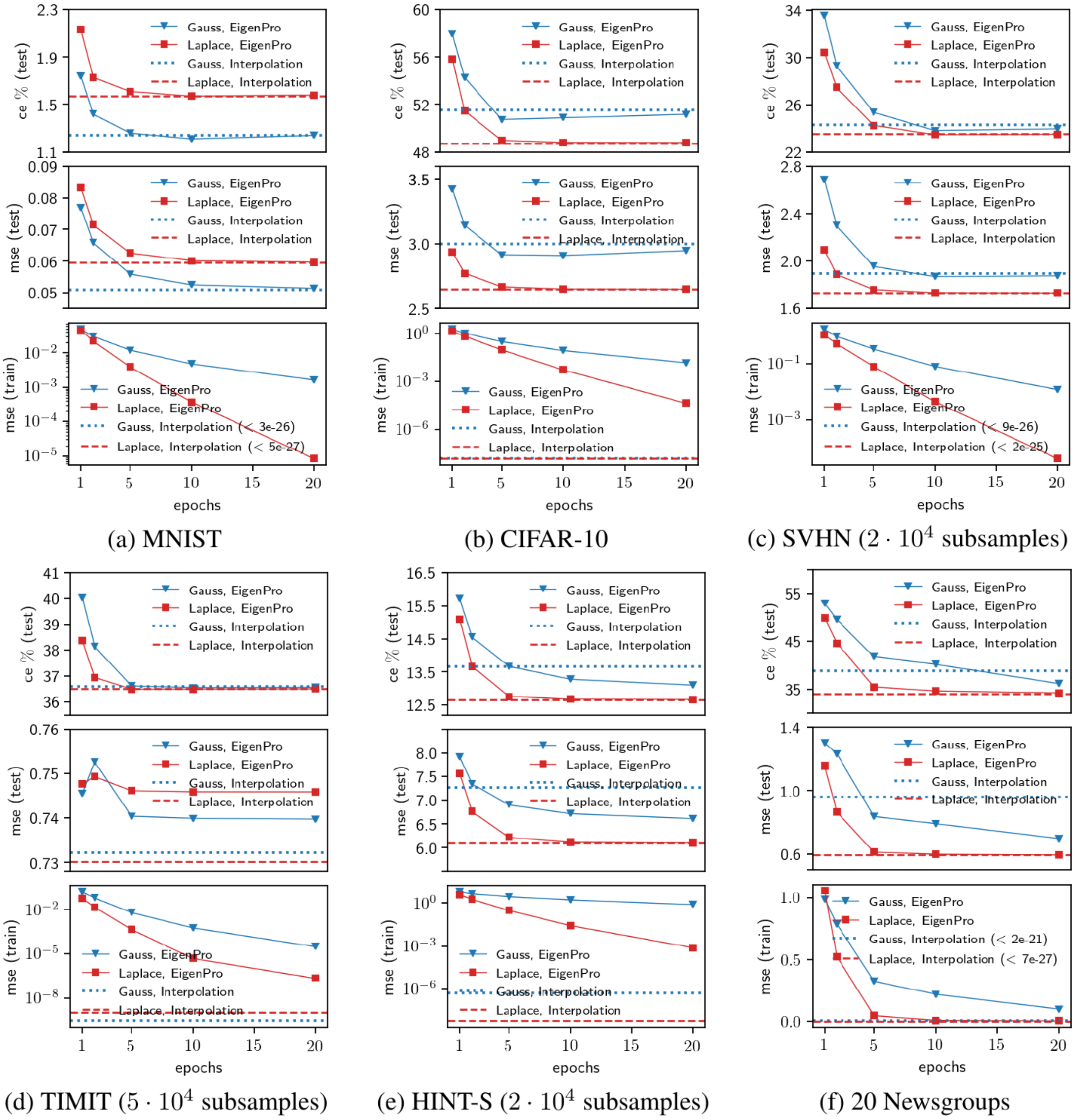
This good performance is similar to deep neural networks that interpolate the training data, as seen in sources such as Zhang et al. (2016).

3.4 Existing Bounds for Interpolated Kernel Classifiers

We will discuss how the norm of the classifiers change with respect to data size when overfitting. Note that in this paper, classifiers are called **overfitted** if the classification loss for the training set is 0 or near 0, and classifiers are called **interpolated** when the mean square error is 0 or near 0. Therefore, an interpolated classifier is overfitted, but not necessarily the other way round.

Let $(x_1, y_1), ..., (x_n, y_n) \in \Omega \times \{-1, 1\}$ be the training dataset, and $\Omega \subset \mathbb{R}^d$ be bounded. The datapoints are chosen from some probability measure P on $\Omega \times \{-1, 1\}$, and the noise of the function y on x is nonzero.

Definition 9. For a function $h \in \mathcal{H}$, and $t \in \mathbb{R}^+$, it **t-overfits** the data if it is overfitted



(a) MNIST

(b) CIFAR-10

(c) SVHN ($2 \cdot 10^4$ subsamples)

(d) TIMIT ($5 \cdot 10^4$ subsamples)

(e) HINT-S ($2 \cdot 10^4$ subsamples)

(f) 20 Newsgroups

Figure 3.1: "ce" refers to classification error, and mse refers to the mean square loss. All methods resulted in 0% classification error on the training set. All datasets were subsampled to reduce complexity required to train.

with 0 classification loss, and $0 < t < y_i h(x_i)$ for $1 \leq i \leq n$.

This condition is still weaker than interpolation, which requires $h(x_i) = y_i$, but still stronger than overfitting. The rationale for introducing t-overfitting is that any overfitted classifier can be scaled by a factor of $1/\gamma$ to reduce the norm by a factor of γ, so any overfitted classifier may have an arbitrarily small norm. Hence, this additional constraint has to be added.

Definition 10. **(γ-shattering and the fat-shattering dimension)**. Let F be a set of functions of $f : \Omega \to \mathbb{R}$ and $S = \{x_1, x_2, ..., x_n\} \subseteq \Omega$, and $\gamma \in \mathbb{R}^+$. Then we say S is γ-shattered by F if there exists $r_1, r_2, ..., r_n \in \mathbb{R}$ such that for each $b \in \{0,1\}^n$ there is a function $f_b \in F$ such that

$$f_b(x_i) \geq r_i + \gamma \text{ if } b_i = 1,$$
$$f_b(x_i) \leq r_i - \gamma \text{ if } b_i = 0, \text{ for } 1 \leq i \leq n.$$

We say $r = (r_1, r_2, ..., r_n)$ witnesses the shattering. Also, we say that F has γ-dimension D if D is the maximum cardinality of a subset S of Ω that is γ-shattered by F. If no such maximum exists, we say that F has infinite γ-dimension. The γ-dimension of F is denoted $fat_F(\gamma)$. This defines a function $fat_F : \mathbb{R} \to \mathbb{N} \cup \{0, \infty\}$, which we call the fat-shattering dimension of F.

Belkin et al. (2018b) suggests a theorem on how the norm of the function in the RKHS changes with respect to t-overfitting.

