Optimal Learning*

Peter Binev†[1], Andrea Bonito[2], Ronald DeVore[2], and Guergana Petrova[2]

[1]Department of Mathematics, University of South Carolina, Columbia, SC 29208
[2]Department of Mathematics, Texas A&M University, College Station, TX 77843

March 31, 2022

Abstract

This paper studies the problem of learning an unknown function f from given data about f. The learning problem is to give an approximation $\hat{f}$ to f that predicts the values of f away from the data. There are numerous settings for this learning problem depending on (i) what additional information we have about f (known as a model class assumption), (ii) how we measure the accuracy of how well $\hat{f}$ predicts f, (iii) what is known about the data and data sites, (iv) whether the data observations are polluted by noise. A mathematical description of the optimal performance possible (the smallest possible error of recovery) is known in the presence of a model class assumption. Under standard model class assumptions, it is shown in this paper that a near optimal $\hat{f}$ can be found by solving a certain discrete over-parameterized optimization problem with a penalty term. Here, near optimal means that the error is bounded by a fixed constant times the optimal error. This explains the advantage of over-parameterization which is commonly used in modern machine learning. The main results of this paper prove that over-parameterized learning with an appropriate loss function gives a near optimal approximation $\hat{f}$ of the function f from which the data is collected. Quantitative bounds are given for how much over-parameterization needs to be employed and how the penalization needs to be scaled in order to guarantee a near optimal recovery of f. An extension of these results to the case where the data is polluted by additive deterministic noise is also given.

1 Introduction

Learning an unknown function f from given data observations is a dominant theme in data science. The central problem is to use the data observations of f to construct a function $\hat{f}$ which approximates f away from the data. This paper is concerned with evaluating how well such an approximation $\hat{f}$ performs and determining the best possible performance among all choices of an $\hat{f}$. Given answers to these fundamental questions, we then turn to the construction of numerical procedures and evaluate their performance against the known best possible performance.

Key words and phrases. Learning from Data, approximation from data, optimal algorithms, model classes.

*2020 *Mathematics subject classification.* 41A65, 41A46, 41A63

*This research was supported by the ONR Contract N00014-20-1-278, the NSF Grants DMS 1720297, DMS 2038080, DMS 2110811, DMS 2134077, and the NSF-Tripods Grant CCF-1934904.

†Corresponding author: binev@math.sc.edu

To place ourselves in a firm mathematical setting, we assume that f is in some Banach space X of functions and the performance of the approximant $\hat{f}$ is measured by $\|f - \hat{f}\|_X$. Typical choices for X are the $L_p(\Omega)$ spaces with Ω a domain in $\mathbb{R}^d$, or smoothness spaces such as Sobolev spaces on Ω. The latter case arises in the context of solving Partial Differential Equations (PDEs).

In the absence of additional information about f, it is easy to see that there can be no performance guarantee, i.e. for any choice of $\hat{f}$, the error $\|f - \hat{f}\|_X$ can be arbitrarily large for a function f which satisfies the data. The additional information we impose on f is referred to as model class information. The appropriate model class for a learning problem depends very much on the underlying application and is a compilation of all that is known about the function f from analysis of the application. For example, in PDE applications, the model class is typically provided by physics or regularity theorems for the PDE in hand. In other applications, such as image or video classification, appropriate model classes are less transparent and open for debate.

Mathematically, a *model class* is a compact subset K of X. Given such a model class, the learning problem is to determine a best approximant $\hat{f}$ given only the information that f is in K and f satisfies the given data. A best function $\hat{f}$ is called the *optimal recovery* of f.

Optimal recovery has the following well-known mathematical description (see e.g. [17, 27, 6]). Let us denote the set of all possible candidates for f by K^*, i.e.,

$$K^* := \{f \in K : f \text{ satisfies the data}\}. \tag{1.1}$$

Let $B(K^*)_X$ be a smallest ball (usually called the *Chebyshev ball*) in X that contains K^* and let $z = z(K^*) \in X$ be its center and $R = R(K^*)_X$ be its radius. Then, a best approximation to f is $\hat{f} := z$ and the best error in recovering f from the given information is R.

While this gives a simple mathematical description of the optimal recovery of f, it is nowhere close to giving a numerical algorithm for learning since finding z is a numerical challenge whose difficulty depends on the nature of K. Nevertheless, the radius $R(K^*)_X$ gives a benchmark for measuring the success of a numerical procedure. If a numerical procedure produces an $\hat{f} \in X$ that can be shown to give an error

$$\|f - \hat{f}\|_X \le CR(K^*), \quad f \in K^*, \tag{1.2}$$

it is said to be a *near optimal* recovery of f with constant C. Notice that any function $\hat{f} \in B(K^*)_X$ is a near optimal recovery with constant 2. If a numerical procedure is shown to produce a near optimal recovery $\hat{f}$ of f, one can rest assured that no other numerical method will perform better save for the size of the constant C and issues centering on the numerical cost to implement the algorithm.

1.1 Dependence on data

The error $R(K^*)_X$ of optimal recovery depends very much on the given data. We assume throughout our paper that this data is given by the values of m linear functionals $\lambda_1, \dots, \lambda_m$ applied to f. These linear functionals should be defined for all functions in K. In the simplest setting of noiseless data, the values

$$w_i := \lambda_i(f), \quad i = 1, \dots, m, \tag{1.3}$$

is the data information provided to us about f. Instead of K^* we shall use the notation

$$K_w := \{f \in K : \lambda_i(f) = w_i,\ i = 1, \dots, m\}, \quad w = (w_1, \dots, w_m), \tag{1.4}$$

to indicate the dependence of this set on the data. With this notation, the optimal recovery rate of f from the given information is

$$\textbf{optimal recovery rate} \;=\; R(K_w)_X. \tag{1.5}$$

Given that the m data functionals $\lambda_1, \ldots, \lambda_m$ are fixed, we define

$$W := W_K := W(\lambda_1, \ldots, \lambda_m; K) := \{(\lambda_1(g), \ldots, \lambda_m(g)) : \; g \in K\} \subset \mathbb{R}^m, \tag{1.6}$$

which is the set of all possible data vectors that can arise from observing an $f \in K$. So, K_w is defined for all $w \in W$. For all other $w \in \mathbb{R}^m$, we define $K_w := \emptyset$.

The most convenient assumption to make about the λ_j's is that they are linear functionals from the dual space X^* of X. However, a common setting in learning is to measure error in the $X = L_2(\Omega, \mu)$ norm, where $\Omega \subset \mathbb{R}^d$ and μ is a Borel measure, and to assume that the data are point values of f, which of course are not linear functionals on all of X in this case. The latter case can be treated if the model class K admits point evaluation. A natural assumption in this case is that $K \subset C(\Omega)$, where $C(\Omega)$ is the space of continuous functions defined on Ω. Another common setting for point evaluation is to assume that $K \subset H$, where H is a reproducing kernel Hilbert space (RKHS) which may be different from the space X where we measure performance. In §4, we study point evaluation in cases where X itself does not admit point evaluations as linear functionals.

There are the following common settings for the data observations:

Setting I: The common general setting is that the λ_j's are any fixed linear functionals defined on K and we had no influence in their choice.

Setting II: We are free to choose the functionals λ_j, subject to the restriction that there are only m of them.

Setting III: The λ_j's are given by a random selection of m independent draws under some probability distribution.

Settings IV,V,VI: The functionals are chosen as in the above cases (I, II, III) but are restricted to come from a dictionary of possible functionals. Point evaluation falls into this setting.

Since **Setting I** is the most often used, in this paper we try to stay within this setting as much as possible. **Setting II** is usually referred to as *directed learning* and is a well studied setting in functional analysis. If the functionals are allowed to be any m functionals from X^* with the budget m fixed, then the best choice for the m functionals gives an optimal recovery rate

$$d^m(K)_X := \inf_{\lambda_1, \ldots, \lambda_m \in X^*} \sup_{w \in \mathbb{R}^m} R(K_w)_X \tag{1.7}$$

which is known as the *Gelfand width* of K. For classical model classes such as the unit ball $K := U(Y)$ of a smoothness space Y that embeds into X, the Gelfand widths are known at least asymptotically as $m \to \infty$. Standard reference for results on Gelfand widths in classical settings are [23, 16] and the citations therein. Notice that the Gelfand width would tell us the minimum number m of measurements needed to guarantee a desired accuracy of performance. Namely, if we desire recovery error at most ε, then we would need m large enough so that $d^m(K)_X \leq \varepsilon$.

Setting III seems to be the most often studied in modern learning. The optimal performance in this case is given by the expected width

$$d^m_{\text{ave}}(K)_X := \text{Exp}_{\lambda_1,\dots,\lambda_m} \sup_{w\in\mathbb{R}^m} R(K_w)_X, \tag{1.8}$$

where the expectation is taken with respect to random independent draws according to the underlying probability measure.

If restrictions are placed on which measurement functionals are allowed to be used, then the notions of Gelfand widths and expected widths are modified accordingly. In the case that these functionals are required to be point evaluations, the corresponding Gelfand width is referred to as the sampling numbers of K and the information is referred to as *standard information* in the field of Information Based Complexity (IBC). We will denote sampling numbers by

$$s_m(K)_X \tag{1.9}$$

in this paper. Two of the standard references on this line of investigation are [27, 20]. Because of their importance in learning, finding the sampling numbers for various model classes K is an active research topic (see e.g. [4, 15, 13, 18]).

Although this is not the theme of the present paper, let us emphasize that computing the Gelfand widths and expected widths of model classes K is an important problem in analysis. It is also important for the learning community since it gives the best performance that would be possible in a numerical algorithm for learning, and therefore it can serve as a benchmark for evaluating the performance of a particular proposed algorithm. While quite a bit is known about these widths for classical model classes K, most of the known results are not useful in modern learning. Namely, it is known that for classical model classes the sampling numbers (see [13]) and Gelfand widths suffer the curse of dimensionality. This precludes the use of these classical model classes in modern learning where the dimension d of the physical space is very large (for example $d > 10^4$ in many classification problems). Hence, a general open question is to find appropriate model classes in high dimensions that match the directed application and then show that their sampling numbers and/or Gelfand widths avoid the curse of dimensionality.

1.2 Discretization of the learning problem

The above notions are abstract and do not provide a numerical recipe for learning. Rather, they provide only a benchmark for optimal performance. The goal of learning is to construct a numerical algorithm that provably converges to an optimal or near optimal recovery of f, i.e., reaches the optimal benchmark. The development of numerical learning algorithms usually proceeds through two stages. The first is to formulate a discrete optimization problem associated to the data. The second stage is to propose and analyze numerical procedures for solving the discrete optimization. In this paper, we shall primarily concern ourselves with the first stage and ask the question:

Which discrete optimization problems, if they are successfully numerically implemented, are guaranteed to provide the optimal learning possible from the given data?

