



## Optimal learning

Peter Binev<sup>1</sup> · Andrea Bonito<sup>2</sup> · Ronald DeVore<sup>2</sup> · Guergana Petrova<sup>2</sup>

Received: 23 November 2022 / Revised: 10 May 2023 / Accepted: 26 June 2023 /

Published online: 19 February 2024

© The Author(s) under exclusive licence to Istituto di Informatica e Telematica (IIT) 2024

### 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 finite-dimensional 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.

---

✉ Guergana Petrova  
gpetrova@tamu.edu

Peter Binev  
binev@math.sc.edu

Andrea Bonito  
bonito@tamu.edu

Ronald DeVore  
ronaldddevore@tamu.edu

<sup>1</sup> Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

<sup>2</sup> Department of Mathematics, Texas A&M University, College Station, TX 77843, USA

**Keywords** Optimal learning · Over-parametrization · Regularization · Chebyshev radius · Banach space

**Mathematics Subject Classification** 46N10 · 47A52 · 65J20 · 68T05

## 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.

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  $\Omega$  a domain in  $\mathbb{R}^d$ , or smoothness spaces such as Sobolev spaces on  $\Omega$ . 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. [12, 26, 37]). Let us denote the set of all possible candidates for  $f$  by  $K^*$ , i.e.,

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

This is a compact subset of  $K$ . When we are presented the data, all we know is that it came from some  $f \in K$  but we do not know which one. Thus, the task of optimal recovery is to find one function  $g \in X$  which simultaneously approximates all elements in  $K^*$  to an error  $R$  with  $R$  as small as possible. This best error is described in the next paragraph.

We denote by  $B(z, r)_X$  the ball in  $X$  of radius  $r$  with center  $z$ . Given any compact set  $S$  in  $X$ , the Chebyshev radius  $R(S)_X$  of  $S$  is defined as

$$R(S)_X := \inf\{r : S \subset B(z, r)_X \text{ for some } z \in X\}. \quad (1.2)$$

While it can happen that  $R(S)_X$  may not be attained, the number  $R(S)_X$  is well defined and is the *error of optimal recovery*. It can also happen that  $R(S)_X$  is assumed but the center  $z$  is not in  $S$ .

We now return to our set  $S = K^*$ . When  $R = R(K^*)_X$  is assumed by a ball  $B(z, R)$ , then  $z$  would provide an optimal recovery for  $K^*$ . For more details on optimal recovery and Chebyshev balls, we refer the reader to [12].

We return to our set  $K^*$ . While the previous paragraph gives a simple mathematical description of the optimal recovery of all functions  $f \in K^*$ , it is nowhere close to giving a numerical procedure for learning since finding an appropriate  $z$  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

$$\|f - \hat{f}\|_X \leq CR(K^*), \quad f \in K^*, \quad (1.3)$$

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 method.

## 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

$$w_i := \lambda_i(f), \quad i = 1, \dots, m, \quad (1.4)$$

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

$$K_w := \{f \in K : \lambda_i(f) = w_i, i = 1, \dots, m\}, \quad w = (w_1, \dots, w_m), \quad (1.5)$$

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

$$\text{optimal recovery rate} = R(K_w)_X. \quad (1.6)$$

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

$$W := W_K := W(\lambda_1, \dots, \lambda_m; K) := \{(\lambda_1(g), \dots, \lambda_m(g)) : g \in K\} \subset \mathbb{R}^m, \quad (1.7)$$

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 values  $\bar{w}$ , namely for all  $\bar{w} \in \mathbb{R}^m \setminus W$ , we define  $K_{\bar{w}} := \emptyset$ .

The most convenient assumption to make about the  $\lambda_j$ '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  $\mu$  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  $\Omega$ . 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 Sect. 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  $\lambda_j$ '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  $\lambda_j$ , subject to the restriction that there are only  $m$  of them.

**Setting III:** The  $\lambda_j$ '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

$$d^m(K)_X := \inf_{\lambda_1, \dots, \lambda_m \in X^*} \sup_{w \in \mathbb{R}^m} R(K_w)_X, \quad (1.8)$$

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  $m \rightarrow \infty$ . Standard reference for results on Gelfand widths in classical settings are [24, 33] 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  $\varepsilon$ , 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

$$d_{\text{ave}}^m(K)_X := \text{Exp}_{\lambda_1, \dots, \lambda_m} \sup_{w \in \mathbb{R}^m} R(K_w)_X, \quad (1.9)$$

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

$$s_m(K)_X \quad (1.10)$$

in this paper. Two of the standard references on this line of investigation are [29, 37]. Because of their importance in learning, finding the sampling numbers for various model classes  $K$  is an active research topic (see e.g. [10, 21, 23, 27]).

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 procedure 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 [21]) 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 specific 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 design a procedure that provably converges to an optimal or near optimal recovery of  $f$ , i.e., reaches the optimal benchmark. The development of learning procedures usually proceeds through two stages. The first is to formulate a finite-dimensional optimization problem associated to the data. Here, finite-dimensional means that the

optimization problem depends on only a finite number of parameters. The second stage is to propose and analyze numerical procedures for solving the finite-dimensional optimization. In this paper, we shall primarily concern ourselves with the first stage and ask the question:

**Which finite-dimensional 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 procedure. 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 finite-dimensional 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 procedures based on Chebyshev balls. The remainder of the paper concentrates on introducing finite-dimensional minimization formulations, under a model class assumption, whose solution is near optimal. Each of these finite-dimensional minimizations can be taken of the form

$$\hat{f} \in \operatorname{argmin}_{g \in \Sigma_n} \left( \tau \sum_{j=1}^m [w_j - \lambda_j(g)]^2 + \mu \mathbf{pen}_K(g) \right), \quad (1.11)$$

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

We have chosen to call these problems finite-dimensional minimization problems since the minimization is performed over the set  $\Sigma_n$  depending on a finite number  $n$  of parameters. In going further, we will denote this set simply by  $\Sigma$ . The number of its parameters depends on the accuracy  $\delta > 0$  with which  $\Sigma$  approximates  $K$ . For example, we will impose that

$$\operatorname{dist}(K, \Sigma)_X := \sup_{f \in K} \operatorname{dist}(f, \Sigma)_X = \sup_{f \in K} \inf_{g \in \Sigma} \|f - g\|_X < \delta,$$

for suitably chosen small enough  $\delta > 0$ . The value of  $\delta$  will determine the number  $n = n(\delta)$  of parameters needed to describe  $\Sigma$ . Thus, the smaller the  $\delta$ , the bigger the  $n$ , which corresponds to the fact that the set  $\Sigma$  is described by  $n$  parameters whose number will be (in general) much larger than the number  $m$  of data available for  $f$  (and thus justify the use of the term *over-parametrized* optimization problem).

**Remark 1.1** In general, the minimization problem in (1.11) may not have a unique solution from  $\Sigma_n$ . 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 Sect. 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 Sect. 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 Sect. 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 Sect. 5, we consider the case when the data observations are corrupted by additive deterministic 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 Sect. 7. In numerical implementations it is convenient to use other forms of the loss function appearing in (1.11). We discuss this aspect in Sect. 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 finite-dimensional 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 procedure. While much is known about solutions to this finite-dimensional problem [30, 34] and numerical methods for solving the optimization problem [18, 36], 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 Sect. 8.2.