Theorem 3. *Let $(x_1, y_1), ..., (x_n, x_n) \in \Omega \times \{-1, 1\}$ be data sampled from probability distribution P, and y is not a deterministic function of x. With high probability, for any h that t-overfits the data, there exists some constants $A, B > 0$ depending on t, that satisfies*

$$\|h\|_{\mathcal{H}} > Ae^{Bn^{1/d}}.$$

Proof. Let $L(f) := \mathbb{E}_P[l(f(x), y)]$, the expected classification error. Let $B_R = \{f \in \mathcal{H},\ \|f\|_{\mathcal{H}} < R\}$ be a ball of radius R in RKHS $\mathcal{H}$. Suppose the data is γ-overfitted. Kégl et al. (2001) gives us a high probability of a bound of

$$L(f) < O(\frac{\ln(n)^2}{\sqrt{n}} \sqrt{fat_{B_R}(\gamma/8)}).$$

Thus, we have:

$$fat_{B_R}(\gamma) \gtrapprox O(n\ L(f)^2).$$

Also, from Belkin (2018) we have

$$fat_{B_R}(\gamma) < O((\log(R/\gamma))^d).$$

We then have B_R containing no function that γ overfits the data unless

$$\begin{aligned} (\log(R/\gamma))^d &> O(n) \\ \Longrightarrow log(R/\gamma) &> O(n^{1/d}) \\ \Longrightarrow R &> c_1 \exp(c_2 n^{1/d}) \end{aligned}$$

for some positive constants c_1, c_2. Therefore, there exists some constants $A, B \in \mathbb{R}^+$ such that $\|h\|_{\mathcal{H}} > Ae^{Bn^{1/d}}$. □

Classical bounds for kernel methods (Belkin et al. (2018b), Steinwart & Christmann (2008), Rudi et al. (2015)) are in the form:

$$|\frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i) - L(f)| \leq C_1 \frac{\|f\|_{\mathcal{H}}^a}{n^b} + C_2, \quad C_1, C_2, a, b \geq 0$$

The right side on this will tend to infinity for bigger $\|f\|_{\mathcal{H}}$, which is suggested by Theorem 3, thus the bound is trivial.

This shows that current bounds are insufficient to explain performance on interpolated kernel classifiers.

4. Double Descent

4.1 Double Descent Curve

As proposed in Belkin et al. (2019), a way to reconcile the classical Machine Learning curve with modern machine overparemeterization practice is the double descent curve. As seen in the Figure 4.1, in the first part of the curve, when the capacity of $\mathcal{H}$ (the number of free parameters in the model in this case) is smaller, it resembles the classical U-shaped curve. However, when the capacity of $\mathcal{H}$ increases further, the test risk hits a maximum (typically at the point of interpolation), and further increase of capacity of $\mathcal{H}$ decreases the test risk, sometimes to the point of the graph going below the minimum of the initial U-shaped curve itself.

This curve is observed empirically in several significant models such as neural networks and many different datasets. The paper by Belkin provides several examples, including small Neural Networks and Random Forests, etc., and we will focus on one of them, in Section 4.3. Some other experimental demonstrations of the double descent curve are described in the Appendix A, for random forests (A.2), and with artificial data and noise (A.1).

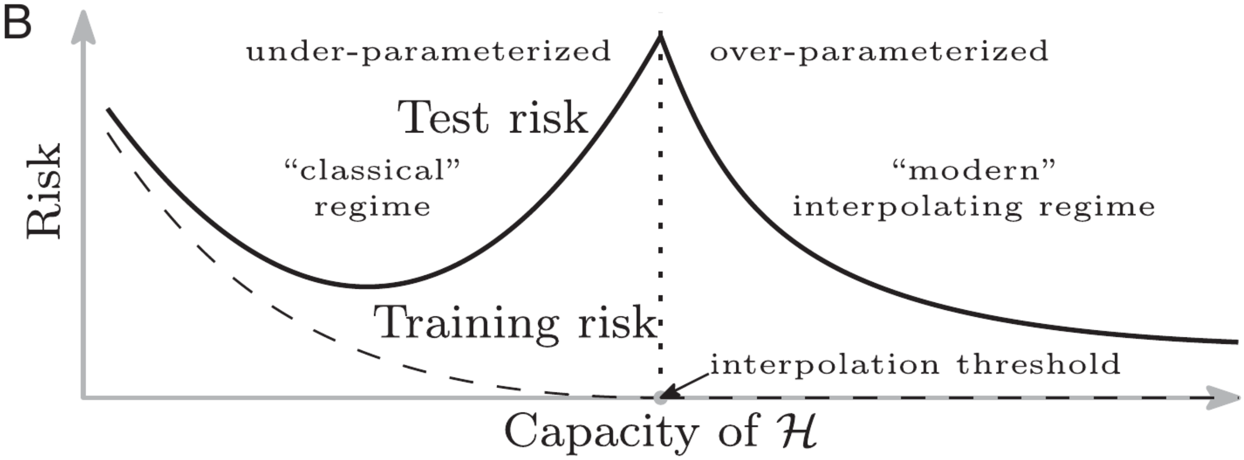


Figure 4.1: Curve showing the double descent proposition. When the capacity of $\mathcal{H}$ is large enough the test risk may be lower than that at the local minimum.

4.2 Random Fourier Features

For a feature map $\phi : \mathbb{R}^d \to \mathbb{R}^{d'}$, the kernel trick allows easy computation for positive definite kernel k where $k(x, y) = \langle \phi(x), \phi(y) \rangle$. We want to find a randomized feature map $z : \mathbb{R}^d \to \mathbb{R}^D$ such that

$$\begin{aligned} k(x, y) &= \langle \phi(x), \phi(y) \rangle \\ &\approx \langle z^{\mathrm{T}}(x), z(y) \rangle. \end{aligned}$$

As suggested by Rahimi & Recht (2008), for a shift-invariant kernel k (where $k(x, y) = k(x - y)$), we consider the mapping $z(x) = cos(w^{\mathrm{T}}x + b)$, where w is drawn from the probability distribution p:

$$p(w) = \frac{1}{2\pi} \int k(h) \ \exp(-iw^{\mathrm{T}}h) \ \mathrm{d}h \tag{4.1}$$

when we compute the Fourier transform of the (properly scaled) kernel k, and b is drawn from the uniform distribution on $[0, 2\pi]$.

We know that the fourier transform of $k(\cdot)$ is a probability distribution from Bochner's theorem:

Theorem 4. *(Bochner Rudin (1990)).For a continuous kernel $k(x - y)$ it is a positive definite kernel if and only if $k(\cdot)$ is the fourier transform of a non-negative measure.*

As will be used in the next section, note that: For $k(\Delta) = \exp(-\|\Delta\|_2^2/2)$, $p(w) = (2\pi)^{-d/2}\exp(-\|w\|_2^2/2)$.

We now have:

$$k(x - y) = \int_{\mathbb{R}^d} p(w)\mathrm{exp}(iw^{\mathrm{T}}(x - y)) \ \mathrm{d}w = \mathbb{E}_w[e^{iw^{\mathrm{T}}x}(e^{iw^{\mathrm{T}}y})^*].$$

Therefore, we can use $e^{iw^{\mathrm{T}}x}(e^{iw^{\mathrm{T}}y})^*$ as an (unbiased) estimate of $k(x, y)$. Let $\phi_w(x) = e^{iw^{\mathrm{T}}x}$ We can also use $z_w(x) = \sqrt{2}cos(w^{\mathrm{T}}x + b)$ instead of $\phi_w(x)$, as suggested by Rahimi & Recht (2008).

Proposition 1. *For $z_w(x) = \sqrt{2}cos(w^T x + b)$, where w is drawn from probability distribution* *p in* (4.1) *and* b *drawn from a uniform random variable on* $[0, 2\pi]$,

$$\mathbb{E}_w[z_w(x)z_w(y)] = k(x, y)$$

Proof.

$$\begin{aligned} z_w(x) &= 2\ \frac{\sqrt{2}}{2} cos(w^{\mathrm{T}}x + b) \\ &= \frac{1}{\sqrt{2}}\ (e^{i(w^{\mathrm{T}}x+b)} + e^{-i(w^{\mathrm{T}}x+b)}) \\ &= \frac{1}{\sqrt{2}}\ (\phi_w(x)e^{ib} + \phi_w(x)^* e^{-ib}) \end{aligned}$$

where $\phi_w(x) = e^{iw^{\mathrm{T}}x}$, and z^* is the complex conjugate of z.

$$\begin{aligned} z_w(x)z_y(y) &= \frac{1}{2}[\phi_w(x)\phi_w(y)e^{i2b} + \phi_w(x)^*\phi_w(y)^* e^{-i2b} \\ &\quad + \phi_w(x)\phi_w(y)^* + \phi_w(x)^*\phi_w(y)] \\ \mathbb{E}[z_w(x)z_y(y)] &= \frac{1}{2}\mathbb{E}[\phi_w(x)\phi_w(y)e^{i2b} + \phi_w(x)^*\phi_w(y)^* e^{-i2b}] \\ &\quad + \frac{1}{2}\mathbb{E}[\phi_w(x)\phi_w(y)^*] + \frac{1}{2}\mathbb{E}[\phi_w(x)^*\phi_w(y)] \end{aligned}$$