This paper provides an answer to this question in a variety of settings. Namely, it is shown that the solution to a suitable over-parameterized penalized least squares optimization problem gives a near optimal learning algorithm. This fact may shed some light on why over-parameterized learning

using neural networks is preferred in modern machine learning. We touch upon techniques for numerically implementing the discrete optimization only briefly when we discuss some concrete examples.

1.3 Outline of the paper

In the next section, we begin by recalling the mathematical description of optimal learning algorithms based on Chebyshev balls. The remainder of the paper concentrates on introducing discrete minimization formulations, under a model class assumption, whose solution is near optimal. Each of these discrete minimizations can be taken of the form

$$\hat{f} \in \operatorname*{argmin}_{g \in \Sigma_n} \left(\tau \sum_{j=1}^{m} [w_j - \lambda_j(g)]^2 + \mu \ \mathbf{pen}_K(g) \right), \tag{1.10}$$

where $\tau, \mu > 0$ are suitably chosen parameters, Σ_n is a linear space of dimension n or a nonlinear space described by n parameters, and **pen** is a penalty term depending on the model class K.

Remark 1.1. *Note that in general, the minimization problem in* (1.10) *may not have a unique solution from* Σ_n*. Unless stated otherwise, the statements of the theorems that follow hold for any minimizer.*

We determine a penalty term for each model class K of X and each of the above settings for the data. These are given in §3. If the model class K has additional structure, for example if it is convex and centrally symmetric about the origin, then the penalty can be simplified and is presented in §3.1. The case of data consisting of point evaluations needs a slightly different treatment since the data is no longer necessarily given by a linear functional on X. Point evaluation is considered in §4.

The above description of the learning problem assumes that the data are exact. A more realistic assumption is that the data observations are corrupted by noise. We have chosen to treat the noiseless case first and then later address how the addition of noise deteriorates the accuracy of best recovery. In §5, we consider the case when the data observations are corrupted by additive noise. In this paper, we do not treat the more common assumption in statistics of stochastic noise and the corresponding minimax estimates since the treatment of that case requires substantially new ideas. However, we do discuss the case of random sampling in §7. In numerical implementations it is convenient to use other forms of the loss function appearing in (1.10). We discuss this aspect in §6.

In the final section of this paper, we study a couple of specific settings in learning with the aim of discussing the numerical aspects of implementing the proposed discrete optimization. In our first example, we treat the case when the model class K is the unit ball of a Sobolev space and the recovery error is measured in $L_2(\Omega)$. This setting is not realistic for the modern problems of learning, but it does allow us to put forward a specific numerical method for solving the optimal discretization for which convergence of the numerical method is known, namely the Finite Element Method. Our second example is more germane to modern learning. While we continue to measure error in $L_2(\Omega)$, the model class K is taken as the convex hull of the ReLU single layer neural network dictionary. We describe the correct optimization problem for an optimal learning algorithm. While much is known about solutions to this discrete problem [21, 24] and numerical methods for solving the optimization problem [26, 11], very fundamental questions concerning what is the asymptotic

behavior of the optimal error of recovery are not yet settled. This is discussed in more details in §8.2.

2 Learning in a Banach space setting

We begin by considering the case where we measure accuracy in a Banach space X and the model class K is simply a compact subset of X. We assume that the data are the observations (1.3), where the $\lambda_j \in X^*$ are linear functionals on X. The vector $w := (w_j)_{j=1}^m \in \mathbb{R}^m$ is called the *data observation vector* of the unknown f and the λ_j, $j = 1, \dots, m$, are the observation functionals. Without loss of generality, we can assume that the λ_j's are normalized to have norm one, $\|\lambda_j\|_{X^*} = 1$, $j = 1, \dots, m$. We shall also use the notation

$$\lambda(g) := (\lambda_1(g), \dots, \lambda_m(g)) \in \mathbb{R}^m, \quad g \in X, \tag{2.1}$$

and (1.6) throughout this paper. Notice that since λ is continuous on K and K is compact in X, the set W_K is a compact subset of $\mathbb{R}^m$. Obviously, all these quantities depend on the λ but we generally do not indicate this dependence since we think of the observation functionals as fixed.

As noted in the introduction, the totality of information we have about the unknown function f is that $f \in K_w$ for the given data observations w. As with K^*, we define $B(K_w)_X$ to be a Chebyshev (i.e., smallest) ball in X which contains K_w. An optimal recovery of f is the Chebyshev center z_w of $B(K_w)_X$ and the error of optimal recovery is the Chebyshev radius $R(K_w) := R(K_w)_X$ of $B(K_w)_X$. The goal of learning is to find a numerical procedure which would take the data and the knowledge of K and create an approximant $\hat{f} \in X$ such that

$$\|f - \hat{f}\|_X \le C R(K_w)_X, \tag{2.2}$$

with a reasonable constant C. We call such an approximant $\hat{f}$ a *near optimal recovery* for K_w with constant C.

Remark 2.1. *It can happen that $R(K_w)_X$ is zero. This would mean that there is only one function in K_w, i.e., only one function from K that fits the data. In this case we would only have near optimality in the above sense if $\hat{f} = f$. To avoid this exceptional case, we assume in going forward that $R(K_w)_X > 0$ in the theorems that follow. It is easy to formulate a version of each of these theorems to handle the case $R(K_w)_X = 0$ but we leave that task to the reader.*

In this paper, we are interested in formulating discrete optimization problems whose solution would provide a near optimal approximant $\hat{f}$ to f. We begin by giving sufficient conditions on a function $\hat{f}$ to be a near optimal approximant.

2.1 A preliminary result

It seems very doubtful that a numerical method would find an element $g \in K_w$ when given just w and the knowledge of K. A more reasonable numerical task would be to find a $g \in X$ such that g almost satisfies the data and is close to K. We can formulate the concept of almost satisfying the data in many equivalent ways since the data observations are finite. To be concrete, we shall use the weighted empirical ℓ_2 norm

$$\|v\| := \|v\|_{\ell_2(\mathbb{R}^m)} := \Big[\frac{1}{m}\sum_{j=1}^m |v_j|^2\Big]^{1/2}, \quad v \in \mathbb{R}^m. \tag{2.3}$$

Let us suppose that when given an $\varepsilon > 0$ we can find a $g_\varepsilon \in X$ for which

$$\|\lambda(g_\varepsilon) - w\| \leq \varepsilon \quad \text{and} \quad \operatorname{dist}(g_\varepsilon, K)_X \leq \varepsilon. \tag{2.4}$$

A numerical scheme may have the ability to drive ε to zero at the expense of higher levels of computation. The question arises as to *which level of accuracy ε would guarantee that g_ε provides a near optimal recovery. Equivalently, we would need g_ε to provide good approximation to the Chebyshev center z_w of K_w.* To formulate such a bound, we introduce the following *expandend Chebyshev radius*

$$R(K(w,\varepsilon))_X, \quad \text{where} \quad K(w,\varepsilon) := \bigcup_{w' \colon \|w'-w\| \leq \varepsilon} K_{w'}, \tag{2.5}$$

which is the Chebyshev radius of the inflated set $K(w,\varepsilon)$. Notice that $K_w \subset K(w,\varepsilon)$ for all $\varepsilon > 0$. We discuss properties of $R(K(w,\varepsilon))_X$ in more detail in the next subsection. The behavior of this expanded radius is important for deciding how much over parameterization is needed for near optimal recovery. For now, we prove the following lemma.

Lemma 2.2. *For any compact subset K of X and any $w \in \mathbb{R}^m$, we have*

$$\lim_{\varepsilon \to 0^+} R(K(w,\varepsilon))_X = R(K_w)_X. \tag{2.6}$$

Proof: Since the sets $K(w,\varepsilon)$, $\varepsilon > 0$, are nested, the function $R(K(w,\varepsilon))_X$, $\varepsilon > 0$, is decreasing as ε decreases. Hence, the limit in (2.6) exists. Suppose that this limit is R_0 and $R_0 > R(K_w)_X$. Then, for each $\varepsilon > 0$ there is an $f_\varepsilon \in K(w) \subset K(w,\varepsilon)$ with $\|f_\varepsilon - z_w\|_X \geq R_0$. The $f_{1/n}$, $n \geq 1$, come from the compact set K and hence there is a subsequence of them which converges to a limit f^* in K. This limit function satisfies $\|f^* - z_w\|_X \geq R_0$. We also know that $\lambda(f^*) = w$ and so $f^* \in K_w$. This means that $\|f^* - z_w\|_X \leq R(K_w)$. This contradicts $R_0 > R(K_w)$ and proves (2.6). □

Remark 2.3. *Let us record for further use that for any fixed $\gamma > 0$ and $w \in \mathbb{R}^m$, we have*

$$\lim_{\varepsilon \to 0^+} R(K(w,\gamma+\varepsilon))_X = R(K(w,\gamma))_X. \tag{2.7}$$

This is proved as in Lemma 2.2 by using the fact that the collection of sets $K(w,\gamma+\varepsilon)$, $\varepsilon > 0$, is a monotone family.

The following theorem gives a quantitative bound on the recovery performance of a constructed function g_ε in terms of how closely it fits the data and how close it is to the model class K.

Theorem 2.4. *If g_ε is any function in X satisfying* (2.4)*, then*

$$\|f - g_\varepsilon\|_X \leq \varepsilon + 2R(K(w,2\varepsilon))_X, \quad f \in K_w. \tag{2.8}$$

If $R(K_w) \neq 0$, then for any $C > 2$ and for ε suitably small the function g_ε is a near best recovery of f with constant C.