There is a rather vast literature on optimal recovery and learning. We close this introduction with a few remarks which can serve to explain how our results fit into the current literature. Let  $X$  be the Banach space in whose norm  $\|\cdot\|_X$  we measure the optimal recovery or learning error. If the model class  $K$  is the unit ball  $K := \{f : \|f\|_Y \leq 1\}$  of a Banach space  $Y$  which compactly embeds into  $X$ , we let  $g$  be the minimum norm interpolant of the data. That is,  $g$  is the function in  $Y$  which satisfies the data and has smallest  $Y$  norm. Obviously,  $g$  is in  $K$  and is therefore a near optimal recovery with constant  $C \leq 2$ . If  $Y$  is a Hilbert space (regardless of whether  $X$  is) then it can be shown that  $g$  is actually the Chebyshev center of  $K$  and  $g$  is an optimal recovery ( $C = 1$ ). The function  $g$  is sometimes referred to as an interpolating spline even though it is not necessarily a spline function in the classical sense. Of

course, this does not directly give an algorithm since one still must compute  $g$ . There is a general numerical strategy for computing  $g$  which rests on computing the Riesz representers of the linear functional  $\lambda_j$  viewed as functionals on  $Y$ . If one adopts this approach then one still must answer the question of how accurate the computation of the Riesz representers must be in order to have near optimality of recovery. There are some settings in which one can prove that the minimum norm interpolant  $g$  lives in a finite dimensional linear space or finite dimensional manifold (see [38] and [30] for recent literature of this type). When such representer theorems are proven, they provide a powerful numerical tool. The approach discussed in the present paper differs from the minimum norm interpolant approach in that we treat more general model classes  $K$  and we put forward a finite-dimensional optimization problem based on penalties that is always guaranteed to give a near optimal solution. Moreover, we quantify how fine we must choose the discretization  $\Sigma_n$  and how small we must choose the penalty parameter  $\tau$ .

Optimization problems of the type proposed here to find near optimal recovery are very common in the literature and fall under the notions of Tikhonov regularization, least squares minimization, and LASSO problems (see [19]). While these methods are used often, the typical results in the literature do not describe how this is to be carried out if one wants to guarantee a near optimal recovery. Certified performance is usually proven in particular applications (such as sparse signal recovery) and fixed form of the loss functions (see [6–8, 25]). In this paper, we present a general framework and derive provable bounds on how much over-parameterization is needed and how the penalization has to be scaled in order to obtain a near optimal recovery of the observed function.

## 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 (1.4), 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  $\lambda_j$ ,  $j = 1, \dots, m$ , are the observation functionals. Without loss of generality, we can assume that the  $\lambda_j$ 's are normalized to have norm one, that is,  $\|\lambda_j\|_{X^*} = 1$ ,  $j = 1, \dots, m$ . All theorems that follow can be restated in the general case, with the norms of the functionals present as constants at the appropriate places. We shall also use the notation

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

and (1.7) throughout this paper. Notice that since  $\lambda$  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  $\lambda$  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

$$\|f - \hat{f}\|_X \leq CR(K_w)_X, \quad (2.2)$$

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 finite-dimensional 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  $\ell_2$  norm

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

Notice that the data mapping  $\lambda$  is a linear mapping from  $X$  to  $\mathbb{R}^m$  whose norm is one when we use (2.3) as the norm for  $\mathbb{R}^m$ . This means that  $\lambda$  is a Lipschitz mapping:

$$\|\lambda(f) - \lambda(g)\| \leq \|f - g\|_X, \quad f, g \in X. \quad (2.4)$$

We shall use this fact repeatedly, usually without further mention, in this paper.

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

$$\|\lambda(g_\varepsilon) - w\| \leq \varepsilon \quad \text{and} \quad \text{dist}(g_\varepsilon, K)_X \leq \varepsilon. \quad (2.5)$$

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

$$R(K(w, \varepsilon))_X, \quad \text{where } K(w, \varepsilon) := \bigcup_{w': \|w' - w\| \leq \varepsilon} K_{w'}, \quad (2.6)$$

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*

$$\lim_{\varepsilon \rightarrow 0^+} R(K(w, \varepsilon))_X = R(K_w)_X. \quad (2.7)$$

**Proof** Since the sets  $K(w, \varepsilon)$ ,  $\varepsilon > 0$ , are nested, the function  $R(K(w, \varepsilon))_X$ ,  $\varepsilon > 0$ , is decreasing as  $\varepsilon$  decreases. Hence, the limit in (2.7) exists. Suppose that this limit is  $R_0$  and  $R_0 > R(K_w)_X$ . Let us fix  $\xi > 0$  such that  $R_0 > R_0 - \xi > R(K_w)_X$ . Since for each  $\varepsilon > 0$  we have  $R(K(w, \varepsilon))_X \geq R_0$ , there is an  $f_\varepsilon \in K(w, \varepsilon)$  with  $\|f_\varepsilon - z_w\|_X \geq R_0 - \xi$ , where  $z_w$  is the Chebyshev center of  $B(K_w)_X$ . Then we take  $\varepsilon_n \rightarrow 0$  and consider the corresponding sequence  $f_{\varepsilon_n} \in K(w, \varepsilon_n)$  with  $\|f_{\varepsilon_n} - z_w\|_X \geq R_0 - \xi$ . Since  $K$  is compact, this sequence has a subsequence which converges to a limit  $f^*$  in  $K$  and  $\|f^* - z_w\|_X \geq R_0 - \xi > R(K_w)_X$ . We also know that  $\lambda(f^*) = w$  and so  $f^* \in K_w$ . This means that  $\|f^* - z_w\|_X \leq R(K_w)_X$ . This contradicts the assumption  $R_0 > R(K_w)_X$  and proves (2.7).  $\square$

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

$$\lim_{\varepsilon \rightarrow 0^+} R(K(w, \gamma + \varepsilon))_X = R(K(w, \gamma))_X. \quad (2.8)$$

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  $g_\varepsilon$  in terms of how closely it fits the data and how close it is to the model class  $K$ .

**Theorem 2.4** *If  $g_\varepsilon$  is any function in  $X$  satisfying (2.5), then*

$$\|f - g_\varepsilon\|_X \leq \varepsilon + 2R(K(w, 2\varepsilon))_X, \quad f \in K_w. \quad (2.9)$$

If  $R(K_w) \neq 0$ , then for any  $C > 2$  and for  $\varepsilon$  suitably small the function  $g_\varepsilon$  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

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

where we have used (2.4). 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 (2.9). 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.  $\square$

## 2.2 The behavior of $R(K_w)_X$

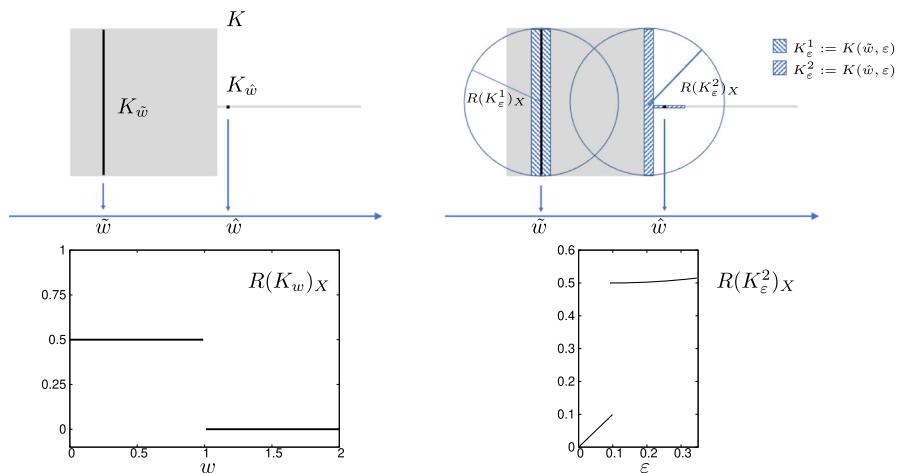
The analysis that follows in this paper depends on the function  $\varepsilon \mapsto R(K(w, \varepsilon))_X$  and so it may be useful to the reader to make a few remarks on this function. Its behavior depends very much on  $K$  and needs to be analyzed for each  $K$  individually. From the above estimates, we see that a critical issue in quantitative bounds for the performance of learning procedures is the rate of convergence of  $R(K(w, \varepsilon))_X$  to  $R(K_w)_X$  as  $\varepsilon \rightarrow 0^+$ . 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  $K := [0, 1]^2 \cup ([1, 2] \times \{\frac{1}{2}\}) \subset X = \mathbb{R}^2$ , equipped with the Euclidean norm. We take the measurement functional  $\lambda$  to be the first coordinate of a point  $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ :  $\lambda(\mathbf{x}) = x_1$ . Thus, we have

$$K_w = \begin{cases} \{(w, y) \in \mathbb{R}^2 : y \in [0, 1]\}, & w \in [0, 1], \\ \{(w, \frac{1}{2})\}, & w \in [1, 2], \\ \emptyset, & w \in (-\infty, 0) \cup (2, \infty). \end{cases}$$