As mentioned earlier in Theorem 4, $\mathbb{E}_w[\phi_w(x)\phi_w(y)^*] = k(x - y)$. Also $\phi_w(x)\phi_w(y)^* = (\phi_w(x)^*\phi_w(y))^*$.

$$\begin{aligned} \mathbb{E}[z_w(x)z_y(y)] &= \frac{1}{2}\mathbb{E}[\phi_w(x)\phi_w(y)e^{i2b} + \phi_w(x)^*\phi_w(y)^* e^{-i2b}] + \frac{1}{2}k(x - y) + \frac{1}{2}[k(x - y)]^* \\ &= \frac{1}{2}\mathbb{E}[\phi_w(x)\phi_w(y)e^{i2b} + \phi_w(x)^*\phi_w(y)^* e^{-i2b}] + k(x - y) \end{aligned}$$

Since k is a real kernel, $k(x - y) = (k(x - y))^*$.

$$\begin{aligned} \mathbb{E}_{w,b}[\phi_w(x)\phi_w(y)e^{i2b}] &= \frac{1}{2\pi}\int_{\mathbb{R}^d}\int_0^{2\pi} p(w)\phi_w(x)\phi_w(y)e^{i2b}\mathrm{d}b\ \mathrm{d}w \\ &= \frac{1}{2\pi}\int_{\mathbb{R}^d} p(w)\phi_w(x)\phi_w(y)\int_0^{2\pi} e^{i2b}\mathrm{d}b\ \mathrm{d}w \\ &= 0 \end{aligned}$$

Since $\int_0^{2\pi} e^{i2b}\mathrm{d}b = 0$. Similarly, $\mathbb{E}_{w,b}[\phi_w(x)^*\phi_w(y)^* e^{-i2b}] = 0$.

$$\therefore \mathbb{E}[z_w(x)z_y(y)] = k(x - y).$$

□

The variance of the estimate is decreased by using z, a D dimensional vector by concatenating D of z_w and normalizing by a constant $\sqrt{D}$. We let:

$$z(x) = \sqrt{\frac{2}{D}}[cos(w_1^{\mathrm{T}}x + b_1)...cos(w_D^{\mathrm{T}}x + b_D)]$$

with randomly drawn w_i and b_i as described previously.

4.3 Experiment

4.3.1 Function Class

Assume we have datapoints $\{(x_j, y_j)\}_{1\leq j\leq n}$, where $x_j \in \mathbb{R}^d$, $y_j \in \mathbb{R}$.
Let $\xi(x; w) := \exp(\sqrt{-1}\ w^{\mathrm{T}}x)$. Consider the class of functions, $\mathcal{H}_N$ with N number of complex free parameters. That consist of $h_N : \mathbb{R} \to \mathbb{C}$ where:

$$h_N(x) = \sum_{j=1}^{N} a_j \xi(x; w_j).$$

Where $w_1, ..., w_N \in \mathbb{R}^d$ are sampled independently from the standard Gaussian distribution. With this, $\mathcal{H}_N$ becomes a better approximation of the RKHS corresponding to the Gaussian RBF kernel when N is huge. We denote the RKHS corresponding ot the Gaussian RBF kernel with H_∞. As per standard machine learning procedure, we want to find $a_1, ..., a_N$ so that the function $h_N \in \mathcal{H}_N$ is predicts y given unseen x well.

4.3.2 Algorithm

Choose $a = (a_1, ..., a_N)$ only through minimization of the square loss, i.e.

$$\min_a \frac{1}{n} \sum_{k=1}^{n} (h_N(x_k) - y_k)^2.$$

However, when the minimizer is not unique (which will be the case when $N > n$), choose a such that it has the smallest L2-norm out of all the minimizers.

This choice of norm for a is meant to be similar to the RKS norm $\|h\|_{\mathcal{H}_\infty}$. Below, we give some intuition to the reasoning:

Let $f(x)$ be the minimum norm RKHS interpolant function for the datapoints.

$$\begin{aligned} f(x) &= \sum_i \beta_i k(x_i, x) \because \text{(Representer Theorem)} \\ &\approx \sum_i \beta_i z(x_i)^{\mathrm{T}} z(x) \\ &= a^{\mathrm{T}} z(x) = \hat{f}(x) \end{aligned}$$

Where $a = \sum_i \beta_i z(x_i)$. The norm of the function from the random fourier features approximation is:

$$\begin{aligned} \|a\| = a^{\mathrm{T}} a &= (\sum_i \beta_i z^{\mathrm{T}}(x_i))(\sum_i \beta_i z(x_i)) \\ &= \sum_i \sum_j \beta_i \beta_j z^{\mathrm{T}}(x_i) z(x_j) \\ &\approx \sum_i \sum_j \beta \beta_j k(x_i, x_j) \\ &= \|f\|_{\mathcal{H}_\infty} \end{aligned}$$

4.3.3 Results

This was tested on the MNIST dataset, with more information in the original paper.

The initial part of the curve as $N < n$ follows a standard U-shaped descent curve, However, after the point of interpolation when there is 0 training risk (when $N = n$), the test loss decreases, even dipping below the previous local minimum as N grows larger (around $N = 2n$). It keeps decreasing as N increases further, moving towards the original kernel solution as seen in Figure 4.2.

4.4 Plausible Explanations

Why does the test risk decrease even when increasing N after the point of interpolation?

If we look at equation (1.1), it seems counter-intuitive to increase the complexity of the function class after the point of interpolation. After all, the L_{emp} remains at 0, while the complexity of $\mathcal{H}$ increases, thus increasing the upper bound of the generalization gap. It seems that this upper bound is insufficient in explaining the decrease in test risk.