Proof: We know that there is an $h \in K$ such that $\|g_\varepsilon - h\|_X \leq \varepsilon$. This h satisfies

$$\|\lambda(h) - w\| \leq \|\lambda(h) - \lambda(g_\varepsilon)\| + \|\lambda(g_\varepsilon) - w\| \leq 2\varepsilon,$$

and thus $h \in K_{w'}$ where $\|w - w'\| \leq 2\varepsilon$. Hence, h is in the set $K(w,2\varepsilon)$. Now any $f \in K_w$ is also in $K(w,2\varepsilon)$. This means that $\|f - h\|_X \leq 2R(K(w,2\varepsilon))_X$. Since $\|g_\varepsilon - h\|_X \leq \varepsilon$, we obtain (2.8). If $C > 2$ and $\varepsilon > 0$ is sufficiently small, then $\varepsilon + 2R(K(w,2\varepsilon))_X$ is smaller than $CR(K_w)_X$ because of Lemma 2.2. □

2.2 The behavior of $R(K_w)_X$

The analysis that follows in this paper depends on the function $\epsilon \mapsto R(K(w,\epsilon))_X$ and so it may be useful to the reader to make a few remarks on this function. Its behavior depends very much on K and needs to be analyzed for each K individually. From the above estimates, we see that a critical issue in quantitative bounds for the performance of learning algorithms is the rate of convergence of $R(K(w,\varepsilon))_X$ to $R(K_w)_X$ as $\varepsilon \to 0^+$. It is easy to give examples of compact sets K for which $R(K(w,\varepsilon))_X$ tends to $R(K_w)_X$ arbitrarily slowly. Concerning the behavior of $R(K_w)_X$, let us note that this may not be a continuous function of w. To illustrate these issues, we consider the following simple example of a compact set in $\mathbb{R}^2$.

Example: *We define the compact set* $K := [0,1]^2 \cup \left([1,2] \times \{\frac{1}{2}\}\right) \subset X = \mathbb{R}^2$, *equipped with the Euclidean norm. We take the measurement functional* λ *to be the first coordinate of a point* $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$*:* $\lambda(\mathbf{x}) = x_1$.

For this example the function $R(K_w)_X$ is a discontinuous function of w (see the bottom-left graph in Figure 2.1). Now we consider the sets $K_\epsilon^1 := K(\tilde{w}, \epsilon)$ with $\tilde{w} \in (0,1)$ and $K_\epsilon^2 := K(\hat{w}, \epsilon)$ with $\hat{w} = 1.1$, pictured in Figure 2.1 (top-right), along with their corresponding Chebyshev balls. The graph of $R(K_\epsilon^2)_X$ as a function of ε is presented at the bottom-right. This function is a discontinuous function of ε with the discontinuity occuring at $\varepsilon = 0.1$. If we move the point $\hat{w}$ to be closer to 1, then the jump discontinuity in $R(K(\hat{w},\varepsilon))_X$ as a function of $\varepsilon > 0$ will move closer to 0. The main observation to make here is that the convergence of $R(K(\hat{w},\varepsilon))_X$, $\varepsilon \to 0^+$, towards $R(K_{\hat{w}})_X = 0$ is not uniform in $\hat{w}$ and depends on the distance of $\hat{w}$ to the square.

This example shows that obtaining quantitative bounds on the performance of numerical algorithms via the construction of a g_ε will be very much dependent on the set K and will therefore need its own ad hoc analysis.

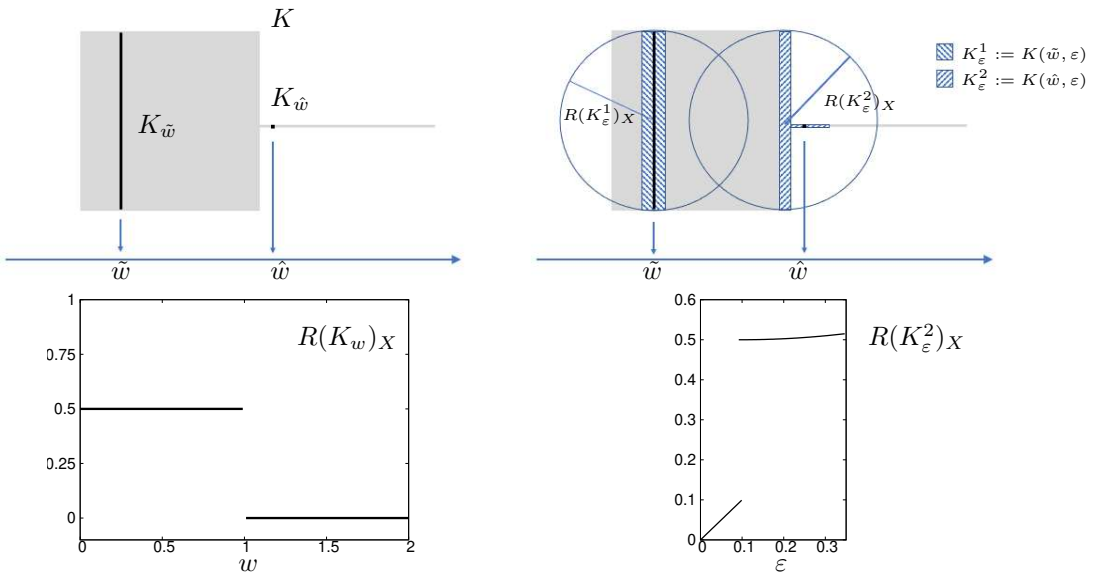

Figure 2.1: Top-Left: the set $K = [0,1]^2 \cup \left([1,2] \times \{\frac{1}{2}\}\right)$ and K_w for two different measurements, $\tilde{w} \in (0,1)$ and $\hat{w} = 1.1$; Bottom-Left: $R(K_w)_X$ as a function of w; Top-Right: the sets K_ε^1 and K_ε^2 and their corresponding Chebyshev balls; Bottom-Right: Graph of $R(K_\varepsilon^2)_X$ as a function of ε.

3 Near optimal recovery through discretization

The results of the previous section do not constitute a numerical learning algorithm. Rather, they only show that optimal performance can be obtained if an algorithm provides a function g_ε satisfying condition (2.4).

In this section, we begin our discussion of numerical learning algorithms by formulating discrete optimization problems whose successful numerical implementation would yield a near optimal numerical recovery algorithm. Thus, the problem of constructing near optimal numerical learning algorithms is reduced to questions centering around the convergence of optimization algorithms for the derived discrete optimization problem.

Any numerical algorithm for learning is based on some platform for approximation. The most common platforms use polynomials, splines, wavelets, or neural networks. Let Σ_n, $n = 1, 2, \ldots$, be the sets used for the approximation, where n denotes the complexity of Σ_n. The two main examples we have in mind are the cases where Σ_n is a linear subspace of X of dimension n and the case where Σ_n is a parametric nonlinear manifold of functions from X depending on Cn parameters. The most common example in the latter case is the nonlinear manifold consisting of the outputs of a neural network (NN) with n hidden neurons and some specified architecture and activation function (see [5] for an overview).

The first question we address is how we should use Σ_n to build a numerical algorithm. The answer depends heavily on the structure of K and is discussed in the sub-sections that follow.

3.1 Convex model classes

We begin with the most favorable case where K is a compact convex centrally symmetric (about the origin) subset of X. Any such set can be written as the unit ball $K = U(Y)$ of a normed linear subspace Y of X, where the norm on Y is induced by K (see e.g. [28]). Since K is compact, we have the embedding inequality

$$\|g\|_X \le C_0 \|g\|_Y, \quad g \in Y, \tag{3.1}$$

where C_0 is the embedding constant which depends only on K.

In this setting, we introduce the loss function

$$\mathcal{L}_\mu(g) := \|\lambda(g) - w\| + \mu \|g\|_Y, \tag{3.2}$$

for any $\mu > 0$. This loss function is defined for all $g \in X$ but is infinite if g is not in Y.

The following theorem describes a discrete optimization problem whose solution is a near optimal recovery.

Theorem 3.1. *Let $K = U(Y)$ with Y a normed linear subspace of X and let the set $\Sigma \subset X$ satisfy the condition*

$$\operatorname{dist}(K, \Sigma \cap K)_X < \delta. \tag{3.3}$$

Then, if $f \in K_w$, the function

$$\hat{f} := \hat{f}_{\Sigma,\mu} \in \operatorname*{argmin}_{g \in \Sigma} \mathcal{L}_\mu(g) \tag{3.4}$$

is a near optimal recovery of f, that is,

$$\|f - \hat{f}\|_X \le C R(K_w)_X, \tag{3.5}$$

for any $C > 2$, provided that $R(K_w)_X \neq 0$, $\delta \leq \mu^2$ and μ is sufficiently small. More precisely, it is sufficient that

$$\varepsilon := \mu \max(C_0, 1 + \mu) \tag{3.6}$$

satisfies the inequality

$$\varepsilon + 2R(K(w, 2\varepsilon))_X \leq CR(K_w)_X,$$

which is possible for $\mu > 0$ because of Lemma 2.2.

Proof: Let f be any function in $K = U(Y)$ which satisfies the data, i.e., f is in K_w, and let $S \in \Sigma \cap K$ satisfy $\|f - S\|_X \leq \delta$. From the definition of $\hat{f}$, we know that

$$\|w - \lambda(\hat{f})\| + \mu\|\hat{f}\|_Y \leq \|w - \lambda(S)\| + \mu\|S\|_Y \leq \delta + \mu, \tag{3.7}$$

where the first term was estimated by

$$\|w - \lambda(S)\| = \|\lambda(f - S)\| \leq \|f - S\|_X \leq \delta,$$

and the second term uses that $S \in K$ so that $\|S\|_Y \leq 1$.

We now assume that $\delta \leq \mu^2$ and μ is small. We see from (3.7) that $\hat{f}$ almost satisfies the data since

$$\|w - \lambda(\hat{f})\| \leq \delta + \mu \leq \mu(1 + \mu).$$

Also, $\hat{f}$ is close to K since (3.7) shows that $\|\hat{f}\|_Y \leq 1 + \mu$. Therefore $(1 + \mu)^{-1}\hat{f} \in K$, and from (3.1) we have

$$\operatorname{dist}(\hat{f}, K)_X \leq \|\hat{f} - (1 + \mu)^{-1}\hat{f}\|_X = \frac{\mu}{1 + \mu}\|\hat{f}\|_X \leq \frac{\mu}{1 + \mu}C_0\|\hat{f}\|_Y \leq \mu C_0.$$

In other words, $g = \hat{f}$ satisfies (2.4) for $\varepsilon := \mu \max(C_0, 1 + \mu)$. Theorem 2.4 shows that for any $C > 2$, the function $\hat{f}$ is a near optimal recovery with constant C, provided μ (and hence δ) is sufficiently small. The last statement of the theorem follows from Theorem 2.4. □

Remark 3.2. *In practice, numerical optimizers may not find a global minimizer $\hat{f}$ in* (3.4)*. Rather, they are more likely to produce $\tilde{f} \in \Sigma$ such that*

$$\mathcal{L}_\mu(\tilde{f}) \leq \mathcal{L}_\mu(\hat{f}) + \tilde{\varepsilon}$$

for some $\tilde{\varepsilon} > 0$. In this case, the conclusion of Theorem 3.1 remains valid provided $\tilde{\varepsilon}$ is sufficiently small; namely $\tilde{\varepsilon} \leq \delta \leq \frac{1}{2}\mu^2$. Indeed, the estimate (3.7) *gives*

$$\|w - \lambda(\tilde{f})\| + \mu\|\tilde{f}\|_Y \leq \delta + \mu + \tilde{\varepsilon} \leq 2\delta + \mu \leq \mu(1 + \mu),$$

and the proof is completed as in the theorem.