For this example the function  $R(K_w)_X$  is a discontinuous function of  $w$  (see the bottom-left graph in Fig. 1). Now we consider the sets  $K_\varepsilon^1 := K(\tilde{w}, \varepsilon)$  with  $\tilde{w} \in (0, 1)$  and  $K_\varepsilon^2 := K(\hat{w}, \varepsilon)$  with  $\hat{w} = 1.1$ , pictured in Fig. 1 (top-right), along with their corresponding Chebyshev balls. The graph of  $R(K_\varepsilon^2)_X$  as a function of  $\varepsilon$  is presented at the bottom-right. This function is a discontinuous function of  $\varepsilon$  with the discontinuity occurring 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  $R(K(\hat{w}, \varepsilon))_X$ ,  $\varepsilon \rightarrow 0^+$ , towards  $R(K_{\hat{w}})_X = 0$  is not uniform in  $\hat{w}$  and depends on the distance of  $\hat{w}$  to  $[0, 1]$ .

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



**Fig. 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  $K_\varepsilon^1$  and  $K_\varepsilon^2$  and their corresponding Chebyshev balls; Bottom-Right: Graph of  $R(K_\varepsilon^2)_X$  as a function of  $\varepsilon$

### 3 Near optimal recovery through discretization

The results of the previous section do not constitute a numerical learning procedure. Rather, they only show that optimal performance can be obtained if an algorithm provides a function  $g_\varepsilon$  satisfying condition (2.5).

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

Any numerical procedure for learning is based on some method of approximation. The most common tools used are polynomials, splines, wavelets, or neural networks. Let  $\Sigma_n$ ,  $n = 1, 2, \dots$ , be the sets used for the approximation, where  $n$  denotes the complexity of  $\Sigma_n$ . The two main examples we have in mind are the cases where  $\Sigma_n$  is a linear subspace of  $X$  of dimension  $n$  and the case where  $\Sigma_n$  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 [11] for an overview).

The first question we address is how we should use  $\Sigma_n$  to build a numerical procedure. 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. [41]). Since  $K$  is compact, we have the embedding inequality

$$\|g\|_X \leq C_0 \|g\|_Y, \quad g \in Y, \quad (3.1)$$

where  $C_0$  is the embedding constant which depends only on  $K$ . In order to simplify the notations, we use the convention  $\|g\|_Y := \infty$  for  $g \notin Y$ . In this setting, we introduce for any  $\mu > 0$  the loss function

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

This loss function is defined for all  $g \in X$  but infinite when  $g \notin Y$ .

We choose here not to raise the norms involved in the loss function to any powers but consider such variants in Sect. 6. Formulation (3.2) is also known as the “square-root” formulation of the loss function since the data fidelity term is not squared. It is analogous to the so-called “square-root LASSO” decoder in statistics, compressed sensing and machine learning. One of the motivations for introducing the square-root LASSO is that it is agnostic to the noise level, that is, the tuning parameter  $\mu$  can be chosen independently of the norm of the noise, whereas in the standard LASSO the optimal tuning parameter should depend on the norm of the noise. These decoders have recently become quite popular in learning problems. While they were first popularized in high-dimensional statistics (see [39]), recently they were studied in the context of (weighted) sparse recovery in compressed sensing (see [1, 3, 15, 31]) and deep learning for high-dimensional function approximation (see [2]).

The following theorem describes a finite-dimensional 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*

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

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

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

*is a near optimal recovery of  $f$ , that is,*

$$\|f - \hat{f}\|_X \leq CR(K_w)_X, \quad (3.5)$$

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

$$\varepsilon := \mu \max(C_0, 1 + \mu) \quad (3.6)$$

satisfies the inequality

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

which is possible for  $\mu > 0$  small enough 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  $f_\Sigma \in \Sigma \cap K$  satisfy  $\|f - f_\Sigma\|_X \leq \delta$ . From the definition of  $\hat{f}$ , we know that

$$\|w - \lambda(\hat{f})\| + \mu\|\hat{f}\|_Y \leq \|w - \lambda(f_\Sigma)\| + \mu\|f_\Sigma\|_Y \leq \delta + \mu, \quad (3.7)$$

where the first term was estimated by

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

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

We now assume that  $\delta \leq \mu^2$  and  $\mu$  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

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

In other words,  $g_\varepsilon := \hat{f}$  satisfies (2.5) 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  $\mu$  (and hence  $\delta$ ) is sufficiently small. The last statement of the theorem follows from Theorem 2.4.  $\square$

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

### 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

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

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*

$$\text{dist}(K, \Sigma)_X < \delta. \quad (3.9)$$

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

$$\hat{f} \in \underset{g \in \Sigma}{\operatorname{argmin}} \mathcal{L}_K(g) \quad (3.10)$$

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

$$\|f - \hat{f}\|_X \leq CR(K_w)_X, \quad (3.11)$$

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

$$2\delta + 2R(K(w, 4\delta))_X \leq CR(K_w)_X,$$

*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  $f_\Sigma \in \Sigma$  satisfy  $\|f - f_\Sigma\|_X \leq \delta$ . From the definition of  $\hat{f}$ , we know that

$$\|w - \lambda(\hat{f})\| + \text{dist}(\hat{f}, K)_X \leq \|w - \lambda(f_\Sigma)\| + \text{dist}(f_\Sigma, K)_X \leq \delta + \delta = 2\delta, \quad (3.12)$$

where the first term was estimated by

$$\|w - \lambda(f_\Sigma)\| = \|\lambda(f - f_\Sigma)\| \leq \|f - f_\Sigma\|_X \leq \delta.$$

It follows that the function  $g_\varepsilon := \hat{f}$  satisfies (2.5) with  $\varepsilon = 2\delta$ . From Theorem 2.4, we find that

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

If  $C > 2$  and  $\delta$  is sufficiently small, Lemma 2.2 gives that the right side of (3.13) does not exceed  $CR(K_w)_X$ .  $\square$

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 procedure for near optimal recovery since we have not given a numerical recipe for solving the corresponding finite-dimensional problem. This is discussed in more details in Sect. 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  $\Omega$  is the closure of a bounded domain in  $\mathbb{R}^d$ . For the moment, we assume that we have noiseless observations

$$w_i = f(x_i), \quad x_i \in \Omega \subset \mathbb{R}^d, \quad i = 1, \dots, m, \quad (4.1)$$

where the *data sites*  $x_i$  come from  $\Omega$ . The most common choice of the metric in which to measure recovery error is an  $X = L_p(\Omega)$  norm on  $\Omega$  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  $\mathcal{L}_\mu$  or  $\mathcal{L}_K$ , it is enough to have point evaluation well defined for functions in the model class  $K$  and functions in  $\Sigma$ .