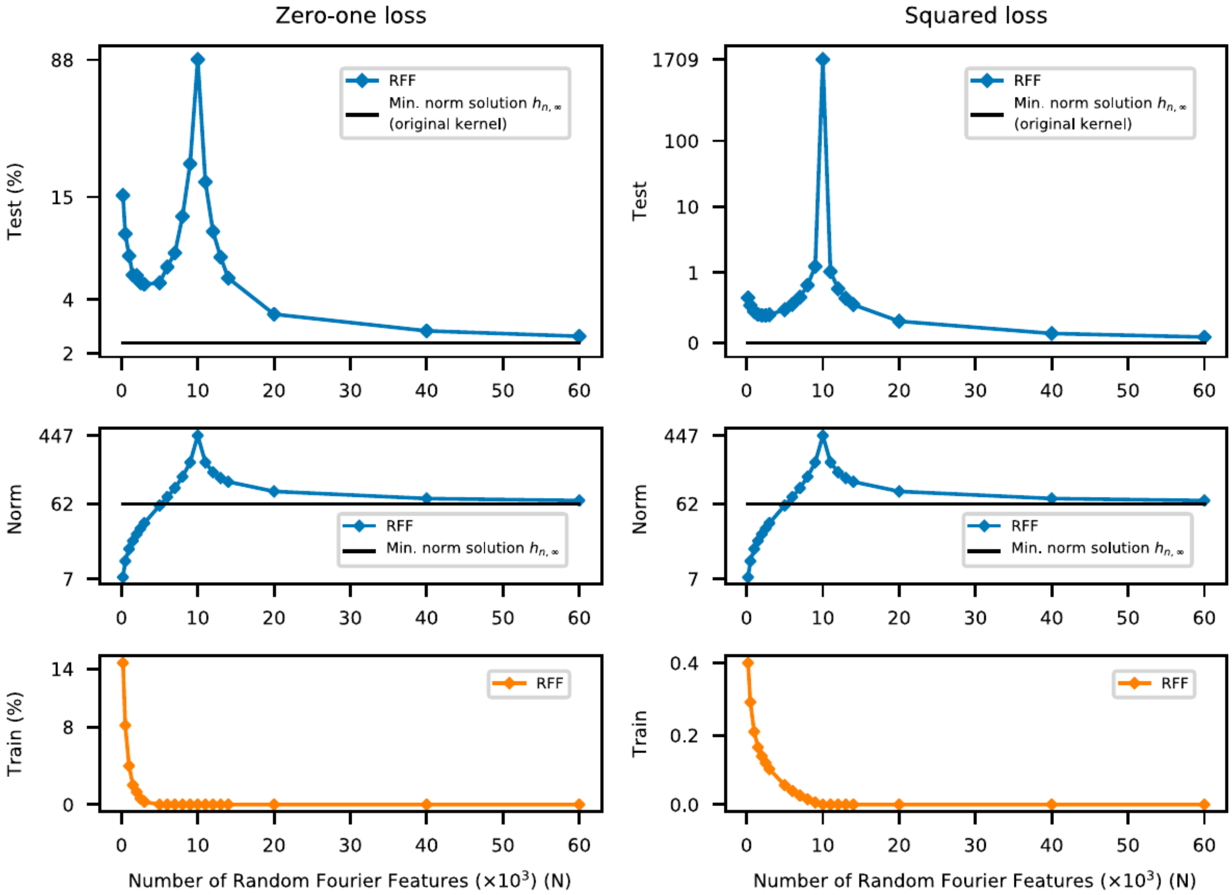


Figure 4.2: Experimental Results: RFF function class on MNIST. $n = 10^4$, test risk and coefficient norm is in logarithmic scale.

A plausible explanation is that the capacity of the function class might not be a good inductive bias (or a good indicator for the inductive bias) for the problem. For example, if the inductive bias is a smooth function, or a smaller norm, or a larger margin (for classification), then by considering a class with a wider range of fucntions, the algorithm may find a function that is able to better fit that inductive bias.

For example, in the case of the experiment presented, we are able to find a with smaller L2-norm when we increase N, as was shown in the Figure 4.2, which might explain the decrease of test risk.

5. Approximation and Estimation

5.1 Notations and Definitions

Let $\pi_m(\mathbb{R}^d)$ be a multivariate polynomial with d variables and degree $\leq m$, i.e.

$$\pi_m(\mathbb{R}^d) = \{p(x) = \sum_{k \leq m} c_k x^k\}$$

Let $C^k(X)$ be the set of functions on X that are k times continuously differentiable.

For a point $x \in \mathbb{R}^d$, it has the components of its coordinates $\chi_1, .., \chi_d$, whereas we represent n points in $\mathbb{R}^d$ as $x_1, .., x_n$.

We denote $\mathbb{N}_0$ as the set of non-negative integers.

We denote the multi-index vector with its components as $\alpha = (\alpha_1, ..., \alpha_d)^{\mathrm{T}} \in \mathbb{N}_0^d$, and $|\alpha| := \|\alpha\|_1$.

For $X \subseteq \mathbb{R}^d$, $f \in C^k(X)$, $|\alpha| \leq k$ and $x \in \mathbb{R}^d$, we denote:

$$D^\alpha f := \frac{\partial^{|\alpha|}}{\partial \chi_1^{\alpha_1} \cdots \partial \chi_d^{\alpha_d}} f$$

Definition 11. Consider the set of points $X = \{x_1, ..., x_n\} \subseteq \mathbb{R}^d$ and $\pi_m(\mathbb{R}^d)$ with $n \geq \dim(\pi_m(\mathbb{R}^d))$. X is called $\pi_m(\mathbb{R}^d)$-unisolvent if there is no polynomial in $\pi_m(\mathbb{R}^d)$ (besides the zero polynomial) that is zero on all the points.

5.2 Interpolation Estimates

Let $X = \{x_1, ..., x_n\}$ be a set of points that is $\mathcal{P}$ unisolvent. For $p_1, .., p_Q$ that form a basis of $\mathcal{P}$, let $P = (p_j(x_i)) \in \mathbb{R}^{n \times Q}$. Let Φ be a positive definite kernel and $A = (\Phi(x_i, x_j)) \in \mathbb{R}^{n \times n}$. We let $e^{(j)}$ represent the jth unit vector. Consider the linear system:

$$\begin{pmatrix} A & P \\ P^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \alpha^{(j)} \\ \beta^{(j)} \end{pmatrix} = \begin{pmatrix} e^{(j)} \\ 0 \end{pmatrix}. \tag{5.1}$$

This linear system is uniquely solvable for $j \in \{1, 2, .., n\}$.

Proof. For $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ that lies in the null space of the matrix $\begin{pmatrix} A & P \\ P^{\mathrm{T}} & 0 \end{pmatrix}$, we have the 2 equations:

$$\begin{aligned} &A\alpha + P\beta = 0, \\ &P^{\mathrm{T}}\alpha = 0. \end{aligned}$$

From the first equation, multiplying both sides by α^{T}, we get:

$$\begin{aligned} &\alpha^{\mathrm{T}}A\alpha + \alpha^{\mathrm{T}}P\beta = 0 \\ \Longrightarrow &\alpha^{\mathrm{T}}A\alpha + (P^{\mathrm{T}}\alpha)^{\mathrm{T}}\beta = 0 \\ \Longrightarrow &\alpha^{\mathrm{T}}A\alpha = 0 \end{aligned}$$

Since we know Φ is a positive definite kernel, then $\alpha = 0$, so we know $P\beta = 0$. Since X is $\mathcal{P}$-unisolvent, so $\beta = 0$. Hence, the linear system (5.1) is uniquely solvable. □

Let u_j^*, V_X be defined such that:

$$\begin{aligned} u_j^* &:= \sum_{i=1}^{n} \alpha_i^{(j)} \Phi(\cdot, x_i) + \sum_{k=1}^{Q} \beta_k^j p_k \\ V_X &:= \{\sum_{i=1}^{n} \alpha_i \Phi(\cdot, x_i) \ : \sum_{i=1}^{n} \alpha_i p(x_i) = 0, p \in \mathcal{P}\} + \mathcal{P}. \end{aligned} \tag{5.2}$$

So we have $u_j^* \in V_X$ and

$$u_i^*(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases}$$

Also, for all $f \in V_X$, we have

$$f = \sum_{i=1}^{n} u_i^* f(x_i). \tag{5.3}$$

For our next theorem, we further define:

$$\begin{aligned} R(\cdot) &:= (\Phi(\cdot, x_1), ..., \Phi(\cdot, x_n))^{\mathrm{T}} \in \mathbb{R}^n \\ S(\cdot) &:= (p_1(\cdot), ..., p_Q(\cdot))^{\mathrm{T}} \in \mathbb{R}^Q \end{aligned} \tag{5.4}$$

Theorem 5. *For Φ a positive definite kernel on $\Omega \subseteq \mathbb{R}^d$ and points $X = \{x_1, ..., x_n\} \subseteq \Omega$ is $\mathcal{P}$-unisolvent. There exists functions $v_i^*(\cdot)$ for $i = \{1, 2, .., Q\}$ such that for $u^*(x) =$*

$(u_1^*(x), ..., u_n^*(x))$ where u_i^* is defined in (5.2). we have:

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} u^*(x) \\ v^*(x) \end{pmatrix} = \begin{pmatrix} R(x) \\ S(x) \end{pmatrix} \tag{5.5}$$

Proof. One of the things to be proven is that $P^{\mathrm{T}} u^*(x) = S(x)$. Since $\mathcal{P} \subseteq V_X$, then $(p_1, ..., p_Q) \in V_X$, so by (5.3), we have: $p_i(x) = \sum p_i(x_j) u_i^*(x)$, showing $P^{\mathrm{T}} u^*(x) = S(x)$. Next, we need to show that there exists $v*$ such that $Au^*(x) + Pv^*(x) = R(x)$, so we need to show $Au^*(x) - R(x) \in P(\mathbb{R}^Q)$.