Remark 3.3. *Notice that if Σ is convex (e.g. if Σ is a linear space) and if $\|\cdot\|_Y$ is strictly convex, then the minimizer of $\mathcal{L}_\mu$ over Σ is unique because $\mathcal{L}_\mu$ is always strictly convex on Σ.*

3.2 General model classes

We next want to give a discretization problem whose solution is near optimal for any model class K, i.e., for any compact set $K \subset X$. For this, we introduce the loss function

$$\mathcal{L}_K(g) := \|\lambda(g) - w\| + \operatorname{dist}(g, K)_X, \quad g \in X. \tag{3.8}$$

The following theorem holds.

Theorem 3.4. *Let K be any compact subset of X and let the set $\Sigma \subset X$ satisfy the condition*

$$\operatorname{dist}(K, \Sigma)_X < \delta. \tag{3.9}$$

Then, if $f \in K_w$, the function

$$\hat{f} \in \operatorname*{argmin}_{g \in \Sigma} \mathcal{L}_K(g) \tag{3.10}$$

is a near optimal recovery f, that is, we have for any $C > 2$

$$\|f - \hat{f}\|_X \le C R(K_w)_X, \tag{3.11}$$

provided that $R(K_w)_X \neq 0$ and δ is sufficiently small. More precisely, it is sufficient that

$$2\delta + 2R(K(w, 4\delta))_X \le C R(K_w)_X,$$

which is possible because of Lemma 2.2.

Proof: Let f be any function in K which satisfies the data, i.e., f is in K_w, and let $S \in \Sigma$ satisfy $\|f - S\|_X \le \delta$. From the definition of $\hat{f}$, we know that

$$\|w - \lambda(\hat{f})\| + \operatorname{dist}(\hat{f}, K)_X \le \|w - \lambda(S)\| + \operatorname{dist}(S, K)_X \le \delta + \delta = 2\delta, \tag{3.12}$$

where the first term was estimated by

$$\|w - \lambda(S)\| = \|\lambda(f - S)\| \le \|f - S\|_X \le \delta.$$

It follows that the function $g_\varepsilon := \hat{f}$ satisfies (2.4) with $\varepsilon = 2\delta$. From Theorem 2.4, we find that

$$\|f - \hat{f}\|_X \le 2\delta + 2R(K(w, 4\delta))_X, \quad f \in K_w. \tag{3.13}$$

If $C > 2$ and δ is sufficiently small, Lemma 2.2 gives that the right side of (3.13) does not exceed $C R(K_w)_X$. □

The difference between the case of a general model class K and the special case where K is convex and centrally symmetric is in the form of the penalty term in the loss function. Ostensibly, the penalty in the general case would be more difficult to numerically implement. Also note that the penalty term does not require a parameter to balance it with the data fitting term. This is because we simply want both terms to be small simultaneously.

Remark 3.5. *Let us emphasize that the results in this section do not yet give a numerical algorithm for near optimal recovery since we have not given a numerical recipe for solving the corresponding discrete problem. This is discussed in more details in §8.*

Remark 3.6. *Similarly to Remark 3.2, if we only approximately solve the minimization problem we still obtain a near optimal recovery provided the numerical error is small enough.*

4 The special case of point values and recovery in L_p

We turn next to what is the most common setting in machine learning where the data comes from point evaluations. We assume that f is a function defined on $\Omega \subset \mathbb{R}^d$, $d \geq 1$, where Ω is the closure of a bounded domain in $\mathbb{R}^d$. For the moment, we assume that we have noiseless observations

$$w_i = f(x_i), \quad i = 1, \ldots, m, \tag{4.1}$$

where the *data sites* x_i come from Ω. The most common choice of the metric in which to measure recovery error is an $X = L_p(\Omega, \nu)$ norm with ν a measure on Ω and $1 \leq p \leq \infty$. Since point evaluation is not a linear functional on X, we cannot apply the results of the previous section. Note however, that to define the $\mathcal{L}_\mu$ or $\mathcal{L}_K$, it is enough to have point evaluation well defined for functions in the model class K and functions in Σ.

In order to guarantee that point evaluation is well defined for functions in K, we make the following assumption on the model class K.

Main Assumption: *We assume the model class K is a compact subset of $C(\Omega)$. So, in particular, all functions in K have well defined point values.*

Even though we impose this assumption on K, we continue to measure the performance of the learning algorithm in a Banach space X for which we have an embedding

$$\|f\|_X \leq C_X \|f\|_{C(\Omega)}. \tag{4.2}$$

Notice that such an embedding holds for $L_p(\Omega, \nu)$ spaces, $1 \leq p < \infty$.

Let K be a model class satisfying our **Main Assumption**. As in the previous section, we shall consider two settings depending on whether K is convex and centrally symmetric, or K is a general compact set. The results we give in this section are similar to those of the preceding section with the modifications necessary to handle the new setting of point evaluation.

As before, let K_w be the set of all $f \in K$ which satisfy the given measurements, i.e. (4.1). As in the previous section, the optimal recovery for a model class K with the data (4.1) is given by the Chebyshev center of K_w and the optimal error is the Chebyshev radius $R(K_w)_X$. This radius will depend on X.

Now, let us see what the previous section says since our setting is slightly different. As before, we let

$$K(w, \varepsilon) := \bigcup_{\|w' - w\| \leq \varepsilon} K_{w'}, \tag{4.3}$$

and let $R(K(w, \varepsilon))_X$ be the Chebyshev radius of this inflated set in X.

The following are the analogues of Theorem 2.4 and Lemma 2.2. We use the notation

$$\lambda_{\mathbf{x}}(g) := (g(x_1), \ldots, g(x_m)) \in \mathbb{R}^m, \quad \mathbf{x} := (x_1, \ldots, x_m) \subset \Omega, \quad g \in C(\Omega), \tag{4.4}$$

when discussing point evaluation data. Note that when $g, h \in C(\Omega)$, it follows from definition (2.3) that

$$\|\lambda_{\mathbf{x}}(g) - \lambda_{\mathbf{x}}(h)\| \leq \|h - g\|_{C(\Omega)}.$$

Proposition 4.1. *Let X satisfy the embedding* (4.2) *and let K satisfy our* **Main Assumption**. *If $f \in K_w$, where $w = \lambda_{\mathbf{x}}(f)$, and g_ε is any function in $C(\Omega)$ with $\|w - \lambda_{\mathbf{x}}(g_\varepsilon)\| \leq \varepsilon$ and $\operatorname{dist}(g_\varepsilon, K)_{C(\Omega)} \leq \varepsilon$, then*

$$\|f - g_\varepsilon\|_X \leq C_X \varepsilon + 2R(K(w, 2\varepsilon))_X. \tag{4.5}$$

Proof: The proof is the same as that of Theorem 2.4 after choosing h in K to satisfy the inequality $\|g_\varepsilon - h\|_{C(\Omega)} \le \varepsilon$. □

Remark 4.2. *In the last proposition, it is natural to ask why we need g_ε close to K in the norm of $C(\Omega)$ and not just in the norm of X. The following example clarifies that issue. Let $\Omega = [0,1]$ and $X = L_2[0,1]$. Consider a set $\mathbf{x}$ of data sites. Let $K = \{f_1, f_2\}$ where $f_1 \equiv 1$ and $f_2 \equiv 0$. Let $w = (1,1,\dots,1)$ which is data satisfied only by $f = f_1$. If g_ε is one at the data and $\|g_\varepsilon\|_{L_2[0,1]} < \varepsilon$, then g_ε satisfies the data and is close to K in the X norm. However, the left side of (4.5)is close to 1 and the right side is close to 0, so the Proposition using distance in X is not valid.*

Lemma 4.3. *If X satisfies (4.2) and K satisfies our* **Main Assumption***, then, we have*

$$\lim_{\varepsilon \to 0^+} R(K(w,\varepsilon))_X = R(K_w)_X. \tag{4.6}$$

Proof: The proof is similar to that of Lemma 2.2 and so we only indicate the small differences. The limit R_0 on the left in (4.6) exists from monotonicity. If $R_0 > R(K_w)_X$, then because of the compactness of K in $C(\Omega)$ there is a sequence $\varepsilon_n \to 0^+$ and a sequence $f_n \in K(w,\varepsilon_n)_X$ that converges to a limit f^* in K in the $C(\Omega)$ norm, and because of (4.2), in the X norm. Then, $\|f^* - z_w\|_X \ge R_0$. From the convergence in $C(\Omega)$, it follows that f^* is in K_w and so we have $\|f^* - z_w\|_X \le R(K_w)_X$, which contradicts that $\|f^* - z_w\|_X \ge R_0$ and $R_0 > R(K_w)$. □

Let us now assume that $K = U(Y)$, where Y is a subspace of $C(\Omega)$ equipped with a norm $\|\cdot\|_Y$. Typical examples for Y are smoothness spaces: Lipschitz, Sobolev, Besov spaces. The **Main Assumption** is simply requiring that Y compactly embeds into $C(\Omega)$. Therefore, we know that

$$\|f\|_{C(\Omega)} \le C_Y, \quad f \in K.$$

We shall use this inequality as we proceed without mentioning it. Such embeddings typically follow from Sobolev embedding theorems.

The following theorem describes a discrete optimization problem whose solution is a near optimal recovery. In the statement of this theorem we use the loss function $\mathcal{L}_\mu$ as given in (3.2) using point evaluation functionals (see (4.4)).

Theorem 4.4. *Let $K = U(Y)$ with Y a normed linear subspace of $C(\Omega)$ satisfy the* **Main Assumption***, let X satisfy the embedding (4.2), and let the set Σ satisfy the condition*

$$\operatorname{dist}(K, \Sigma \cap K)_{C(\Omega)} < \delta. \tag{4.7}$$

If $f \in K_w$, where $w = \lambda_{\mathbf{x}}(f)$, then the function

$$\hat{f} := \hat{f}_{\Sigma,\mu} \in \operatorname*{argmin}_{g \in \Sigma} \mathcal{L}_\mu(g), \quad \text{where} \quad \mathcal{L}_\mu(g) := \|\lambda_{\mathbf{x}}(g) - w\| + \mu \|g\|_Y, \tag{4.8}$$

is a near optimal recovery of f, that is,

$$\|f - \hat{f}\|_X \le C R(K_w)_X, \tag{4.9}$$

for any $C > 2$, provided that $R(K_w)_X \ne 0$, and δ and μ are sufficiently small. More precisely, it is enough to choose $\delta \le \mu^2$ and μ small enough so that $C_X \varepsilon + 2R(K(w,2\varepsilon))_X \le C R(K_w)_X$ with $\varepsilon := \mu \max(\mu + 1, C_Y)$.