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 procedure in a Banach space  $X$  for which we have an embedding

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

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

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

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

$$\begin{aligned} \lambda_{\mathbf{x}}(g) &:= (g(x_1), \dots, g(x_m)) \in \mathbb{R}^m, \quad \mathbf{x} := (x_1, \dots, x_m), \quad x_i \in \Omega \subset \mathbb{R}^d, \\ g &\in C(\Omega), \end{aligned} \quad (4.4)$$

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  $g_{\varepsilon}$  is any function in  $C(\Omega)$  with  $\|w - \lambda_{\mathbf{x}}(g_{\varepsilon})\| \leq \varepsilon$  and  $\text{dist}(g_{\varepsilon}, K)_{C(\Omega)} \leq \varepsilon$ , then*

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

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

**Remark 4.2** In the last proposition, it is natural to ask why we need  $g_\varepsilon$  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  $g_\varepsilon$  is one at the data and  $\|g_\varepsilon\|_{L_2[0,1]} < \varepsilon$ , then  $g_\varepsilon$  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*

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

**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 we fix  $\xi > 0$  such that  $R_0 > R_0 - \xi > R(K_w)_X$  and because of the compactness of  $K$  in  $C(\Omega)$  there is a sequence  $\varepsilon_n \rightarrow 0^+$  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,  $\|f^* - z_w\|_X \geq R_0 - \xi > R(K_w)$ . From the convergence in  $C(\Omega)$ , it follows that  $f^*$  is in  $K_w$  and so we have  $\|f^* - z_w\|_X \leq R(K_w)_X$ , which contradicts that  $\|f^* - z_w\|_X > R(K_w)$ .  $\square$

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

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

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

The following theorem describes a finite-dimensional 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  $\Sigma$  satisfy the condition*

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

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

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

is a near optimal recovery of  $f$ , that is,

$$\|f - \hat{f}\|_X \leq CR(K_w)_X, \quad (4.9)$$

for any  $C > 2$ , provided that  $R(K_w)_X \neq 0$ , and  $\delta$  and  $\mu$  are sufficiently small. More precisely, it is enough to choose  $\delta \leq \mu^2$  and  $\mu$  small enough so that  $C_X \varepsilon + 2R(K(w, 2\varepsilon))_X \leq CR(K_w)_X$  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  $f_\Sigma \in \Sigma \cap K$  to satisfy  $\|f - f_\Sigma\|_{C(\Omega)} \leq \delta$  and use Proposition 4.1 and Lemma 4.3.  $\square$

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  $\Sigma$  satisfy the condition*

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

*If  $f \in K_w$ , where  $w = \lambda_x(f)$ , then the function*

$$\hat{f} := \hat{f}_\Sigma \in \underset{g \in \Sigma}{\operatorname{argmin}} \mathcal{L}'_K(g), \quad \text{where } \mathcal{L}'_K(g) := \|\lambda_x(g) - w\| + \text{dist}(g, K)_{C(\Omega)}, \quad (4.11)$$

*is a near optimal recovery of  $f$ , that is*

$$\|f - \hat{f}\|_X \leq CR(K_w)_X, \quad (4.12)$$

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

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

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

## 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

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

where the real numbers  $\eta_j$ ,  $j = 1, \dots, m$ , are unknown to us. However, to derive quantitative results on performance, we will have to make some assumptions on the

unknown noise vector  $\eta := (\eta_1, \dots, \eta_m)$ . We assume that all we know about  $\eta$  is its size.

We continue to let  $w_j = f(x_j)$ ,  $j = 1, \dots, m$ , and  $w := (w_1, \dots, w_m)$ . 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  $f(x_j) = w_j$ ,  $j = 1, \dots, m$ , 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

$$\|\eta\| \leq \gamma < \infty. \quad (5.2)$$

Then, the totality of information we have about  $f$  is that  $f \in K$  and  $f$  satisfies the data  $\tilde{w} - \eta$  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

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

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

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

where the parameter  $\tau$  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  $\Sigma$  satisfy the condition*

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

*Consider the function*

$$\begin{aligned} \hat{f} &:= \hat{f}_{\Sigma, \tau} \in \underset{g \in \Sigma}{\operatorname{argmin}} \mathcal{L}_{K, \tau}(g), \\ \text{with } \mathcal{L}_{K, \tau}(g) &:= \tau \|\tilde{w} - \lambda_{\mathbf{x}}(g)\| + \text{dist}(g, K)_{C(\Omega)}, \end{aligned} \quad (5.6)$$

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

$$\|f - \hat{f}\|_X \leq CR(K(\tilde{w}, \gamma))_X, \quad (5.7)$$

*for any  $C > 2$ , provided  $R(K(\tilde{w}, \gamma))_X \neq 0$ ,  $\delta \leq \tau^2$  and  $\tau$  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  $\eta$  satisfying  $\|\eta\| \leq \gamma$ . Let  $f_\Sigma \in \Sigma$  satisfy  $\|f - f_\Sigma\|_{C(\Omega)} \leq \delta$ . Then, we know that  $\|\lambda_x(f_\Sigma) - w\| \leq \delta$  and so  $\|\lambda_x(f_\Sigma) - \tilde{w}\| \leq \delta + \gamma$ . Since  $f_\Sigma \in \Sigma$ , we have

$$\begin{aligned} \tau \|\tilde{w} - \lambda_x(\hat{f})\| + \text{dist}(\hat{f}, K)_{C(\Omega)} &\leq \tau \|\tilde{w} - \lambda_x(f_\Sigma)\| + \text{dist}(f_\Sigma, K)_{C(\Omega)} \\ &\leq \tau(\delta + \gamma) + \delta < \tau\gamma + 2\delta. \end{aligned}$$

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

$$\|\tilde{w} - \lambda_x(\hat{f})\| < \gamma + 2\frac{\delta}{\tau} \leq \gamma + 2\tau \quad \text{and} \quad \text{dist}(\hat{f}, K)_{C(\Omega)} \leq \tau\gamma + 2\delta. \quad (5.8)$$

Now let  $h \in K$  satisfy

$$\|\hat{f} - h\|_{C(\Omega)} \leq \tau\gamma + 2\delta \leq \tau\gamma + 2\tau^2 = \tau(\gamma + 2\tau) < 3\tau. \quad (5.9)$$

Then, we have

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

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

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

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

$$\begin{aligned} \|f - \hat{f}\|_X &\leq \|f - h\|_X + \|h - \hat{f}\|_X \leq \|f - h\|_X + C_X \|h - \hat{f}\|_{C(\Omega)} \\ &\leq 2R(K(\tilde{w}, \gamma + 5\tau))_X + 3C_X\tau. \end{aligned} \quad (5.10)$$

If we use Remark 2.3, we obtain (5.7) for any  $C > 2$ , provided we take  $\tau$  suitably small, see (2.8).  $\square$

**Remark 5.2** The appearance of the parameter  $\tau$  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  $\gamma$  or  $\tau$  in order to arrive at the inequality (5.10). However, to guarantee that the recovery is near optimal, one needs to choose  $\tau$  (and hence  $\delta$ ) 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

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

where  $\alpha > 0$  and  $\beta > 0$  are fixed. The modified loss function (6.1) generalizes the original loss function (3.2) by raising the data fidelity and regularization term to possibly different powers. The case  $\alpha = \beta = 2$  is particularly appealing when  $Y$  is a Hilbert space as it leads to a simple expression for the derivative of  $\mathcal{L}'_\mu$ . Also, the modified loss function can be viewed as a generalization of the classical LASSO procedure, which corresponds to  $\alpha = 2$  and  $\beta = 1$ . In some special settings, such loss functions have been considered before in the context of recovery of sparse signals, see [15, 31] and the references therein.

**Remark 6.1** If  $\Sigma$  is convex (e.g. if it is a finite dimensional linear space), if  $\|\cdot\|_Y$  is strictly convex, and if  $\alpha, \beta \geq 1$ , then the minimizer of  $\mathcal{L}'_\mu(g)$  over  $g \in \Sigma$  is unique. In this case, if  $\Sigma$  is a finite dimensional linear space, the solution  $\hat{f}$  of the resulting optimization problem (6.2) can be computed by available optimization algorithms. 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.

Following the ideas from Sect. 3.1, we establish similar near optimality results for this loss function in the case where the measurements are linear functionals on  $X$ .