$(P(\mathbb{R}^Q))^{\perp}$ is the null space of its transpose. Consider $\omega \in \text{null}(P^{\mathrm{T}}) \subseteq \mathbb{R}^n$, so we need to show that $\omega^{\mathrm{T}}(Au^*(x) - R(x)) = 0$, i.e. $\omega^{\mathrm{T}} Au^*(x) = \omega^{\mathrm{T}} R(x)$.

$$\begin{aligned} P^{\mathrm{T}}\omega &= 0 \\ \Longrightarrow \omega^{\mathrm{T}} R(x) &\in V_X \\ \omega^{\mathrm{T}} R(\cdot) &= \sum_{i=1}^{n} u_i^*(\cdot)\omega^{\mathrm{T}} R(x_i) \quad (\because (5.3)) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} u_i^*(\cdot)\omega_j \Phi(x_i, x_j) \\ &= \omega^{\mathrm{T}} Au^*(\cdot). \end{aligned}$$

□

Lemma 5. *v^* has the property where, for v_i with $1 \leq i \leq Q$, we have: $v_i(x_j) = 0$ for $1 \leq j \leq n$.*

Proof.

$$\begin{aligned} Au^*(x_i) &= Ae^{(i)} \\ &= \begin{pmatrix} \Phi(x_i, x_1) \\ \vdots \\ \Phi(x_i, x_n) \end{pmatrix} \\ Au^*(x_i) + Pv^*(x_i) &= R(x_i) = Au^*(x_i) \\ \Longrightarrow Pv^*(x_i) &= 0 \end{aligned}$$

Since X is $\mathcal{P}$-unisolvent, $v^*(x_i) = 0$. □

We are able to rewrite an interpolant as:

$$s_{f,X}(\cdot) = \sum_{i=1}^{n} f(x_i)u_i^*(\cdot). \tag{5.6}$$

By differentiating 5.5, we have:

$$\begin{pmatrix} A & P \\ P^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} D^a u^*(x) \\ D^a v^*(x) \end{pmatrix} = \begin{pmatrix} D^a R(x) \\ D^a S(x) \end{pmatrix}. \tag{5.7}$$

We will define the power function, as defined in Definition 11.2 in Wendland (2004):

Definition 12. Suppose $X \subset \mathbb{R}^d$ is open, with $k : X \times X \to \mathbb{R}$ be a positive definite kernel. For $\alpha \in \mathbb{N}_0^d$, $\hat{X} = \{x_1, x_2, ..., x_n\} \subseteq X$, the power function $P_{k,\hat{X}}^{(\alpha)}(x)$ is defined by:

$$\begin{aligned} (P_{k,\hat{X}}^{(\alpha)}(x))^2 := D_1^\alpha D_2^\alpha k(x,x) - 2\sum_{j=1}^{n} D^\alpha u_j^*(x) D_1^\alpha k(x, x_j) \\ + \sum_{i,j=1}^{n} D^\alpha u_i^*(x) D^\alpha u_j^*(x) k(x_i, x_j). \end{aligned}$$

Theorem 6. *For an open set $\Omega \subseteq \mathbb{R}$ and a positive definite kernel $k \subseteq C^{2k}(\Omega \times \Omega)$, and the set of points $X = \{x_1, ..., x_n\} \subseteq \Omega$ is $\mathcal{P}$-unisolvent. Let $\mathcal{H}$ be RKHS corresponding to the kernel k, a function $f \in \mathcal{H}$ and its interpolant be $s_{f,X}$. For every $x \in \Omega, a \in \mathbb{N}_0^d, |a| \leq k$, we have:*

$$|D^a f(\cdot) - D^a s_{f,X}(\cdot)| \leq P_{k,X}^{(a)}(\cdot) \|f\|_{\mathcal{H}}. \tag{5.8}$$

Proof. This is proved in Theorem 11.4 of Wendland (2004). We shall proof the case of $a = 0$, which is the case which will be used in a later theorem. First, we note that:

$$\begin{aligned} \|k(\cdot, x) - \sum_{i=1}^{n} u_i k(\cdot, x_i)\|_{\mathcal{H}}^2 &= k(x,x) - 2\sum_{i=1}^{n} u_i^* k(x, x_i) + \sum_{i,j=1}^{n} u_i^* u_j^* k(x_i, x_j) \\ &= (P_{k,X}^{(0)}(x))^2. \end{aligned}$$

Next, using (5.6), we have:

$$\begin{aligned} s_{f,X}(x) &= \sum_{i=1}^{n} f(x_i)u_i^*(x) \\ &= \sum_{i=1}^{n} u_i^*(x)\langle f, k(\cdot, x_i)\rangle_{\mathcal{H}} \quad (\because \text{reproducing 6property}) \\ &= \langle f, \sum_{i=1}^{n} u_i^*(x)k(\cdot, x_i)\rangle \\ \implies |f(x) - s_{f,X}(x)| &= |\langle f, k(\cdot, x)\rangle_{\mathcal{H}} - \langle f, \sum_{i=1}^{n} u_i^*(x)k(\cdot, x_i)\rangle| \\ &= |\langle f, k(\cdot, x) - \sum_{i=1}^{n} u_i^*(x)k(\cdot, x_i)\rangle_{\mathcal{H}}| \\ &\leq \|f\|_{\mathcal{H}}\|k(\cdot, x) - \sum_{i=1}^{n} u_i^*(x)k(\cdot, x_i)\|_{\mathcal{H}} \quad (\because \text{Cauchy-Schwarz inequality}) \\ &= \|f\|_{\mathcal{H}} P_{k,X}^{(0)}(x). \end{aligned}$$

□

Definition 13. The fill distance (or sometimes referred to as 'fill' for short) for a set of points $X = \{x_1, ..., x_N\} \subseteq \Omega$ for a bounded domain Ω is defined to be

$$h_{X,\Omega} := \sup_{x\in\Omega} \min_{1\leq j\leq N} \|x - x_j\|_2.$$

Theorem 7. *Let Ω be a cube in $\mathbb{R}^d$ and $k = \phi(\|\cdot\|_2)$ be a positive definite function with $f = \phi(\cdot)$ satisfying the condition that there exists l_0 and constant $M > 0$ such that for all $r \geq 0$ and $l \geq l_0$, $|f^{(l)}(r)| \leq l!M^l$. Then there exists a constant $c > 0$ such that the error between a function $f \in \mathcal{H}$ (where $\mathcal{H}$ is the RKHS corresponding to the kernel k), and its interpolant $s_{f,X}$ for all data points $X = \{x_1, ..., x_n\}$ can be bounded by:*

$$\|f - s_{f,X}\|_{L_\infty(\Omega)} \leq exp(-c/h_{X,\Omega})\|f\|_{\mathcal{H}}$$

with sufficiently small fill $h_{X,\Omega}$.

Proof. From Theorem 11.9 in Wendland (2004), we have:

$$P_{\Phi,X}^2(x) \leq [1 + c_1(2N)]^2 \|f - p\|_{L_\infty(G)}$$

Where G is on the interval $[0, 4(c_2(2N))^2h^2]$, $x \in \Omega$, $p \in \pi_N(\mathbb{R})$, $h = h_{X,\Omega}$.

From Theorem 11.21 in Wendland (2004), we have for sufficiently small fill distance $h_{X,\Omega} \leq \frac{c_0}{2N}$, there exists a constant γ_d, where the constants c_1, c_2 can be replaced by:

$$\begin{aligned} c_1(2N) &= \exp(2d\gamma_d(2N+1)) \\ c_2(2N) &= 2c_2N \end{aligned}$$

So G is on the interval $[0, 16N^2c_2^2h^2]$.