Proof: The proof is the same as that of Theorem 3.1 except that now we choose $S \in \Sigma \cap K$ to satisfy $\|f - S\|_{C(\Omega)} \leq \delta$ and use Proposition 4.1 and Lemma 4.3. □

We also have the analogue to Theorem 3.4.

Theorem 4.5. *Let K satisfy the* **Main Assumption**, *let X satisfy the embedding* (4.2), *and let the set Σ satisfy the condition*

$$\operatorname{dist}(K, \Sigma)_{C(\Omega)} < \delta. \tag{4.10}$$

If $f \in K_w$, where $w = \lambda_{\mathbf{x}}(f)$, then the function

$$\hat{f} := \hat{f}_\Sigma \in \operatorname*{argmin}_{g \in \Sigma} \mathcal{L}_K(g), \quad \textit{where} \quad \mathcal{L}'_K(g) := \|\lambda_{\mathbf{x}}(g) - w\| + \operatorname{dist}(g, K)_{C(\Omega)}, \tag{4.11}$$

is a near optimal recovery of f, that is

$$\|f - \hat{f}\|_X \leq C R(K_w)_X, \tag{4.12}$$

for any $C > 2$, provided $R(K_w)_X \neq 0$ and δ is sufficiently small. More precisely, it is sufficient that

$$C_X \varepsilon + 2R(K(w, 2\varepsilon)_X \leq C R(K_w)_X, \quad \textit{where} \quad \varepsilon := 2\delta.$$

Proof: The proof is the same as that of Theorem 3.4 except that now we choose $S \in \Sigma$ to satisfy $\|f - S\|_{C(\Omega)} \leq \delta$ and use Proposition 4.1 and Lemma 4.3. □

5 Noisy measurements

In this section, we consider the case when the measurements are corrupted by an additive deterministic noise. Namely, we assume that our measurements are now given by

$$\tilde{w}_j = f(x_j) + \eta_j, \quad j = 1, \ldots, m, \tag{5.1}$$

where the real numbers η_j, $j = 1, \ldots, m$, are unknown to us. However, to derive quantitative results on performance, we will have to make some assumptions on the unknown noise vector $\boldsymbol{\eta} := (\eta_1, \ldots, \eta_m)$. We assume that all we know about $\boldsymbol{\eta}$ is its size.

We continue to let $w_j = f(x_j)$, $j = 1, \ldots, m$, and $w := (w_1, \ldots, w_m)$. We put ourselves in the same setting as in the previous section where $f \in K$ and K satisfies our **Main Assumption**. We let K_w again be the set of $f \in K$ such that $f(x_j) = w_j$, $j = 1, \ldots, m$, and continue to use the inflated sets $K(w, \varepsilon)$, $\varepsilon > 0$.

We assume that we have a bound on the noise vectors of the form

$$\|\boldsymbol{\eta}\| \leq \gamma < \infty. \tag{5.2}$$

Then, the totality of information we have about f is that $f \in K$ and f satisfies the data $\tilde{w} - \boldsymbol{\eta}$ where $\tilde{w}$ is our observation vector and $\|\eta\| \leq \gamma$. It follows that the totality of information we have about f is that it is in the set $K(\tilde{w}, \gamma)$. This means that the error of optimal recovery of f from such noisy observations is given by

$$\textbf{best noisy rate} = R(K(\tilde{w}, \gamma))_X. \tag{5.3}$$

We formulate recovery results for any set K that satisfies our **Main Assumption**. For any function $g \in C(\Omega)$, we continue to use the notation $\lambda_{\mathbf{x}}(g) := (g(x_1), \dots, g(x_m))$, where $\mathbf{x} = (x_1, \dots, x_m) \in \Omega$. To recover f from the noisy observations $\tilde{w}$, we use the loss function

$$\mathcal{L}_{K,\tau}(g) := \tau \|\tilde{w} - \lambda_{\mathbf{x}}(g)\| + \operatorname{dist}(g, K)_{C(\Omega)}, \tag{5.4}$$

where the parameter τ is a positive real number.

Theorem 5.1. *Let K satisfy the* **Main Assumption**, *let X satisfy the embedding* (4.2), *and let the set Σ satisfy the condition*

$$\operatorname{dist}(K, \Sigma)_{C(\Omega)} < \delta. \tag{5.5}$$

Consider the function

$$\hat{f} := \hat{f}_{\Sigma,\tau} \in \operatorname*{argmin}_{g \in \Sigma} \mathcal{L}_{K,\tau}(g), \quad \textit{where} \quad \mathcal{L}_{K,\tau}(g) := \tau \|\tilde{w} - \lambda_{\mathbf{x}}(g)\| + \operatorname{dist}(g, K)_{C(\Omega)}, \tag{5.6}$$

where $\tilde{w}$ are the noisy data observations of a function $f \in K$ with an unknown noise vector $\boldsymbol{\eta}$ which satisfies $\|\boldsymbol{\eta}\| \le \gamma$ for some finite number $\gamma \le 1$ considered unknown. Then, $\hat{f}$ is a near optimal recovery of f, that is,

$$\|f - \hat{f}\|_X \le C R(K(\tilde{w}, \gamma))_X, \tag{5.7}$$

for any $C > 2$, provided $\delta \le \tau^2$ and τ is sufficiently small.

Proof: We assume $\tau < 1$ and let $f \in K_w$ and $\tilde{w}$ be the noisy observations of f with noise vector $\boldsymbol{\eta}$ satisfying $\|\boldsymbol{\eta}\| \le \gamma$. Let $S \in \Sigma$ satisfy $\|f - S\|_{C(\Omega)} \le \delta$. Then, we know that $\|\lambda_{\mathbf{x}}(S) - w\| \le \delta$ and so $\|\lambda_{\mathbf{x}}(S) - \tilde{w}\| \le \delta + \gamma$. Since $S \in \Sigma$, we have

$$\tau \|\tilde{w} - \lambda_{\mathbf{x}}(\hat{f})\| + \operatorname{dist}(\hat{f}, K)_{C(\Omega)} \le \tau \|\tilde{w} - \lambda_{\mathbf{x}}(S)\| + \operatorname{dist}(S, K)_{C(\Omega)} \le \tau(\delta + \gamma) + \delta < \tau\gamma + 2\delta.$$

It follows that for $\delta \le \tau^2$ we have

$$\|\tilde{w} - \lambda_{\mathbf{x}}(\hat{f})\| < \gamma + 2\frac{\delta}{\tau} \le \gamma + 2\tau \quad \text{and} \quad \operatorname{dist}(\hat{f}, K)_{C(\Omega)} \le \tau\gamma + 2\delta. \tag{5.8}$$

Now let $h \in K$ satisfy

$$\|\hat{f} - h\|_{C(\Omega)} \le \tau\gamma + 2\delta \le \tau\gamma + 2\tau^2. \tag{5.9}$$

Then, we have

$$\begin{aligned} \|\tilde{w} - \lambda_{\mathbf{x}}(h)\| &\le \|\tilde{w} - \lambda_{\mathbf{x}}(\hat{f})\| + \|\lambda_{\mathbf{x}}(\hat{f}) - \lambda_{\mathbf{x}}(h)\| \le \gamma + 2\tau + \|\hat{f} - h\|_{C(\Omega)} \\ &\le \gamma + 2\tau + \tau\gamma + 2\tau^2 < \gamma + \tau(4 + \gamma) \le \gamma + 5\tau. \end{aligned}$$

span

So, $h \in K(\tilde{w}, \gamma + 5\tau)$ and so is f. We therefore obtain

$$\|f - h\|_X \le 2R(K(\tilde{w}, \gamma + 5\tau))_X.$$

Finally, from (5.9) and the embedding inequality (4.2) we have

$$\|f - \hat{f}\|_X \le \|f - h\|_X + \|h - \hat{f}\|_X \le 2R(K(\tilde{w}, \gamma + 5\tau))_X + C_X(\tau\gamma + 2\tau^2). \tag{5.10}$$

If we take τ suitably small we obtain (5.7) because of (2.7). □

Remark 5.2. *The appearance of the parameter τ in the case of noisy observations is quite natural since the confidence in the measurements decreases as the noise level increases. When there is no noise, we have complete confidence in the measurements and so we can take $\tau = 1$ as was done in the previous section.*

Remark 5.3. *Note that in the above, it is not necessary to know either γ or τ in order to arrive at the inequality* (5.10)*. However, to guarantee that the recovery is near optimal, one needs to choose τ (and hence δ) sufficiently small depending on the nature of K.*

Remark 5.4. *The most common setting for noise in statistics is to assume that the noise vector is composed of independent random draws with respect to an underlying probability distribution. Optimal performance in such a setting is referred to as minimax rates. We do not treat this case in this paper since it requires some substantially new ideas.*

6 Variants

This section considers variants of the minimization problems already discussed and emphasises certain aspects of these problems that are useful in numerical implementation. Since the treatment of the other cases is similar, we concentrate on the model assumption $f \in K = U(Y)$ and the loss function $\mathcal{L}_\mu$ given in (3.2). An alternative to $\mathcal{L}_\mu$ is the loss function

$$\mathcal{L}'_\mu(g) := \|w - \lambda(g)\|^\alpha + \mu \|g\|_Y^\beta, \tag{6.1}$$

where $\alpha > 0$ and $\beta > 0$ are fixed. Following the ideas from Subsection 3.1, we establish similar near optimality results for this loss function.