**Theorem 6.2** *Let  $K = U(Y)$  with  $Y$  a normed linear subspace of  $X$  and let the set  $\Sigma$  satisfy (3.3). Then, for any  $C > 2$ , the function*

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

*is a near optimal recovery, i.e.,*

$$\|f - \hat{f}\|_X \leq CR(K_w)_X, \quad f \in K_w, \quad (6.3)$$

*provided  $R(K_w)_X \neq 0$ ,  $\delta^\alpha \leq \mu^2$  and  $\mu$  is sufficiently small. Moreover, in the case of numerical optimization producing  $\tilde{f}$  such that  $\mathcal{L}'_\mu(\tilde{f}) \leq \mathcal{L}'_\mu(\hat{f}) + \epsilon$  for some  $\epsilon > 0$ , the estimate (6.3) remains valid with  $\hat{f}$  replaced by  $\tilde{f}$  and provided that  $\epsilon \leq \delta^\alpha \leq \frac{1}{2}\mu^2$  and  $\mu$  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  $f_\Sigma \in \Sigma \cap K$  satisfy  $\|f - f_\Sigma\|_X \leq \delta$ . From the definition of  $\hat{f}$ , we know that

$$\|w - \lambda(\hat{f})\|^\alpha + \mu \|\hat{f}\|_Y^\beta \leq \|w - \lambda(f_\Sigma)\|^\alpha + \mu \|f_\Sigma\|_Y^\beta \leq \delta^\alpha + \mu, \quad (6.4)$$

where the first term was estimated by

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

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

We now assume that  $\delta^\alpha \leq \mu^2$  and  $\mu$  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

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

This means that  $g_\varepsilon := \hat{f}$  satisfies (2.5) 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  $\mu$  (and hence  $\delta$ ) is sufficiently small.

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

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

and the proof is completed as above.  $\square$

## 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  $\lambda_1, \dots, \lambda_m$  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  $\lambda_1(f), \dots, \lambda_m(f)$ ? The Gelfand width

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

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

The Gelfand widths of model classes  $K$  are a well studied concept in Functional Analysis and Approximation Theory (see e.g. the book of Pinkus [33]). 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. [32, 34]). 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  $\lambda_1, \dots, \lambda_m \in X^*$  is with high probability near optimal (see [9, 13] and the many books written on compressed sensing such as [16]).

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 the concept of Gelfand widths is that in practical applications of sampling, one does not have access to arbitrary chosen general linear functionals of the target signal. Instead, the available functionals are more restricted. For this reason, one typically imposes restrictions on the functionals  $\lambda_j$ ,  $j = 1, \dots, m$ . 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  $\Omega$  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

$$s_m(K)_X := \inf_{x_1, \dots, x_m \in \Omega} \sup_{f \in K} R(K_{f(\mathbf{x})})_X, \quad f(\mathbf{x}) := (f(x_1), \dots, f(x_m)), \quad (7.2)$$

$$m = 1, 2, \dots$$

The points  $x_1, \dots, x_m$  that give the infimum in  $s_m(K)_X$  are the optimal sampling sites. For spaces  $X$  for which point evaluations are linear functionals, we obviously have  $s_m(K)_X \geq d^m(K)_X$  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  $m \rightarrow \infty$ .

### 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  $\rho$  on  $\Omega$ . The relevant measure of performance is the averaged sampling numbers given by

$$\bar{s}_m(K, \rho)_X := \text{Exp}_X \sup_{f \in K} R(K_{f(\mathbf{x})})_X. \quad (7.3)$$

## 8 Examples

The main objective of this paper is to describe the optimal performance that is possible for a learning procedure 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  $\Sigma_n$  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 finite-dimensional 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 finite-dimensional 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  $\varepsilon \rightarrow 0^+$ . This in turn depends very much on the particular  $K$  and requires an ad-hoc analysis depending on  $K$ . We describe a typical way to proceed in the setting of Theorem 3.1. The first step is to construct a sequence of spaces  $\{\Sigma_n\}_n$  for which  $\text{dist}(K, \Sigma \cap K)_X \lesssim n^{-r}$  for some  $r > 0$ . The next step is to provide bounds for the Chebyshev radius and the inflated Chebyshev radii corresponding to the data  $w = \lambda(f) \in \mathbb{R}^m$ . Typically we prove an estimate like  $R(K(w, \varepsilon))_X \leq R(K_w)_X + \varepsilon$ . In that case, according to the theorem, we need  $n \gtrsim [R(K_w)_X]^{-2/r}$  and a value  $\mu \asymp n^{-r/2}$  in the loss function (3.2) to guarantee that  $\hat{f}$  is a near optimal recovery of  $f$ .

In order to illustrate what is involved in such an analysis, 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,  $1 \leq q \leq \infty$ . This is an extensively studied setting in IBC. The texts [29, 37] 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 [21] and the references in that paper.

For our model classes, we consider the unit ball  $K := U(W^s(L_p(\Omega))), s > 0, 1 < p \leq \infty$ , 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  $\Omega$  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 Sect. 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 [22, 28]. 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 [28]). Additionally, it is known that  $m$  random sample sites are near optimal save for a possible logarithm [21]. Procedures for near optimal recovery are known using quasi-interpolants (see [21]).

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

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

with  $\mu$  chosen sufficiently small. According to Sect. 6, we may also obtain near optimal performance by using the modified loss

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

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

### 8.1.1 Analysis

As a starting point for our analysis, let us recall the following known lemma.

**Lemma 8.1** *Let  $1 \leq p \leq \infty$ ,  $n \geq 2$ , and  $0 = \xi_1 < \xi_2 < \dots < \xi_n = 1$ . Then given any  $f$  with  $f' \in L_p[0, 1]$ , the piecewise linear function  $S$  which interpolates  $f$  at the points  $\xi_1, \dots, \xi_n$ , and has breakpoints only at these points, satisfies the inequalities:*

- (i)  $\|S'\|_{L_p[0,1]} \leq \|f'\|_{L_p[0,1]}$ ,
- (ii)  $\|f - S\|_{L_2[0,1]} \leq \|f'\|_{L_p[0,1]} h^s$ , where  $h := \max_{1 \leq j < n} |\xi_{j+1} - \xi_j|$  and  $s := 1 - (1/p - 1/2)_+$ .

**Proof** For notational convenience, we assume  $p < \infty$ . The same proof holds when  $p = \infty$ . Let  $I_j := [\xi_j, \xi_{j+1}]$  and let  $\mu_j$  be the slope of  $S$  on  $I_j$ . Then, we have

$$\begin{aligned} |\mu_j| &= \frac{1}{|I_j|} \left| \int_{I_j} f' \right| \leq |I_j|^{-1} \left[ \int_{I_j} |f'|^p \right]^{1/p} |I_j|^{1-1/p} \\ &= |I_j|^{-1/p} \left[ \int_{I_j} |f'|^p \right]^{1/p}, \quad 1 \leq j < n, \end{aligned} \quad (8.3)$$

and thus

$$|\mu_j|^p \leq |I_j|^{-1} \int_{I_j} |f'|^p.$$

This means that  $\int_{I_j} |S'|^p \leq \int_{I_j} |f'|^p$ ,  $1 \leq j < n$ , and (i) follows.

To prove (ii), we note that  $f - S$  vanishes at each  $\xi_j$ ,  $1 \leq j \leq n$ , and therefore we have

$$\|f - S\|_{L_\infty(I_j)} \leq \int_{I_j} |f' - S'| \leq \int_{I_j} |f'| \leq \left[ \int_{I_j} |f'|^p \right]^{1/p} |I_j|^{1-1/p}, \quad 1 \leq j < n. \quad (8.4)$$

Here, we used the fact that on  $I_j$ ,  $S' = \mu_j = \frac{1}{|I_j|} \int_{I_j} f'$  is the best  $L_1(I_j)$  approximation to  $f'$  by constants. Using (8.4), we obtain

$$\begin{aligned} \|f - S\|_{L_2[0,1]}^2 &= \sum_{j=1}^{n-1} \int_{I_j} |f - S|^2 \leq \sum_{j=1}^{n-1} \left[ \int_{I_j} |f'|^p \right]^{2/p} |I_j|^{2-2/p+1} \\ &\leq h^{3-2/p} \sum_{j=1}^{n-1} \left[ \int_{I_j} |f'|^p \right]^{2/p}. \end{aligned} \quad (8.5)$$

If  $p \leq 2$ , the last sum is bounded by  $\|f'\|_{L_p[0,1]}^2$  because an  $\ell_{2/p}$  norm is bounded by an  $\ell_1$  norm, and we arrive at

$$\|f - S\|_{L_2[0,1]}^2 \leq h^{3-2/p} \|f'\|_{L_p[0,1]}^2.$$

This gives (ii) when  $p \leq 2$ . The case  $p \geq 2$  in (ii) follows from the case  $p = 2$ .  $\square$

**Remark 8.2** Notice that from the bound (8.4) we get  $\|f - S\|_{L_\infty[0,1]} \leq h^{1-1/p} \|f'\|_{L_p[0,1]}$ . We will use this inequality later in this paper.