For p the Taylor series of f about 0, and up to the term t^N, we then have:

$$\begin{aligned} |f(t) - p(t)| &\leq t^{N+1}\frac{|f^{N+1}(t')|}{(N+1)!} \\ &\leq M^{N+1}t^{N+1} \quad \text{(By assumption)} \\ \Longrightarrow \|f-p\|_{L_\infty(G)} &\leq (M \cdot 16N^2c_2^2h^2)^{N+1} \\ &= (C_0N^2h^2)^{N+1} \quad \text{for constant } C_0 = Mc_2^2 \end{aligned}$$

Also, we have:

$$\begin{aligned} [1 + c_1(2N)]^2 &= [1 + \exp(2d\gamma_d(2N+1))]^2 \\ &\leq [2\exp(2d\gamma_d(2N+1))]^2 \\ &= 4\exp(4d\gamma_d(2N+1)) \\ &= \exp(\log 4 + 4d\gamma_d(2N+1)) \\ &\leq \exp(C_1(N+1)) \quad \text{for sufficiently large } C_1. \end{aligned}$$

We then have:

$$\begin{aligned} P^2_{\Phi,X}(x) &\leq [1 + c_1(2N)]^2\|f-p\|_{L_\infty(G)} \\ &\leq \exp(C_1(N+1))(C_0N^2h^2)^{N+1} \\ &= (C_0N^2h^2\exp(C_1))^{N+1} \\ &= (C_2N^2h^2)^{N+1} \quad \text{for constant } C_2 = C_0\exp(C_1). \end{aligned} \tag{5.9}$$

For $C_3 = \min(\frac{c_0}{2}, \frac{1}{\sqrt{e}C_2})$, and N such that

$$\frac{C_3}{N+1} \leq h \leq \frac{C_3}{N}$$

, which gives us:

$$\begin{aligned} h &\leq \frac{c_0}{2N}, \\ -(N+1) &\leq -C_3/h, \\ N^2h^2 \leq &C_3^2 \leq \frac{1}{eC_2} \\ \Longrightarrow &C_2N^2h^2 \leq 1/e. \end{aligned}$$

We then have:

$$P^2_{\Phi,X}(x) \leq e^{-(N+1)} \leq e^{-C_3/h}.$$

Now, using $C = C_3/2$ and $|f(x) - s_{f,X}(x)| \leq P_{\Phi,X}(x)|f|_{\mathcal{H}}$ (from Theorem (6)), we have:

$$|f(x) - s_{f,X}(x)| \leq e^{-C/h_{X,\Omega}}|f|_{\mathcal{H}},$$

which is what we wanted to prove. □

Theorem 8. *With the same conditions as Theorem 7, except that f satisfies the stricter condition $|f^{(l)}(r)| \leq M^l$, we can get a better error bound of:*

$$\|f - s_{f,X}\|_{L_\infty(\Omega)} \leq \exp(\frac{c\log(h_{X,\Omega})}{h_{X,\Omega}})\|f\|_{\mathcal{H}}.$$

Proof. The inequality at 5.9 becomes:

$$P^2_{\Phi,X}(x) \leq \frac{(C_2N^2h^2)^{N+1}}{(N+1)!}.$$

Using Stirling's inequality $1/n! \leq (e/n)^n$, we have:

$$P^2_{\Phi,X}(x) \leq (eC_2Nh^2)^{N+1}.$$

Similarly, with $C_3 = \min(\frac{c_0}{2}, \frac{1}{eC_2})$, and N such that

$$\frac{C_3}{N+1} \leq h \leq \frac{C_3}{N},$$

we then get

$$eC_2Nh \leq 1,$$

which gives us:

$$P^2_{\Phi,X}(x) \leq h^{N+1} \leq h^{C_3/h} = e^{C_3 \log h/h}.$$

Following the steps of the previous theorem then gives us our result. □

5.3 Approximation Theorem

5.3.1 Fill Distance of points in a Cube

First, we need the result that: with $h_{X,\Omega}$ as the fill on the order of $O(n/\log n)^{-1/d}$ (using the theorem S1 in Belkin's paper). We give a proof sketch for this:

Whern n is big enough, we (approximately) divide the cube Ω into n smaller subcubes, where each subcube has $1/n$ the volume of Ω (Let the volume of Ω be 1 for simplicity). Let g_k be a vertice of one of the subcubes. Let N be the number of vertices of the subcubes. Let's say all x_i are not within r distance from a point g.
Then $\min_{1\leq i\leq n}\|g - x_i\| > r > 0$ i.e. $x_i \notin B(g,r), 1 \leq i \leq n$.

$$\begin{aligned}
\mathbb{P}(x_i \notin B(g,r), 1 \leq i \leq n) &= \prod_{i=1}^{n} \mathbb{P}(x_i \notin B(g,r)) \\
&= [\mathbb{P}(x_i \notin B(g,r))]^n \\
&= [1 - \mathbb{P}(x_i \in B(g,r))] \\
&= (1 - c_d r^d)^n \\
&\approx \exp(-c_d r^d n).
\end{aligned}$$

Where c_d is some constant. It arises due to ratio of volume of ball vs volume of where x_i can be. The distance from a point x in the cube to some $x_i \leq \operatorname{dist}(x, g_k) + \operatorname{dist}(g_k, x_i)$. For g_k being one of the vertices of the subcube x resides in. For the subcube with length a and dimension d and volume n^{-1}, since

$$a_d = n^{-1} \Longrightarrow a = n^{-1/d}.$$

So the end-to-end distance in the subcube is $a\sqrt{d} = n^{-1/d}\sqrt{d}$, which goes to 0 for big n, so $\operatorname{dist}(x, g_k)$ will be small.

The distance from a point x in the cube to some $x_i \approx \text{dist}(g_k, x_i)$ for large n.

$$\begin{aligned}
\mathbb{P}(h_{X,\Omega} > r) &\approx \mathbb{P}(\exists \text{ vertice } g_k | x_i \notin B(g_k, r), 1 \leq i \leq n) \\
&= \mathbb{P}(\bigcup_k \{x_i \notin B(g_k, r), 1 \leq i \leq n\}) \\
&\leq \sum_{k=1}^{N} \mathbb{P}(x_i \notin B(g_k, r), 1 \leq i \leq n) \\
&= \sum_{k=1}^{N} \exp(-c_d r^d n) \\
&\leq 2^d n \exp(-c_d r^d n) \\
&= \exp(d \log 2 + \log n - c_d r^d n)
\end{aligned}$$

$\mathbb{P}(h_{X,\Omega} > r) \to 0$ when:

$$\begin{aligned}
&c_d r^d n >> \log n \\
\implies\ & r >> (\frac{\log n}{n})^{1/d} c_d^{1/d}.
\end{aligned}$$

i.e. We have a low probability that $\mathbb{P}(h_{X,\Omega} > r)$ if r grows with respect to $(\frac{n}{\log n})^{-1/d}$ (for large n).

The below theorem gives us some justification as to why the minimum norm interpolating function was chosen, though this only works under noiseless conditions:

Theorem 9. *For $\mathcal{H}$ an RKHS corresponding to a positive definite kernel, fix $h^* \in \mathcal{H}$. Let $(x_1, y_1), ..., (x_n, y_n)$ be i.i.d. random variables where x_i drawn uniformly at random from a compact cube $\Omega \subseteq \mathbb{R}^d$, $y_i = h^*(x_i) \forall i$. There exists $A, B > 0$ such that for any interpolating $h \in \mathcal{H}$ with high probability*

$$\sup_{x \in \Omega} |h(x) - h^*(x)| A e^{-B(n/\log n)^{1/d}} (\|h^*\|_{\mathcal{H}} + \|h\|_{\mathcal{H}})$$

Proof. We consider $f(x) := h(x) - h^*(x)$. Since h is interpolating, we have $f(x_i) = 0$ for all x_i. The interpolant of f, $s_{f,X}$ will then be the zero function. Thus, using Theorem 7, we have:

$$\begin{aligned}\|f\|_{L_\infty(\Omega)} &= \sup_{x\in\Omega} |h(x) - h^*(x)| \\ &< \exp(-c(n/\log n)^{1/d})\|f\|_{\mathcal{H}} \\ &\leq \exp(-c(n/\log n)^{1/d})(\|h^*\|_{\mathcal{H}} + \|h\|_{\mathcal{H}})\end{aligned}$$

□

Another form we can have is using Proposition 14.1 in Wendland (2004):