Theorem 6.1. *Let $K = U(Y)$ with Y a normed linear subspace of X and let the set Σ satisfy* (3.3)*. Then, for any $C > 2$, the function*

$$\hat{f} := \hat{f}_{\Sigma,\mu} \in \operatorname*{argmin}_{g \in \Sigma} \mathcal{L}'_\mu(g) \tag{6.2}$$

is a near optimal recovery, i.e.,

$$\|f - \hat{f}\|_X \le C R(K_w)_X, \quad f \in K_w, \tag{6.3}$$

provided $R(K_w)_X \neq 0$, $\delta^\alpha \le \mu^2$ and μ is sufficiently small. Moreover, in the case of numerical optimization producing $\tilde{f}$ such that $\mathcal{L}'_\mu(\tilde{f}) \le \mathcal{L}'_\mu(\hat{f}) + \epsilon$ for some $\epsilon > 0$, the estimate (6.3) *remains valid with $\hat{f}$ replaced by $\tilde{f}$ and provided that $\epsilon \le \delta^\alpha \le \frac{1}{2}\mu^2$ and μ is sufficiently small.*

Proof: Let f be any function in $K = U(Y)$ which satisfies the data, i.e., f is in K_w, and let $S \in \Sigma \cap K$ satisfy $\|f - S\|_X \le \delta$. From the definition of $\hat{f}$, we know that

$$\|w - \lambda(\hat{f})\|^\alpha + \mu \|\hat{f}\|_Y^\beta \le \|w - \lambda(S)\|^\alpha + \mu \|S\|_Y^\beta \le \delta^\alpha + \mu, \tag{6.4}$$

where the first term was estimated by

$$\|w - \lambda(S)\| = \|\lambda(f - S)\| \le \|f - S\|_X \le \delta,$$

and the second term uses that $S \in K$ so that $\|S\|_Y \leq 1$.

We now assume that $\delta^\alpha \leq \mu^2$ and μ small. We see from (6.4) that $\hat{f}$ almost satisfies the data since

$$\|w - \lambda(\hat{f})\|^\alpha \leq \delta^\alpha + \mu \leq \mu(1+\mu).$$

Also, $\hat{f}$ is close to K since (6.4) shows that $\|\hat{f}\|_Y^\beta \leq 1 + \mu$ and so $(1+\mu)^{-\frac{1}{\beta}}\hat{f} \in K$ and from (3.1) we have

$$\|\hat{f} - (1+\mu)^{-\frac{1}{\beta}}\hat{f}\|_X = (1-(1+\mu)^{-\frac{1}{\beta}})\|\hat{f}\|_X \leq (1-(1+\mu)^{-\frac{1}{\beta}})C_0\|\hat{f}\|_Y \leq ((1+\mu)^{\frac{1}{\beta}} - 1)C_0.$$

This means that $g = \hat{f}$ satisfies (2.4) for $\varepsilon := \max\left(C_0((1+\mu)^{\frac{1}{\beta}} - 1), (\mu^2+\mu)^{\frac{1}{\alpha}}\right)$. Theorem 2.4 shows that for any $C > 2$, the function $\hat{f}$ is a near optimal recovery with constant C provided μ (and hence δ) is sufficiently small.

In the case of a numerical approximation $\tilde{f}$ to $\hat{f}$, the estimate (6.4) gives

$$\|w - \lambda(\tilde{f})\|^\alpha + \mu\|\tilde{f}\|_Y \leq \delta^\alpha + \mu + \epsilon \leq 2\delta^\alpha + \mu \leq \mu^2 + \mu,$$

and the proof is completed as above. □

Remark 6.2. *If Σ is convex (e.g. if it is a finite dimensional linear space), if $\|.\|_Y$ is strictly convex, and if $\alpha, \beta \geq 1$, then the minimizer of $\mathcal{L}'_\mu(g)$ over $g \in \Sigma$ is unique. Non-uniqueness can occur for other settings, for example when the second term is a quasi-norm or some other nonconvex regularizer. The latter are sometimes preferred due to their better performance in special cases.*

7 Sampling rates

Although this is not the main topic of this paper, an important issue in learning is how many samples m are needed to guarantee that an $f \in K$ can be learned with a prescribed accuracy. In this section, we mention three concepts that give a benchmark for the accuracy issue. We refer to these concepts in the next section where we discuss what our results say in two common settings for model classes in learning.

So far, we have discussed learning primarily from the viewpoint that we were given data and wish to recover the function f which gave rise to this data. In that setting, we had no role in the choice of the data sites. A natural question is if we are given a budget m of samples we can take of a function $f \in K$, what would be the best choice of data sites. Historically, there are three concepts that address this issue: Gelfand widths, sampling numbers, and averaged sampling numbers. We briefly introduce these notions in this section.

7.1 Gelfand widths

Suppose that K is a compact set in the Banach space X and we are allowed to use our knowledge of K to introduce m sampling functionals $\lambda_1, \ldots, \lambda_m$ to use in sampling the elements of K. Which functionals should we choose and what is the accuracy at which we could recover any $f \in K$ from the data $\lambda_1(f), \ldots, \lambda_m(f)$? The Gelfand width

$$d^m(K)_X := \inf_{\lambda_1,\ldots,\lambda_m \in X^*} \sup_{f \in K} R(K_{\lambda(f)})_X, \quad \text{where} \quad \lambda(f) := (\lambda_1(f), \ldots, \lambda_m(f)), \tag{7.1}$$

is the optimal accuracy we can achieve in the worst case sense.

The Gelfand widths of model classes K is a well studied concept in Functional Analysis and Approximation Theory (see e.g. the book of Pinkus [23]). The Gelfand widths of classical model classes K in classical Banach spaces X are for the most part known and the Gelfand widths of novel model classes proposed in modern learning are currently being investigated (see e.g. [24, 22]). Let us also note that Gelfand widths were the origins of compressed sensing which studies the encoding and decoding of signals f from a model class K described by sparsity. There it is shown that a random choice of $\lambda_1, \ldots, \lambda_m \in X^*$ is with high probability near optimal (see [7, 3] and the many books written on compressed sensing such as [9]).

For us, the Gelfand width $d^m(K)_X$ gives a lower bound for the accuracy with which we can recover a general $f \in K$ from linear measurements of f. A general criticism of Gelfand widths is that it is too general for practical applications of sampling. For this reason, one typically imposes restrictions on the functionals λ_j, $j = 1, \ldots, m$. If one requires that the sampling is done via point evaluation of f, then this leads to the concept of sampling numbers.

7.2 Sampling numbers

Let K be a subset of $C(\Omega)$ with Ω the closure of a bounded domain in $\mathbb{R}^d$. If we restrict the linear functionals used as data observations to be point values of f, then the optimal performance of m such samples of f is given by

$$s_m(K)_X := \inf_{x_1,\ldots,x_m \in \Omega} \sup_{f \in K} R(K_{f(\mathbf{x})})_X, \quad f(\mathbf{x}) := (f(x_1), \ldots, f(x_m)), \quad m = 1, 2, \ldots. \tag{7.2}$$

The points $x_1, \ldots, x_m$ that give the infimum in $s_m(K)_X$ are the optimal sampling sites. We obviously have $s_m(f)_X \geq d^m(K)_X$ and the difference in these two numbers is often substantial. Sampling numbers are well studied for classical model classes K in classical Banach spaces X, especially in the Information Based Complexity community, where it is referred to as standard information. However, for novel model classes of functions of many variables that arise in modern learning there are many open questions on the asymptotic decay of the sampling numbers as $m \to \infty$.

7.3 Average sampling

It is sometimes difficult to determine the sampling numbers of a model class K and even more so the position of the optimal data sites. In this case, one studies the expected performance when the data sites $\mathbf{x}$ are chosen randomly with respect to a probability measure on Ω. The relevant measure of performance is the averaged sampling numbers given by

$$\bar{s}_m(K)_X := \sup_{f \in K} \mathrm{Exp}[R(K_{f(\mathbf{x})})_X]. \tag{7.3}$$

8 Examples

The main objective of this paper is to describe the optimal performance that is possible for a learning algorithm and to show that this optimal performance can be achieved by solving an over-parameterized optimization problem. In this sense, we provided a justification for the use of over-parameterized optimization which is now a common staple in machine learning. Exactly how

this plays out in practice depends very much on the model class K which gives the properties of the function f to be learned.

Two natural questions arise in the numerical implementation of this theory. The first is how fine must we take Σ_n and how to chose the parameters in the loss function in order to guarantee near optimal learning. The second question is to describe a numerical method with convergence guarantees for solving the resulting discrete optimization problem.

We know from the exposition given above that when given a model class K and linear data observations of an $f \in K$ that the optimal accuracy in recovering f from these observations is $R(K_w)_X$ and that a near optimal recovery is given by solving a discrete over-parameterized optimization problem. The amount of over-parameterization necessary depends on $R(K(w,\varepsilon))_X$ and how fast it converges to $R(K_w)_X$ as $\varepsilon \to 0^+$. This in turn depends very much on the particular K and requires an ad hoc analysis depending on K. In order to illustrate what is involved in such an analysis and what is known, we discuss two examples in this section. There are numerous other examples that could be considered and would be relevant to what is done in current practice of machine learning.

8.1 Point values of a smooth function

A traditional setting in learning is to consider the data to be point evaluations of a function f defined on a domain $\Omega \subset \mathbb{R}^d$ and to measure the error of recovering f in an $L_q(\Omega)$ norm, $1 \le q \le \infty$. This is an extensively studied setting in IBC. The texts [27, 20] are general references for this case. Our goal in this section is to shine a light on what the results of the present paper have to say about optimal learning in this setting. For simplicity of discussion, we assume $\Omega := [0,1]^d$ and $q = 2$; the extension to $q \neq 2$ and more general domains can be found for example in [13] and the references in that paper.

For our model classes, we consider the unit ball $K := U(W^s(L_p(\Omega)))$, $s > 0$, $1 < p \le \infty$, of the Sobolev space $W^s(L_p(\Omega))$. In order to have K a compact subset of $C(\Omega)$, we assume $s > d/p$. The results mentioned in this section generalize to the case when Ω is a bounded Lipschitz domain and the Sobolev space is replaced by a more general Besov space as long as we continue to have a compact embedding into $C(\Omega)$.

Let $x_j \in \Omega$, $j = 1, \dots, m$, be m data sites and $w_j = f(x_j)$, $j = 1, \dots, m$, be data observations of an $f \in K$. We take these measurements to be exact; noisy measurements can be treated as discussed in §5. We use our notation $\mathbf{x} = (x_1, \dots, x_m)$ for the data sites.

The optimal recovery error $R(K_w)_X$, $X = L_2(\Omega)$, depends on the position of the data sites as is described for example in [19, 14]. It is known that near optimal sampling sites $\mathbf{x}$ are those that are uniformly spaced and the optimal recovery error $R(K_w)_{L_2(\Omega)}$ in the case of uniform spacing is $\approx m^{-s/d+(1/p-1/2)_+}$. For more general positioning of the point $\mathbf{x}$, the optimal recovery rate is also known and depends on the maximal distance between the points of $\mathbf{x}$ (see [19]). Additionally, it is known that m random sample sites are near optimal save for a possible logarithm [13]. Algorithms for near optimal recovery are known using quasi-interpolants (see [13]).