We take  $X = L_2[0, 1]$  as the space in which we measure the error of performance. To describe in a bit more detail one simple example, we consider the univariate case  $d = 1$  and

$$K = \{f \in W^1(L_p[0, 1]) : \|f\|_{W^1(L_p[0, 1])} \leq 1\}, \quad 1 < p \leq \infty,$$

where

$$\|f\|_{W^1(L_p[0, 1])} := \max\{\|f\|_{L_p[0, 1]}, \|f'\|_{L_p[0, 1]}\}.$$

Let us return to our problem of near optimal recovery of a function in  $K$  from its point values  $w_j = f(x_j)$ ,  $j = 1, \dots, m$ . For convenience, we assume that the endpoints 0, 1 are always data sites and  $w_1 = 0$ . Note that in this case

$$K_w = \{f \in W^1(L_p[0, 1]) : f(x_j) = w_j, j = 1, \dots, m, \|f'\|_{L_p[0, 1]} \leq 1\}.$$

Let  $\Xi_n$  be the union of the point sets  $\{x_1, \dots, x_m\}$  and  $\{0, 1/n, \dots, 1\}$  and let  $\Sigma_n$  be the linear space of piecewise linear functions subordinate to  $\Xi_n$ . Thus, we are in the setting of Theorem 4.4. The above remark tells us that if we choose  $\Sigma = \Sigma_n$  with  $n$  large enough then  $\Sigma$  satisfies

$$\text{dist}(K, \Sigma_n \cap K)_{C[0,1]} \leq n^{-1+1/p}, \quad n \geq 1. \quad (8.6)$$

This means that the hypothesis of Theorem 4.4 are satisfied for  $p > 1$  and so solving the optimization problem (4.8) gives us a near optimal recovery provided  $n$  is sufficiently large. We want to see how large we need to take  $n$  but before doing that we make the following remark.

**Remark 8.3** We know from Lemma 8.1 that the piecewise linear function  $S$  which interpolates the data is in  $K_w$  and therefore is itself a near optimal recovery with constant  $C = 2$ . In other words, in this very special case, using an ad hoc analysis we can avoid solving a minimization problem and simply take the piecewise linear interpolant to the data. Results of this special type are referred to as representer theorems and are preferred over minimization of loss functions when such representer theorems are known, see for e.g. [38]. The results of the present article apply when representer theorems are not known.

We proceed as if a representer theorem is not known to us and instead we apply Theorem 4.4. We want to see how large we would need to take  $n$  and how small we have to take  $\mu$ . For this, we need to give good estimates for  $R(K_w)_{L_2[0,1]}$  and  $R(K(w, 2\varepsilon))_{L_2[0,1]}$ ,  $\varepsilon > 0$ . We continue using the above notation, in particular for  $h(\mathbf{x}) := \max_{1 \leq j < m} |x_{j+1} - x_j|$ ,  $S_w$  for the piecewise linear interpolant to the data  $w$ , and  $I_j = [x_j, x_{j+1}]$ ,  $1 \leq j < m$ . For notational convenience, we only consider the case  $1 < p \leq 2$ .

**Lemma 8.4** *If  $1 < p \leq 2$ ,  $s = 3/2 - 1/p$ , and  $R := R(K_w)_{L_2[0,1]} > 0$ , then we have  $\|S'_w\|_{L_p[0,1]} < 1$  and*

$$\Lambda h(\mathbf{x})^s \leq R \leq h(\mathbf{x})^s, \quad \Lambda := 1 - \|S'_w\|_{L_p[0,1]}, \quad (8.7)$$

and

$$R(K(w, \varepsilon))_{L_2[0,1]} \leq h(\mathbf{x})^s + \varepsilon \sqrt{mh(\mathbf{x})}. \quad (8.8)$$

**Proof** We know from Lemma 8.1 applied for the points  $0 = x_1 < \dots < x_m = 1$  that  $\|S'_w\|_{L_p[0,1]} \leq 1$  and  $S_w \in K_w$ . We first want to prove that if  $R > 0$ , then we necessarily have  $\Lambda > 0$ . Indeed, if  $R > 0$ , then there is a  $g \in K_w$  such that  $\|g - S_w\|_{L_2[0,1]} > 0$  and  $\|g'\|_{L_p[0,1]} \leq 1$ . The function  $\frac{1}{2}(g + S_w)$  also is in  $K_w$ . Since  $p > 1$ , then the strict convexity of the  $L_p$  ball implies that

$$\frac{1}{2}\|g' + S'_w\|_{L_p[0,1]} < \frac{1}{2}(\|g'\|_{L_p[0,1]} + \|S'_w\|_{L_p[0,1]}) \leq 1.$$

From (i) of Lemma 8.1 (applied for  $f = \frac{1}{2}(g + S_w)$ ) we derive that  $\|S'_w\|_{L_p[0,1]} < 1$  and hence  $\Lambda > 0$ .

Now, consider (8.7). The upper inequality follows from Lemma 8.1 with  $m = n$  and  $\xi_j = x_j$ ,  $1 \leq j < m$ . We next prove the lower inequality. Let  $I = I_j$  be an interval  $[x_j, x_{j+1}]$  with  $|I_j| = h(\mathbf{x})$  and let  $I_-$  be the left half of  $I$  and  $I_+$  be the right half of  $I$ . We define  $g' := c[\chi_{I_+} - \chi_{I_-}]$  with  $c := \Lambda|I|^{-1/p}$ , where  $\chi_I$  denotes the

characteristic function of an interval  $J$ . The functions  $S_w \pm g$  are both in  $K_w$  and so  $R \geq \|g\|_{L_2[0,1]}$ . Now  $|g|$  is larger than  $c|I|/4$  on the middle half of  $I$ . Therefore

$$\|g\|_{L_2[0,1]} \geq c(|I|/4)(|I|/2)^{1/2} \geq \frac{1}{4\sqrt{2}}\Lambda|I|^{3/2-1/p} = \frac{1}{4\sqrt{2}}\Lambda h(\mathbf{x})^s.$$

This proves the lower inequality in (8.7).

Next, we prove (8.8). Let  $S_w$  and  $S_{w'}$  be the piecewise linear interpolants for data  $w$  and  $w'$ , respectively. If  $\|w - w'\| \leq \varepsilon$ , then

$$\|S_w - S_{w'}\|_{L_2[0,1]}^2 \leq h(\mathbf{x}) \sum_{j=1}^m |w_j - w'_j|^2 \leq h(\mathbf{x})m\varepsilon^2.$$

Now, if  $f \in K(w, \varepsilon)$  and  $\lambda(f) = w'$ , then

$$\|f - S_w\|_{L_2[0,1]} \leq \|f - S_{w'}\|_{L_2[0,1]} + \|S_w - S_{w'}\|_{L_2[0,1]} \leq h(\mathbf{x})^s + \varepsilon\sqrt{h(\mathbf{x})m}, \quad (8.9)$$

where we used Lemma 8.1. This proves (8.8) and completes the proof of the lemma.  $\square$

We use the above lemma to see how large we have to take  $n$  and how to choose  $\mu$  in Theorem 4.4 in order to guarantee that  $\hat{f}$  is a near optimal recovery with constant  $C$  in the  $X = L_2[0, 1]$  norm. Note that  $C_X = 1$  in this case. We need that  $C$  is large enough and  $\mu$  is small enough so that

$$\varepsilon + 2R(K(w, 2\varepsilon))_{L_2[0,1]} \leq CR(K_w)_{L_2[0,1]}, \quad \varepsilon := \mu \max(\mu + 1, C_Y). \quad (8.10)$$

The lemma gives the bounds  $R(K_w)_{L_2[0,1]} \geq \Lambda h(\mathbf{x})^s$  and  $R(K(w, 2\varepsilon))_{L_2[0,1]} \leq h(\mathbf{x})^s + 2\varepsilon\sqrt{mh(\mathbf{x})}$ . We see that (8.10) holds provided

$$\varepsilon \leq h(\mathbf{x})^{s-\frac{1}{2}}m^{-\frac{1}{2}} \quad \text{and} \quad C = 7\Lambda^{-1}, \quad (8.11)$$

since  $h(\mathbf{x}) \geq 1/m$ . It is enough to take  $\varepsilon \leq m^{-s}$  (since  $s > 1/2$ ) and similarly  $\mu \leq m^{-s}$ . Looking back at Theorem 4.4, we need the approximation error  $\delta$  (as measured in  $C(\Omega)$ ) to satisfy  $\delta \leq \mu^2$ . Since the approximation accuracy in  $C(\Omega)$  is  $O(n^{-1+1/p})$ , we need

$$n \geq m^{\frac{2s}{1-1/p}} \quad \text{and} \quad \mu \leq m^{-s}$$

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  $\Sigma_n$  must approximate  $K$  and how we must choose  $\mu$  so that solving the minimization problem gives a near optimal recovery from the given data.

### 8.1.2 Numerical experiments

We next discuss the numerical implementation of the optimization with the loss (8.2) for this special  $K = U(W^1(L_p[0, 1]))$ . We consider  $\Sigma_n$  to be the space of continuous piecewise linear functions with breakpoints  $\xi_j = j/n$ ,  $0 \leq j \leq n$ . We can parameterize  $\Sigma_n$  using the hat function basis  $H_j$ ,  $j = 0, \dots, n$ , where  $H_j$  is the continuous piecewise linear function which takes the value one at  $\xi_j$  and the value 0 at all other  $\xi_i$ ,  $i \neq j$ . Then each  $g \in \Sigma_n$  can be written as

$$g(x) = g_{\mathbf{c}}(x) := \sum_{j=0}^n c_j H_j(x), \quad x \in [0, 1], \quad (8.12)$$

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

$$\mathcal{L}^*(\mathbf{c}) := \mathcal{L}_\mu(g_{\mathbf{c}}), \quad \mathbf{c} \in \mathbb{R}^n. \quad (8.13)$$

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  $\Sigma_n$  we minimize  $\mathcal{L}^*$  over  $\mathbb{R}^n$  and use the argument  $\mathbf{c}^*$  attaining this minimum to define the minimizer  $\hat{f} := g_{\mathbf{c}^*}$ . 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 [4]).

As a numerical example, we take

$$f(x) = \frac{1}{4}x^{\frac{1}{2}}, \quad x \in [0, 1]. \quad (8.14)$$

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

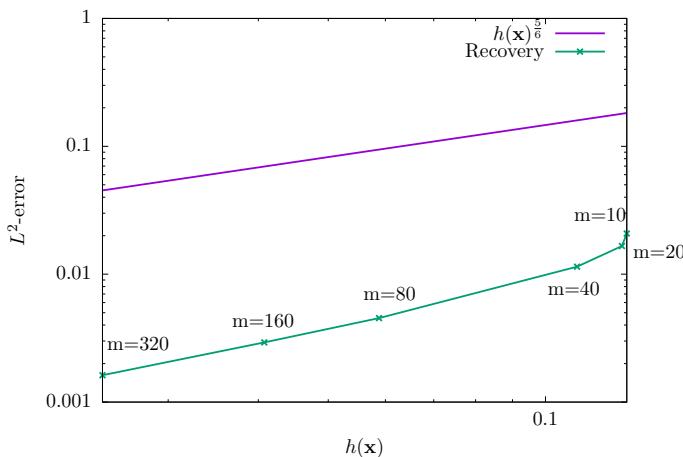
$$K = U(W^1(L_p[0, 1])), \quad p = 3/2. \quad (8.15)$$

This gives that  $s = 5/6$ . While we can implement the 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  $h$

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

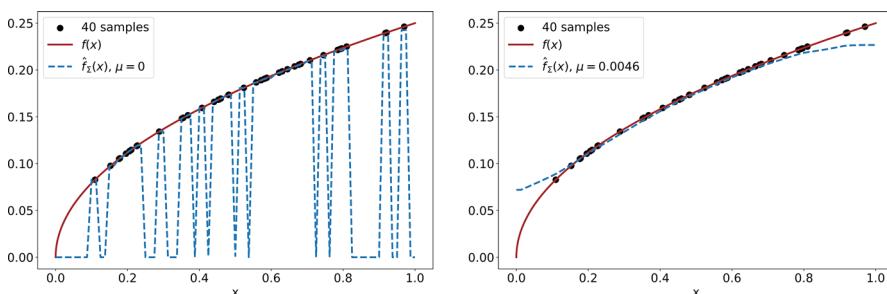
For each of these values of  $m$ , we choose  $n = 2m$  and  $\mu = 0.1m^{-s}$ . Note that this choice of  $n$  is less than suggested by the theoretical estimates.



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

Figure 2 gives a graph of the true recovery rate and compares it with the bound  $h(x)^{\frac{5}{6}}$  which is our bound for the Chebyshev radius of  $K_w$  and hence optimal recovery rate for these data observations. We observe an asymptotic decay better than  $h(x)^s$  (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 [11]). We take the initial parameter choice as zero as we did in the case of a penalty term. Figure 3 compares the minimizing  $\hat{f}$  for the



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

case of  $m = 40$  without regularization ( $\mu = 0$ ) and with our proposed regularization ( $\mu = 0.0046$ ). For the former, the over-parametrized procedure produces a highly oscillating  $\hat{f}$  that interpolates the data samples. In contrast, the constructed  $\hat{f}$  for  $\mu = 0.0046$  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  $\mu = 0$ .

## 8.2 Neural networks

It is now quite common in learning to take  $\Sigma_n$  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  $\Omega$  be the unit Euclidean ball in  $\mathbb{R}^d$  with  $d \geq 1$ . We consider the nonlinear space  $\Sigma_n$  of outputs of a single hidden layer ReLU neural network of width  $n$  on  $\Omega$  (much less is known for deeper networks). Each function  $g \in \Sigma_n$  is of the form

$$g(x) = c_0 + \sum_{j=1}^n c_j(\omega_j \cdot x + b_j)_+, \quad (8.16)$$

where  $\omega_j \in \mathbb{R}^d$ , and the  $c_j, b_j \in \mathbb{R}$ . This representation of  $g$  is not unique. We can require  $\omega_j$  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  $\Sigma_n$  is a nonlinear space of continuous piecewise linear functions on  $\Omega$ .

It is commonly thought that  $\Sigma_n$  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  $\Sigma_n$ . The most reknown of these is the Barron class introduced in [5] defined via Fourier transforms. Several generalization of these classes (see e.g. [14, 30, 34]) 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  $L_q(\Omega)$ ,  $1 \leq q \leq \infty$ , with an approximation rate  $O(n^{-\alpha})$ ,  $n \rightarrow \infty$ , with  $\alpha \geq 1/2$ , 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 [11, 34] for results on the approximation of functions in  $K$  by the elements of  $\Sigma_n$ .

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  $L_q$ ,  $1 \leq q \leq \infty$ , 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 [35] 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  $f \in K$ . Some progress has been made recently in [40], where upper and lower bounds for sampling numbers for the smooth Barron classes have been obtained. 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  $\Sigma_n$ ) to obtain optimal learning.

When using over-parameterized neural networks to solve the finite-dimensional 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 (8.16). It is shown in [30] that there is always a minimizer which has a sparse representation (8.16).

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  $\Sigma_n$  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 [30].

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 procedure to find an  $\hat{f}$  corresponding to least squares loss. The typical setting employs over-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 [17, 20]). 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.

**Acknowledgements** The authors thank Professor Albert Cohen for helpful discussions on the research in this paper. The authors also thank the referees for their excellent comments that improved this manuscript. The work of PB was partially supported by the NSF Grants DMS 1720297 and DMS 2038080. The work of AB was partially supported by the NSF Grant DMS 2110811. The work of RD and GP was partially supported by the ONR Contract N00014-20-1-278, the NSF Grant DMS 2134077, and the NSF-Tripods Grant CCF-1934904.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

1. Adcock, B., Bao, A., Brugiapaglia, S.: Correcting for unknown errors in sparse high-dimensional function approximation. *Numer. Math.* **142**(3), 667–711 (2019)
2. Adcock, B., Brugiapaglia, S., Dexter, N., Moraga, S.: Deep neural networks are effective at learning high-dimensional Hilbert-valued functions from limited data. In: Bruna, J., Hesthaven, J.S., Zdeborova, L. (eds.) *Proceedings of The Second Annual Conference on Mathematical and Scientific Machine Learning*, vol. 145, pp. 1–36 (2021)
3. Adcock, B., Brugiapaglia, S., Webster, C.: *Sparse Polynomial Approximation of High-Dimensional Functions in Computer Science Engineering*. Society for Industrial and Applied Mathematics, Philadelphia (2022)
4. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. *Pac. J. Math.* **16**(1), 1–3 (1966)
5. Barron, A.: Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory* **39**(3), 930–945 (1993)
6. Belloni, A., Chernozhukov, V., Wang, L.: Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika* **98**(4), 791–806 (2011)

7. Bickel, P., Ritov, Y., Tsybakov, A.: Simultaneous analysis of lasso and Dantzig selector. *Ann. Stat.* **37**(4), 1705–1732 (2009)
8. Bunea, F., Lederer, J., She, Y.: The group square-root lasso: theoretical properties and fast algorithms. *IEEE Trans. Inf. Theory* **60**(2), 1313–1325 (2014)
9. Cohen, A., Dahmen, W., DeVore, R.: Compressed sensing and best  $k$ -term approximation. *J. Am. Math. Soc.* **22**(1), 211–231 (2009)
10. Cohen, A., Davenport, M., Leviatan, D.: On the stability and accuracy of least squares approximations. *Found. Comput. Math.* **13**(5), 819–834 (2013)
11. DeVore, R., Hanin, B., Petrova, G.: Neural network approximation. *Acta Numer.* **30**, 327–444 (2021)
12. DeVore, R., Petrova, G., Wojtaszczyk, P.: Data assimilation and sampling in Banach spaces. *Calcolo* **54**(3), 963–1007 (2017)
13. Donoho, D.: Compressed sensing. *IEEE Trans. Inf. Theory* **52**(4), 1289–1306 (2006)
14. E, Ma, C., Wu, L.: Barron spaces and the compositional function spaces for neural network models. arXiv preprint [arXiv:1906.08039](https://arxiv.org/abs/1906.08039) (2019)
15. Foucart, S.: The sparsity of LASSO-type minimizers. *Appl. Comput. Harmon. Anal.* **62**, 441–452 (2023)
16. Foucart, S., Rauhut, H.: An invitation to compressive sensing. In: *A Mathematical Introduction to Compressive Sensing*, pp. 1–39. Springer, New York (2013)
17. Hanin, B., Nica, M.: Finite depth and width corrections to the neural tangent kernel. arXiv preprint [arXiv:1909.05989](https://arxiv.org/abs/1909.05989) (2019)
18. Hastie, T., Tibshirani, R., Tibshirani, R.: Best subset, forward stepwise or lasso? Analysis and recommendations based on extensive comparisons. *Stat. Sci.* **35**(4), 579–592 (2020)
19. Hastie, T., Tibshirani, R., Wainwright, M.: *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, Boca Raton (2015)
20. Jacot, A., Gabriel, F., Hongler, C.: Neural tangent kernel: convergence and generalization in neural networks. *Adv. Neural Inf. Process. Syst.* **31** (2018)
21. Krieg, D., Novak, E., Sonnleitner, M.: Recovery of Sobolev functions restricted to iid sampling. arXiv preprint [arXiv:2108.02055](https://arxiv.org/abs/2108.02055) (2021)
22. Krieg, D., Sonnleitner, M.: Random points are optimal for the approximation of Sobolev functions. arXiv preprint [arXiv:2009.11275](https://arxiv.org/abs/2009.11275) (2020)
23. Krieg, D., Ullrich, M.: Function values are enough for  $L_2$ -approximation. *Found. Comput. Math.* **21**(4), 1141–1151 (2021)
24. Lorentz, G., Golitschek, M., Makovoz, Y.: *Constructive Approximation: Advanced Problems*. Springer, New York (1996)
25. Meinshausen, N., Yu, B.: Lasso-type recovery of sparse representations for high-dimensional data. *Ann. Stat.* **37**(1), 2246–2270 (2009)
26. Micchelli, C., Rivlin, T.: A survey of optimal recovery. In: *Optimal Estimation in Approximation Theory*, pp. 1–54 (1977)
27. Nagel, N., Schäfer, M., Ullrich, T.: A new upper bound for sampling numbers. *Found. Comput. Math.* **22**, 445–468 (2021)
28. Narcowich, F., Ward, J., Wendland, H.: Sobolev bounds on functions with scattered zeros, with applications to radial basis function surface fitting. *Math. Comput.* **74**(250), 743–763 (2005)
29. Novak, E., Wozniakowski, H.: *Tractability of Multivariate Problems*. Vol. I: Linear Information. European Mathematical Society, Helsinki (2008)
30. Parhi, R., Nowak, R.: Banach space representer theorems for neural networks and ridge splines. *J. Mach. Learn. Res.* **22**(43), 1–40 (2021)
31. Petersen, H., Jung, P.: Robust instance-optimal recovery of sparse signals at unknown noise levels. *Inf. Inference* **11**, 845–887 (2022)
32. Petersen, P., Voigtlaender, F.: Optimal learning of high-dimensional classification problems using deep neural networks. arXiv preprint [arXiv:2112.12555](https://arxiv.org/abs/2112.12555) (2021)
33. Pinkus, A.: *N-widths in Approximation Theory*, vol. 7. Springer Science and Business Media, New York (2012)
34. Siegel, J., Xu, J.: Characterization of the variation spaces corresponding to shallow neural networks. arXiv preprint [arXiv:2106.15002](https://arxiv.org/abs/2106.15002) (2021)
35. Siegel, J., Xu, J.: Sharp bounds on the approximation rates, metric entropy, and  $n$ -widths of shallow neural networks. arXiv preprint [arXiv:2101.12365](https://arxiv.org/abs/2101.12365) (2021)

36. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* **58**(1), 267–288 (1996)
37. Traub, J., Wozniakowski, H.: *A General Theory of Optimal Algorithms*. Academic Press, London (1980)
38. Unser, M.: A unifying representer theorem for inverse problems and machine learning. *Found. Comput. Math.* **21**(4), 941–960 (2021)
39. van de Geer, S.: *Estimation and Testing Under Sparsity*. Lecture Notes in Mathematics. Springer, Berlin (2016)
40. Voigtlaender, F.:  $L^p$  sampling numbers for the Fourier-analytic Barron space. arXiv preprint [arXiv:2208.07605v1](https://arxiv.org/abs/2208.07605v1) (2022)
41. Yosida, K.: *Functional Analysis*. Springer Science and Business Media, New York (2012)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.