Proposition 2. *Let $\Omega \subseteq \mathbb{R}^d$ be bounded and measurable. Suppose $X = \{x_1, ..., x_N\} \subseteq \Omega$ is quasi-uniform with respect to $c_{qu} > 0$. Then there exists constants $c_1, c_2 > 0$ depending only ond, Ω and on c_{qu} such that:*

$$c_1 N^{-1/d} \leq h_{X,\Omega} \leq c_2 N^{-1/d}.$$

With the definition of quasi-uniformness being:

Definition 14. For the separation distance of $X = \{x_1, ..., x_N\}$ being defined as

$$q_X := \frac{1}{2} \min_{i\neq j} \|x_i - x_j\|_2.$$

X is **quasi-uniform** with respect to $c_{qu} > 0$ if

$$q_X \leq h_{X,\Omega} \leq c_{qu} q_X$$

We can then use the above proposition with n replacing $n/\log n$ in Theorem 9 for a better bound of:

$$\sup_{x\in\Omega} |h(x) - h^*(x)| < Ae^{-B(n)^{1/d}}(\|h^*\|_{\mathcal{H}} + \|h\|_{\mathcal{H}})$$

if the quasi-uniformness conditions are fulfilled. In either case, by choosing a the smallest norm for h, we can see that it corresponds to the smallest upperbound for $|h(x) - h^*(x)|$. In the next subsection, we have another rationale for choosing a smaller $\|h\|_{\mathcal{H}}$:

5.3.2 Regularity of Functions

Let the $\mathbb{R}$-RKHS $\mathcal{H}$ consist of functions over X. The functions fulfill a Lipschitz-like condition with the Lipschitz constant being the RKHS norm of the function $\|f\|_{\mathcal{H}}$.
For the positive definite symmetric kernel k corresponding to $\mathcal{H}$, we have for all $x, x' \in X$:

$$\begin{aligned} |f(x) - f(x')| &= |\langle f, k(\cdot, x)\rangle_{\mathcal{H}} - \langle f, k(\cdot, x')\rangle_{\mathcal{H}}| \\ &= |\langle f, \Phi(x) - \Phi(x')\rangle_{\mathcal{H}}| \;\; \text{where } \Phi(x)(\cdot) = k(\cdot, x) \\ &\leq \|f\|_{\mathcal{H}} d(x, x'), \end{aligned}$$

where $d(x, x')$ is the pseudometric defined by:

$$d^2(x, x') = k(x, x) - 2k(x, x') + k(x', x').$$

Intuitively, we can see the "speed" of which the function can change is bounded by the norm of the function in the RKHS space.

6. Other Notes

It has been oberved in earlier papers (for example, Shalev-Shwartz et al. (2010)) that low regularization results in optimal performance for kernel machines. Similarly strong performance has also been seen in Random Forests and Adaboost (Wyner et al. (2015)).

6.1 Weaknesses

The double descent curve is harder to observe in multilayer neural networks. One of the reasons being that it is very sensitive to initialization when the model is underparameterized. Also, as shown in Belkin et al. (2018a), adversarial examples are unavoidable when using interpolation learning in non-deterministic data labels. As the sample size increases, the adversarial cases are asymptotically dense, so there is always an adversarial example arbitrarily close to any point.

A. Appendix

A.1 Results with Artificial Data and Noise

A.1.1 Setup

The paper looks at the double descent curve in the situation of artificial data. Define a function $e_k : [0, 2\pi] \to \mathbb{C}$ where

$$e_k(\cdot) := \exp(\sqrt{-1}(k-1)\cdot)$$

where $k \in \mathbb{Z}^+$. Define a probability distribution p on the set of positive integers. Denote $p_k = \mathbb{P}(k), k \in \mathbb{Z}^+$. Let the target function be

$$h^*(\cdot) = \sum_{k \in \mathbb{Z}^+} a_k^* e_k(\cdot)$$

and we let $a_k^* := p_k \forall k$. The datapoints $(x_1, y_1), ..., (x_n, y_n)$ be $x_j \in [0, 2\pi]$ and

$$y_j = h^*(x_j) + \sigma \epsilon_j$$

where $\epsilon_1, ..., \epsilon_n$ are sampled independently from the standard normal distribution.
The signal-to-noise ratio is defined as:

$$\text{SNR} = \frac{\mathbb{E}[h^*(x_i)^2]}{\sigma^2}.$$

The function class $\mathcal{H}_N$ is randomly chosen by:

- Choose N positive integers $k_1, ..., k_N$ sampling from p independently.
- Consider the span of the functions $\{e_{k_1}, ..., e_{k_N}\}$ as the function class $\mathcal{H}_N$.

The regime of choosing the function from the function class is similar to what was used in the RFF experiment 4.3. Choose the function with the smallest square error in the training data. When the training error is 0 (when $N \geq n$), choose the smallest norm function with the 0 error, where the norm of the function $h = \sum_{j=1}^{N} a_{k_j} e_{k_j}$is defined as: $\|h\|_{\mathcal{H}}^2 = \sum_k \frac{a_k}{p_k}$.

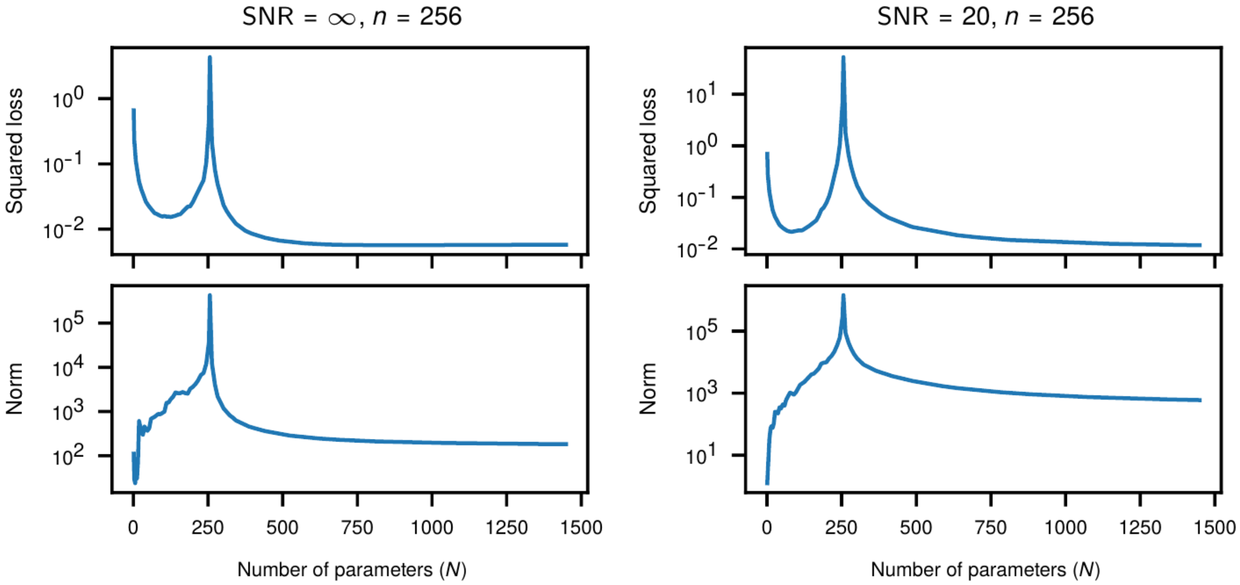


Figure A.1: Left: Noiseless labels. Right: SNR=20. Top: Test MSE. Bottom: norm of function $\|h\|_{\mathcal{H}}$.

A.1.2 Results

The experiment was repeated over multiple n, multiple SNR (by varying σ), and repeated 20 times, and the double descent curves were observed. An example shown is in Figure A.1.

A.2 Results with Random Forests

A.2.1 Setup

The (real life) training data are trained using random forests for regression problems, as described in Breiman (2001). The number of features to be selected when splitting a node is the square root of the number of total features. The maximum number of leaves in each tree (denoted by N_{leaf}^{max}), and the number of trees (denoted by N_{tree}) are varied to change the capacity of the function class. There is also no constraints on the depth of each tree, or when to split the tree node. At most n leaves is required for a tree to interpolate n samples (though the tree tends to interpolate with fewer leaves).