Our results show that a near optimal recovery can be obtained by choosing a sufficiently fine linear or nonlinear space $\Sigma = \Sigma_n$, and solving the penalized least squares problem

$$\hat{f}'_\Sigma := \operatorname*{argmin}_{S \in \Sigma} \left[\left[\frac{1}{m} \sum_{j=1}^m [w_j - S(x_j)]^2 \right]^{1/2} + \mu \|S\|_{W^s(L_p(\Omega))} \right], \tag{8.1}$$

with μ chosen sufficiently small. According to §6, we may also obtain near optimal performance by using the modified loss

$$\hat{f}_\Sigma := \operatorname*{argmin}_{S\in\Sigma} \left[\frac{1}{m}\sum_{j=1}^{m} [w_j - S(x_j)]^2 + \mu \|S\|^p_{W^s(L_p(\Omega))} \right], \tag{8.2}$$

with μ chosen sufficiently small. This latter loss is convenient for numerical implementation as discussed below. There are several natural choices for Σ such as a linear FEM space or a linear space spanned by B-splines or wavelets.

To describe in a bit more detail one simple example, we consider the univariate case $d = 1$ and $K = U(W^1(L_p[0,1]))$, $1 < p \le \infty$. For convenience, we assume that the endpoints $0, 1$ are always data sites. It is easy to see that if $K_w \neq \emptyset$, then the continuous piecewise linear function S_w which interpolates the data and has breakpoints only at the data sites $\mathbf{x}$ is in K_w and hence is a near optimal recovery. In other words, in this special setting, there is a simple near optimal learning algorithm given by piecewise linear interpolation of the data. The optimal recovery rate in this case is $O(h(\mathbf{x})^s)$, where

$$s := 1 - (1/p - 1/2)_+, \tag{8.3}$$

and $h(\mathbf{x}) := \max_{1\le j<m} |x_j - x_{j+1}|$ is the maximal separation between the data sites.

Our results show that there is a general principle based on over-parameterized learning which always leads to a near optimal algorithm. For Σ one can choose any sufficiently fine linear or nonlinear space Σ_n, say of dimension n, and then solve the minimization problem (8.2). We want to see how large one must choose Σ_n and how small one must choose μ in this case.

To give a specific construction, we take $\Sigma = \Sigma_n$ as the linear space of all piecewise linear functions with breakpoints at $\xi_j := j/n$, $j = 0, \dots, n$. Then,

$$\operatorname{dist}(K, \Sigma_n \cap K)_{L_2[0,1]} \le n^{-s}, \tag{8.4}$$

where s is defined by (8.3). As we have already noted, we know that

$$R(K_w)_{L_2[0,1]} = O(h(\mathbf{x})^s). \tag{8.5}$$

Another simple calculation shows that

$$R(K(w,\varepsilon))_{L_2[0,1]} \le C[\varepsilon + h(\mathbf{x})^s]. \tag{8.6}$$

If we now consider the loss (8.1) and look at Theorem 4.4, it says that it is enough to have $\varepsilon \le m^{-s}$ in that theorem and the parameter $\mu \le \varepsilon$ and the approximation accuracy δ (as measured in $C(\Omega)$) to be satisfy $\delta \le \mu^2$. Since the approximation accuracy in $C(\Omega)$ is $O(n^{-1+1/p})$, we see that it is enough to take

$$n \ge m^{\frac{2s}{1-1/p}}$$

and $\mu \le m^{-s}$ in the minimization problem (8.1) to find a near optimal recovery of f. A similar analysis can be made when using the loss function (8.2). This example illustrates when given a compact set K, how one determines how well Σ_n must approximate K and how we must choose μ so that solving the minimization problem gives a near optimal recovery from the given data.

We next discuss the numerical implementation of the optimization with the loss (8.2) for this special $K = W^1(L_p[0,1])$. We can parameterize Σ_n using the hat function basis H_j, $j = 1, \dots, n$,

where H_j is the continuous piecewise linear function which takes the value one at ξ_j and the value 0 at all other ξ_i, $i \neq j$. Then each $S \in \Sigma_n$ can be written as

$$S(x) = S_{\mathbf{c}} := \sum_{j=1}^{n} c_j H_j(x), \quad x \in [0,1], \tag{8.7}$$

where $\mathbf{c} = (x_1, \ldots, c_n)$, Consider now the loss as a function of the parameters $\mathbf{c} = (c_1, \ldots, c_n)$:

$$\mathcal{L}^*(\mathbf{c}) := \mathcal{L}_\mu(S_{\mathbf{c}}), \quad \mathbf{c} \in \mathbb{R}^n. \tag{8.8}$$

The loss function $\mathcal{L}^*$ is strictly convex whenever $1 < p < \infty$ because it is the composition of a strictly convex function with an affine function.

To numerically compute the minimum of $\mathcal{L}_\mu$ over Σ_n we minimize $\mathcal{L}^*$ over $\mathbb{R}^n$ and use the argument $\mathbf{c}^*$ attaining this minimum to define the minimizer $\hat{f} := S_{\mathbf{c}^*}$. To compute $\mathbf{c}^*$ we use gradient descent with a sufficiently small step size and an initial guess. Since the loss is nonnegative and its gradient is locally Lipschitz except at $\mathbf{c} = 0$, the algorithm converges (see [1]).

As a numerical example, we take

$$f(x) = \frac{1}{4} x^{\frac{1}{2}}, \quad x \in [0,1]. \tag{8.9}$$

This function is in $W^1(L_p[0,1])$ for all $p < 2$. As a specific model class that contains f, we take

$$K = W^1(L_p[0,1]),\ p = 3/2. \tag{8.10}$$

This gives that $s = 5/6$. While we can implement the numerical algorithm for any data observations, in order to get a spectrum of performance results, we take random data samples consisting of $m = 10, 20, 40, 80, 160, 320$ observations. The additional observations are chosen randomly while retaining the previous random observations. Thus, we have a nested set of observations.

The random draws turned out to give the following values for h:

$$h(\mathbf{x}) = 0.13, 0.13, 0.108, 0.062, 0.048, 0.021.$$

For each of these values of m, we choose $n = 2m$ and $\mu = 0.1m^{-s}$. Note that this choice of n is less than suggested by the theoretical estimates.

Figure 8.2 gives a graph of the true recovery rate and compare it with the bound $h(\mathbf{x})^{\frac{5}{6}}$ which is our bound for the Chebyshev radius of K_w and hence optimal recovery rate for these data observations. We observe a asymptotic decay better than $h(\mathbf{x})^s$ (because we have taken only one function in K_w and not the supremum over all possible $f \in K_w$).

We next examine what happens if we do not use a penalty term, i.e., we take $\mu = 0$ and $n = 2m$. It is known that applying gradient descent with an initial choice of parameters converges to an interpolant which depends on the initial choice of parameters (see e.g. the discussion in [5]). We take the initial parameter choice as zero as we did in the case of a penalty term. Figure 8.3 compares the minimizing $\hat{f}$ for the case of $m = 40$ without regularization ($\mu = 0$) and with our proposed regularization ($\mu = 0.046$). For the former, the over-parametrized algorithm produces a highly oscillating $\hat{f}$ that interpolates the data samples. In contrast, the constructed $\hat{f}$ for $\mu = 0.046$ exploits the regularity of $f \in W^1(L_p[0,1])$ and yields a recovery error $\|f - \hat{f}_\Sigma\|_{L_2[0,1]} = 0.011$, which is 10 times smaller than when using $\mu = 0$.

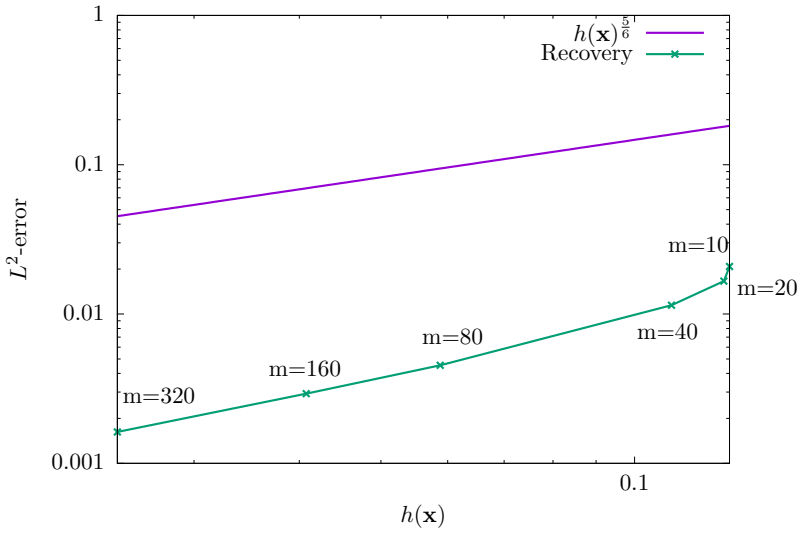

Figure 8.2: Recovery errors using $m = 10, 20, 40, ..., 320$ random samples, $\mu = 0.1m^{-\frac{5}{6}}$, and $n = 2m$. The error is compared with $h(\mathbf{x})^{\frac{5}{6}}$ which is the asymptotic behavior of the optimal error for the class $K = U(W^1(L_p(\Omega)))$, $p = \frac{3}{2}$.

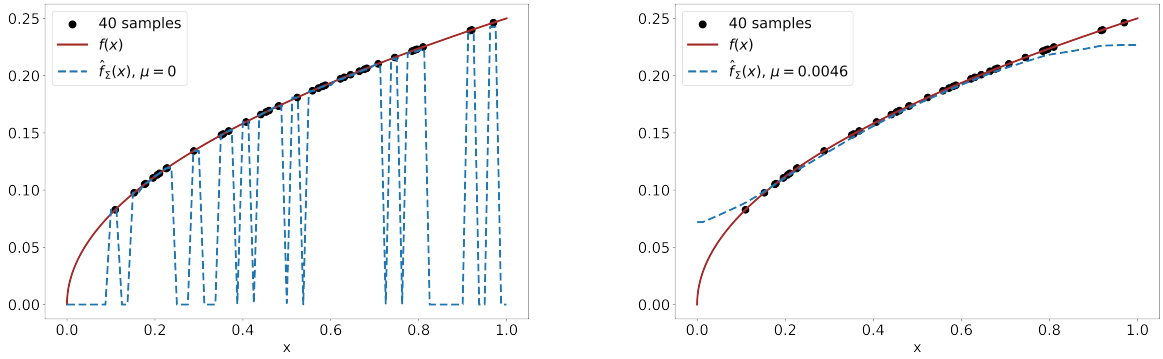

Figure 8.3: Learned function $\hat{f}_\Sigma$ in (8.2) with $p = 3/2$, $\mu = 0$ (left) and $\mu = 0.0046$ (right), and using 40 random samples (corresponding to $h(\mathbf{x}) = 0.108$) of the function $f(x) = \frac{1}{4}x^{\frac{1}{2}}$. The approximation space Σ is the set of continuous piecewise linear functions subordinate to a uniform partitions of $[0, 1]$ using $n = 80$ breakpoints. The recovery error $\|f - \hat{f}_\Sigma\|_{L_2[0,1]}$ with $\mu = 0.046$ is 0.011, which is 10 times smaller than when using $\mu = 0$.

8.2 Neural networks

It is now quite common in learning to take Σ_n as the space of outputs of a neural network depending on n parameters. We consider one often used example of this using ReLU activation. Let Ω be the unit Euclidean ball in $\mathbb{R}^d$ with $d \geq 1$. We consider the nonlinear space Σ_n of outputs of a single hidden layer ReLU neural network of width n on Ω (much less is known for deeper networks). Each

function $S \in \Sigma_n$ is of the form

$$S = \sum_{j=1}^{n} c_j(\omega_j \cdot x + b_j)_+ + c_0, \tag{8.11}$$

where $\omega_j \in \mathbb{R}^d$, and the $c_j, b_j \in \mathbb{R}$. This representation of S is not unique. We can require ω_j to satisfy $\|(\omega_j)\| = \|(\omega_j)\|_{\ell_2} = 1$ by adjusting the outer parameters c_j. We can also require the b_j to be in $[-2, 2]$. The set Σ_n is a nonlinear space of continuous piecewise linear functions on Ω .

It is commonly thought that Σ_n has significantly better approximation properties than more traditional approximation methods based on polynomials, splines, and wavelets and can therefore be more effective when learning a function f from data. This is especially thought to be true when d is large as it is for many modern learning problems. If this is indeed the case then it should be demonstrated through model classes K whose elements can be better approximated by neural networks than by the traditional approximation methods.

Accordingly, several model classes K have been introduced and studied because they have favorable approximation properties when using Σ_n. The most reknown of these is the Barron class introduced in [2] defined via Fourier transforms. Several generalization of these classes (see e.g. [8, 24, 21]) have been prominently studied. Each of these model classes is of the form $K = U(Y)$ where Y is a subspace of $C(\Omega)$. They all have the feature that the functions in K can be approximated in $L_q(\Omega)$, $1 \le q \le \infty$, with an approximation rate $O(n^{-\alpha})$, $n \to \infty$, with $\alpha \ge 1/2$, and hence these model classes do not suffer the curse of dimensionality in terms of d. In going further with our discussion, we let K be any of these model classes. We refer the reader to [5, 24] for results on the approximation of functions in K by the elements of Σ_n.

Optimal learning for these classes can be obtained via over parameterized learning as described in Theorem 3.1. However, several important issues remain unresolved and prevent a complete theory for these model classes. We describe these next where we assume the learning performance is to be measured in $X = L_2(\Omega, \nu)$ metric. Corresponding results are known for L_q, $1 \le q \le \infty$, but in some cases are less precise. The discussion below should be compared with the previous subsection.

Given data observations w at data sites $\mathbf{x}$ a major question that needs to be resolved is what is $R(K_w)_X$? Results in this direction are for the most part unknown although some partial information can be obtained from Gelfand widths and sampling numbers. Recall that Gelfand widths tell us the optimal learning rate that can be obtained for K when using data given by m linear functionals on X. Upper bounds on Gelfand widths are given in [25] and one expects that these bounds are sharp. If we consider learning from point values of a function $f \in K$ the situation is more opaque. The sampling numbers for K are not known. Jonathan Siegel has provided us with an argument based on the Rademacher complexity of K that shows that both the sampling numbers $s_m(K)_X$ and averaged sampling numbers $\bar{s}_m(K)_X$ of K in X are bounded by $Cm^{-1/4}$. However, we do not know lower bounds for sampling numbers and what is perhaps more crucial is we do not know the near best positioning of the points $x_1, \dots, x_m$ at which to sample $f \in K$. Resolving these open questions is important in learning since it tells us how many samples we would need of a function $f \in K$ in order to recover it with a prescribed error $\varepsilon > 0$. Also, it would tell us how much over parameterization we would need (how large to choose n for Σ_n) to obtain optimal learning.

When using over parameterized neural networks to solve the discrete minimization in Theorem 3.1, one can use ridge regression or LASSO applied to the loss as a function of the coefficient in

the representation (8.11). It is shown in [21] that there is always a minimizer which has a sparse representation (8.11).

In summary, for these model classes K, we can numerically find a near optimal recovery of $f \in K$ from given point data but we do not yet know the optimal learning rates nor do we know the optimal points where we should do the sampling. Some crude bounds bounds on performance and the amount of over parameterization are known but definitive results are still lacking.

9 Concluding remarks

We have shown that optimal learning under a model class assumption $f \in K$ is always solved by an over parameterized minimization problem. The use of over parameterization matches what is typically done in modern machine learning. However, it is important to point out that in many settings of modern learning one does not begin with a model class assumption and the loss function that is employed is simply a least squares fitting of the data absent any penalty term. In such a setting, i.e, absent any model class assumption, there can be no theory to describe optimal performance since f can be any function away from the data.

Another setting often studied is to employ neural networks Σ_n in the loss function together with a regularization term in the loss function which penalizes the size of the parameters. Such a penalty term can be viewed as imposing a model class assumption on the function to be learned. A precise formulation of this connection must still be worked out. One case where such a connection is known is when K is the unit ball of the Radon BV space; see [21].

In the setting without a model class assumption, as noted above, there are infinitely many solutions to the over parameterized minimization problem. The standard approach in learning is to choose one of these solutions by using a specific numerical algorithm to find an $\hat{f}$ corresponding to least squares loss. The typical setting employs parameterized deep neural networks in conjunction with minimization methods based on variants of gradient descent. This is sometimes referred to as *deep learning*. The analysis of deep learning revolves around questions of whether such minimization procedures converge, how the limit depends on the initial parameter guess and the learning rate (step size in gradient descent), and if convergence does hold then what is the function $\hat{f}$ that is learned (see the results on the Neural Tangent Kernel [12, 10]). Another way to word this approach is that one does not formulate a well defined learning problem (i.e. with a model class assumption) but rather proposes a specific numerical method to utilize for learning and then centers the discussion on when this works well and why? The current viewpoint is that the numerical method implicitly imposes a model class assumption (described via neural tangent kernels). Why such an implicit model class assumption is natural for the given learning setting is still to be explained.

Acknowledgment: The authors thank Professor Albert Cohen for helpful discussions on the research in this paper.

References

[1] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

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

[3] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *Journal of the American mathematical society*, 22(1):211–231, 2009.

[4] Albert Cohen, Mark A Davenport, and Dany Leviatan. On the stability and accuracy of least squares approximations. *Foundations of computational mathematics*, 13(5):819–834, 2013.

[5] Ronald DeVore, Boris Hanin, and Guergana Petrova. Neural network approximation. *Acta Numerica*, 30:327–444, 2021.

[6] Ronald DeVore, Guergana Petrova, and Przemyslaw Wojtaszczyk. Data assimilation and sampling in banach spaces. *Calcolo*, 54(3):963–1007, 2017.

[7] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[8] Weinan E, Chao Ma, and Lei Wu. Barron spaces and the compositional function spaces for neural network models. *arXiv preprint arXiv:1906.08039*, 2019.

[9] Simon Foucart and Holger Rauhut. An invitation to compressive sensing. In *A mathematical introduction to compressive sensing*, pages 1–39. Springer, 2013.

[10] Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. *arXiv preprint arXiv:1909.05989*, 2019.

[11] Trevor Hastie, Robert Tibshirani, and Ryan Tibshirani. Best subset, forward stepwise or lasso? Analysis and recommendations based on extensive comparisons. *Statistical Science*, 35(4):579–592, 2020.

[12] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[13] David Krieg, Erich Novak, and Mathias Sonnleitner. Recovery of Sobolev functions restricted to iid sampling. *arXiv preprint arXiv:2108.02055*, 2021.

[14] David Krieg and Mathias Sonnleitner. Random points are optimal for the approximation of sobolev functions. *arXiv preprint arXiv:2009.11275*, 2020.

[15] David Krieg and Mario Ullrich. Function values are enough for L_2-approximation. *Foundations of Computational Mathematics*, 21(4):1141–1151, 2021.

[16] George G Lorentz, Manfred v Golitschek, and Yuly Makovoz. *Constructive approximation: advanced problems*. Springer, 1996.

[17] Charles A Micchelli and Theodore J Rivlin. A survey of optimal recovery. *Optimal estimation in approximation theory*, pages 1–54, 1977.

[18] Nicolas Nagel, Martin Schäfer, and Tino Ullrich. A new upper bound for sampling numbers. *Foundations of Computational Mathematics*, pages 1–24, 2021.

[19] Francis Narcowich, Joseph Ward, and Holger Wendland. Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting. *Mathematics of Computation*, 74(250):743–763, 2005.

[20] E Novak and H Wozniakowski. Tractability of multivariate problems, linear information, vol. I. european mathematical society. *Europe*, 2008.

[21] Rahul Parhi and Robert D Nowak. Banach space representer theorems for neural networks and ridge splines. *J. Mach. Learn. Res.*, 22(43):1–40, 2021.

[22] Philipp Petersen and Felix Voigtlaender. Optimal learning of high-dimensional classification problems using deep neural networks. *arXiv preprint arXiv:2112.12555*, 2021.

[23] Allan Pinkus. *N-widths in Approximation Theory*, volume 7. Springer Science & Business Media, 2012.

[24] Jonathan W Siegel and Jinchao Xu. Characterization of the variation spaces corresponding to shallow neural networks. *arXiv preprint arXiv:2106.15002*, 2021.

[25] Jonathan W Siegel and Jinchao Xu. Sharp bounds on the approximation rates, metric entropy, and n-widths of shallow neural networks. *arXiv preprint arXiv:2101.12365*, 2021.

[26] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[27] Joe Fred Traub and H. Wozniakowski. *A general theory of optimal algorithms*. Academic Press, 1980.

[28] Kösaku Yosida. *Functional analysis*. Springer Science & Business Media, 2012.