Training data was interpolated in 2 different ways. In Figure A.2 (A), the experiments were run with no bootstrap re-sampling, as describe in Cutler & Zhao (2001). In (B), the same experiments were run with bootstrap re-sampling. Note that the training data can be interpolated with a single tree with no bootstrap re-sampling, but have to be interpolated

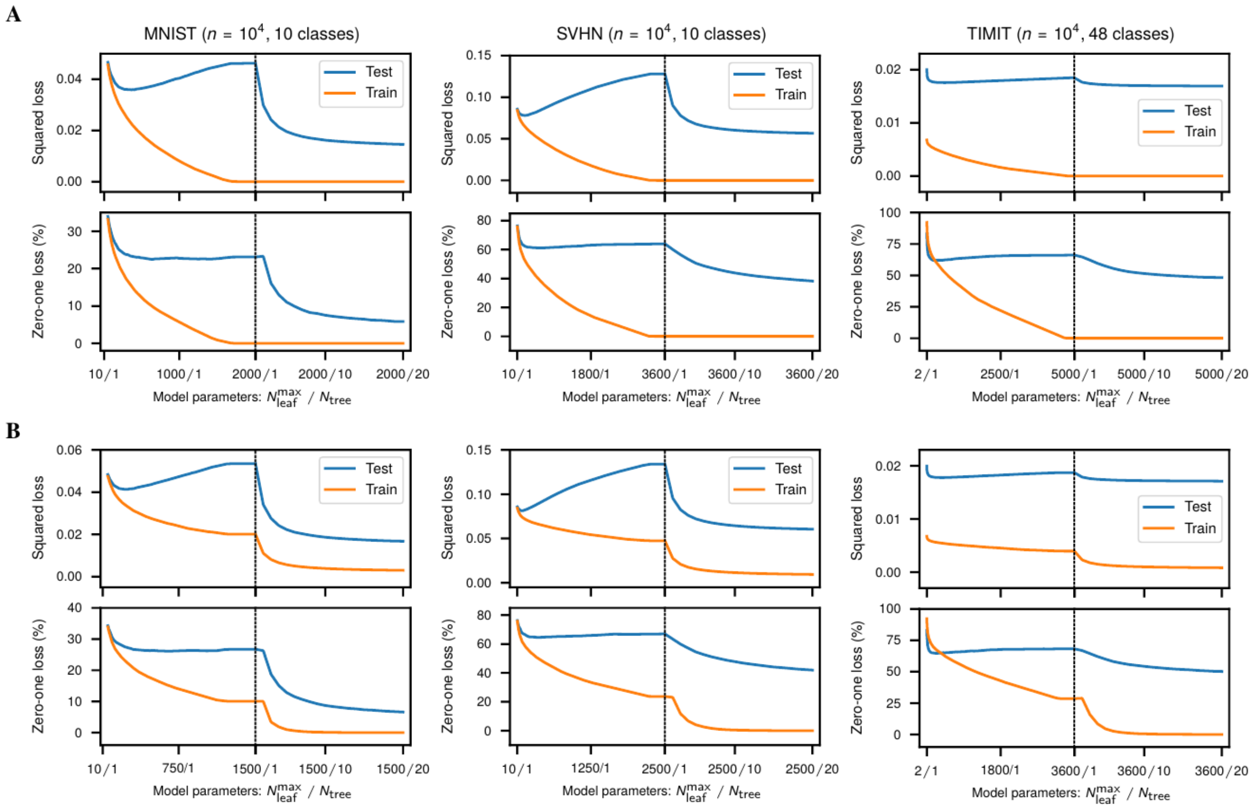


Figure A.2: Model capacity altered with N_{leaf}^{max} and N_{tree}.(A): Bootstrap re-sampling disabled. (B): Bootstrap re-sampling enabled.

with multiple trees with bootstrap resampling.

As we can see in the figure, the double descent curve can be observed in both cases, though they tend to be less obvious in the zero-one loss graphs.

Bibliography

Belkin, Mikhail. 2018. Approximation beats concentration? an approximation view on inference with smooth radial kernels. *arxiv:1801.03437* http://arxiv.org/abs/1801.03437v2.

Belkin, Mikhail, Daniel Hsu, Siyuan Ma & Soumik Mandal. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc Natl Acad Sci USA* 116(32). 15849–15854. doi:10.1073/pnas.1903070116. https://doi.org/10.1073%2Fpnas.1903070116.

Belkin, Mikhail, Daniel Hsu & Partha Mitra. 2018a. Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate.

Belkin, Mikhail, Siyuan Ma & Soumik Mandal. 2018b. To understand deep learning we need to understand kernel learning. *arxiv:1802.01396* http://arxiv.org/abs/1802.01396v3.

Breiman, Leo. 2001. *Machine Learning* 45(1). 5–32. doi:10.1023/a:1010933404324. https://doi.org/10.1023%2Fa%3A1010933404324.

Canziani, Alfredo, Adam Paszke & Eugenio Culurciello. 2016. An analysis of deep neural network models for practical applications .

Cutler, Adele & Guohua Zhao. 2001. Pert-perfect random tree ensembles. *Computing Science and Statistics* 33.

Kégl, Balázs, Tamás Linder & Gábor Lugosi. 2001. Data-dependent margin-based generalization bounds for classification. In *Lecture notes in computer science*, 368–384. Springer Berlin Heidelberg. doi:10.1007/3-540-44581-1_24. https://doi.org/10.1007%2F3-540-44581-1_24.

Krizhevsky, A & G Hinton. 2009. Learning multiple layers of features from tiny images .

Rahimi, Ali & Benjamin Recht. 2008. Random features for large-scale kernel machines. In J. Platt, D. Koller, Y. Singer & S. Roweis (eds.), *Advances in neural information processing systems*, vol. 20, 1177–1184. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Rudi, Alessandro, Raffaello Camoriano & Lorenzo Rosasco. 2015. Less is more:

Nyström computational regularization. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama & R. Garnett (eds.), *Advances in neural information processing systems*, vol. 28, Curran Associates, Inc. https://proceedings.neurips.cc/paper/2015/file/03e0704b5690a2dee1861dc3ad3316c9-Paper.pdf.

Rudin, Walter. 1990. *Fourier analysis on groups.* John Wiley & Sons, Inc. doi:10.1002/9781118165621. https://doi.org/10.1002%2F9781118165621.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg & Li Fei-Fei. 2015. ImageNet large scale visual recognition challenge. *Int J Comput Vis* 115(3). 211–252. doi:10.1007/s11263-015-0816-y. https://doi.org/10.1007%2Fs11263-015-0816-y.

Schölkopf, Bernhard, Ralf Herbrich & Alex J. Smola. 2001. A generalized representer theorem. In *Lecture notes in computer science*, 416–426. Springer Berlin Heidelberg. doi:10.1007/3-540-44581-1_27. https://doi.org/10.1007%2F3-540-44581-1_27.

Shalev-Shwartz, Shai, Yoram Singer, Nathan Srebro & Andrew Cotter. 2010. Pegasos: primal estimated sub-gradient solver for SVM. *Math. Program.* 127(1). 3–30. doi:10.1007/s10107-010-0420-4. https://doi.org/10.1007%2Fs10107-010-0420-4.

Steinwart, Ingo & Andreas Christmann. 2008. *Support vector machines.* Springer New York. doi:10.1007/978-0-387-77242-4. https://doi.org/10.1007%2F978-0-387-77242-4.

Wendland, Holger. 2004. *Scattered data approximation.* Cambridge Monographs on APplied and Computational Mathematics. Cambridge University Press. doi:10.1017/CBO9780511617539.

Wyner, Abraham J., Matthew Olson, Justin Bleich & David Mease. 2015. Explaining the success of adaboost and random forests as interpolating classifiers.

Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht & Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization .