Proceedings of Machine Learning Research vol 65:1–27, 2017

The Sample Complexity of Optimizing a Convex Function

Eric Balkanski ERICBALKANSKI@G.HARVARD.EDU

Yaron Singer YARON@SEAS.HARVARD.EDU

Abstract

In this paper we study optimization from samples of convex functions. There are many scenarios in which we do not know the function we wish to optimize but can learn it from data. In such cases, we are interested in bounding the number of samples required to optimize the function. Our main result shows that in general, the number of samples required to obtain a non-trivial approximation to the optimum of a convex function is exponential in its dimension, even when the function is PAC-learnable. We also obtain strong lower bounds for strongly convex and Lipschitz continuous functions. On the positive side, we show that there are interesting classes of functions and distributions for which the sample complexity is polynomial in the dimension of the function.

Keywords: Convex optimization, PAC learning, sample complexity

1. Introduction

In this paper we consider the problem of optimizing a convex function from training data. The traditional approach in optimization assumes that the algorithm designer either knows the function or has access to an oracle that allows evaluating the function. In many cases, however, we do not know the true function we aim to optimize, but can observe its behavior.

One example is when we aim to find a route that minimizes travel time between two locations in a city. One reasonable approach is to collect data on times traveled on routes in the city, construct a weighted graph which represents congestion between intersections, and find a shortest path on the weighted graph. A different application is that of influence maximization, where the goal is to select a small set of individuals that can initiate a large cascade in a social network. A reasonable approach here is to collect data about past cascades, fit the observations to a model that predicts influence between individuals, and find the set of individuals whose selection will generate the largest cascade according to the learned model. Finally, we can imagine a federal agency interested in reducing greenhouse emissions by taxing gas and subsidizing public transportation. To do so, the agency can collect data on gas prices, public transport costs, and greenhouse emissions from a large number of cities and decide on taxes and subsidies by optimizing a function learned from data.

In all the above examples, optimizing the function learned from data seems like a natural approach. The crux however is that when we optimize a function learned from data, the guarantees on optimization apply to the function learned from data, and not the true function we aim to optimize. The following example shows that this approach can actually lead to arbitrarily bad approximations.

1.1. Optimizing a function learned via ERM can lead to an arbitrarily bad approximation

A natural approach to optimize a function $f : [0,1]^n \to [0,1]$ from samples is to observe samples, learn a surrogate $\tilde{f}$, and return a solution $\tilde{\mathbf{x}} \in \operatorname{argmin}_{\mathbf{x}} \tilde{f}(\mathbf{x})$. While this seems reasonable, it can easily result in an arbitrarily bad approximation. To illustrate this, consider the canonical *Empirical Risk Minimization* (ERM) approach to learning the following convex function:

$$f(\mathbf{x}) = \max\left(0, 1 - \mathbb{1}_{[n/2]}^{\intercal}\mathbf{x}\right),$$

where $\mathbb{1}_{[n/2]}$ denotes the $\{0,1\}$-vector with ith entry equal to 1 if $i \in [n/2] = \{1, \ldots, n/2\}$, and 0 otherwise. It is easy to see that for samples $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^m$ when $\{\mathbf{x}_i\}_{i=1}^m$ are independently drawn from the uniform distribution over $[0,1]^n$, we have that with high probability $f(\mathbf{x}_i) = 0$ for all m samples, when m is polynomial in n. A surrogate learned via ERM for these samples can be:

$$\tilde{f}(\mathbf{x}) = \max\left(0, 1 - \mathbb{1}_{\{n/2+1,\ldots,n\}}^{\intercal}\mathbf{x}\right)$$

since $\tilde{f}(\mathbf{x}_i) = 0$ with high probability for all m samples as well. Note that a minimizer of $\tilde{f}$ is $\tilde{\mathbf{x}} = \mathbb{1}_{\{n/2+1,\ldots,n\}}$, but that $f(\tilde{\mathbf{x}}) = 1$ whereas the optimal solution is $f(\mathbf{x}) = 0$, where $\mathbf{x}$ can be obtained by any point $\mathbf{x}_i$ from the samples. The moral is that learning a surrogate via ERM and optimizing it can result in an arbitrarily poor approximation, *even when the problem can be easily optimized from samples*.[1]

1.2. Optimization from samples

In cases where the function is not known but where sampled data is available, our problem becomes that of *optimization from samples*. A similar framework was recently introduced in the context of submodular functions (Balkanski et al., 2017) (see Section 1.5 for further discussion).

Definition 1 *A class of functions $\mathcal{F}$ is ϵ-optimizable from samples over distribution $\mathcal{D}$ if for every $f \in \mathcal{F}$ and $\delta \in (0,1)$, given $\operatorname{poly}(n, 1/\delta, 1/\epsilon)$ i.i.d. samples $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^m$ where $\mathbf{x}_i \sim \mathcal{D}$, there exists an (not necessarily polynomial time) algorithm that returns a solution $\tilde{\mathbf{x}}$ such that, with probability at least $1 - \delta$ over the samples,*

$$f(\tilde{\mathbf{x}}) - \min_{\mathbf{x}} f(\mathbf{x}) \leq \epsilon.$$

We say that $\mathcal{F}$ is optimizable from samples if there exists a distribution $\mathcal{D}$ such that $\mathcal{F}$ is ϵ-optimizable from samples over $\mathcal{D}$ for all $\epsilon > 0$. [2] As we later show there are interesting classes of functions that can be optimized from polynomially-many samples using polynomial-time algorithms. The question is therefore not whether one can optimize a function from samples, but rather what are the classes of functions for which this is possible. We will consider the class of convex functions, as we know these functions can be efficiently optimized within an arbitrary degree of precision. Per our above discussion on optimizing a surrogate function learned from samples, it seems reasonable to further our restriction to classes that are PAC-learnable.

Can convex functions that are PAC learnable be optimized from training data?

1. Note that it is not true that *any* ERM method leads to an arbitrarily poor approximation. An algorithm that returns a surrogate $\tilde{f} = f$ is trivially an ERM and the optima of the surrogate are an optimal approximation to the true optima. Our impossibility result implies that such an ERM algorithm would require observing exponentially-many samples.
2. Notice that we can never hope to obtain optimization from samples for *every* distribution. The distribution which always returns the same point for example, will never allow optimization from samples.

1.3. Main result

The main result in this paper is an impossibility. We show that there is a class of convex functions that is PAC-learnable and that cannot be optimized from samples. Notice that a consequence of this statement is that there is no learning algorithm that can produce a surrogate whose optimum is a reasonable approximation to the true optimum. More formally, we prove the following theorem.

Theorem *There exists a class of convex functions $\mathcal{F}$, where $f : [0,1]^n \to [0,1]$ for $f \in \mathcal{F}$, that is (ϵ, δ)-PAC learnable and such that, for any distribution $\mathcal{D}$, no algorithm can obtain an approximation strictly better than $1/2 - o(1)$ for the problem of minimizing f, for all $f \in \mathcal{F}$, given $\mathrm{poly}(n)$ i.i.d. samples of f drawn from $\mathcal{D}$, with probability $1/\mathrm{poly}(n)$ over the samples.*

The lower bound (Section 2) is information-theoretic and is not due to computational constraints. This result is tight since finding a $1/2$ approximation is trivial in this domain. We also show lower bounds for strongly convex and Lipschitz continuous convex functions. In Section 3, we show that this class of functions is PAC-learnable. On the positive side, it is not difficult to show that linear functions and polynomials can be optimized from samples under some assumption (Section 4).

1.4. Technical overview

At a high level, we construct a family of convex functions $\mathcal{F}$ where each function is defined by some partition P of $[n] = \{1, \ldots, n\}$, and its value crucially depends on P. The inapproximability comes from showing that polynomially-many samples do not contain enough information to learn the partition that defines the function generating the samples. We then argue that any non-trivial optimization guarantee is impossible to obtain when an algorithm cannot learn the partition.

The class of functions that we construct and the intuition for why these functions are hard to optimize from samples are relatively simple. The challenging part of the analysis is in the design of interdependent conditions that allow us to formally show that the impossibility result holds over *any* distribution generating the samples. We state three conditions called indistinguishability, gap, and balance which rely on a *randomized* collection of partitions, instead of a random partition from a fixed collection. This collection of partitions must then be constructed carefully so that the three conditions are satisfied.

Another interesting technical challenge is in showing that the class of functions we use for the lower bound is PAC learnable. To do so, we develop a learning algorithm which combines classification and linear regression. The labels of the samples needed for the classification step are not known and we develop two techniques to get around this issue.

1.5. Related work

Optimization from samples of submodular functions. The question of optimization from samples was recently introduced in the context of submodular functions (Balkanski et al., 2016, 2017). In general, for the canonical problem of maximizing a monotone submodular function under a cardinality constraint, it is impossible to obtain any reasonable approximation given polynomially-many samples drawn from any distribution (Balkanski et al., 2017). On the positive side, when the functions have bounded curvature, good approximations are achievable (Balkanski et al., 2016).

Although there are interesting connections between convexity and submodularity, the differences remain apparent in this work. On a conceptual level, it is well known that maximization of a monotone submodular function under a cardinality constraint and unconstrained maximization of a non-monotone submodular function require exponentially-many value queries. In addition, the notion of learnability for submodular functions is that of PMAC learnability (Balcan and Harvey, 2011; Feldman et al., 2013; Feldman and Kothari, 2014) which produces a surrogate that approximates a submodular function within some constant-factor approximation. Hassidim and Singer (2016) show that obtaining a constant factor approximation for maximization with such a surrogate requires exponentially-many queries. For these reasons, impossibility results for optimization from samples may seem more plausible in the submodular case than in the convex one. In particular, since learning is often cast as a convex optimization problem, impossibility results for optimization from samples seem counter-intuitive.

From a technical perspective, in comparison to (Balkanski et al., 2017), the differences arise from the departure from constrained monotone optimization to unconstrained non-monotone optimization, which requires novel functions and partitions. The novelty in the partitions is that we construct a randomized collection of partitions instead of having a fixed collection. Finally, note that there is a trivial $1/2$ upper bound in the unconstrained case whereas the lower bound is arbitrarily close to 1 in the constrained case.

Online learning and stochastic convex optimization. The goal of online learning and stochastic convex optimization is also to optimize a function given past observations. The main difference with online learning is that a unique decision has to be made offline given a collection of observations instead of online after each observation. The variant of online learning where the loss functions are convex, online convex optimization, has attracted a lot of attention (Shalev-Shwartz et al., 2012; Hazan et al., 2007; Hazan, 2015). Even more closely related is bandit convex optimization, where the feedback at each iteration consists only of the value of the decision instead of the entire convex function (Flaxman et al., 2005; Bubeck and Eldan, 2015; Agarwal et al., 2010). Although online learning is tailored to many applications, optimization from samples is relevant in situations where the observations are either not the result of previous optimization attempts or only observed after a long period of time. In stochastic convex optimization, the goal is to minimize a convex function that is the expected value of a random objective f drawn from some unknown distribution given samples $f_1, \ldots, f_m$ from the distribution (Shalev-Shwartz et al., 2009; Feldman, 2016).

PAC-learning and convex optimization. The problem of learning a function given sampled data of this function was formalized by Valiant (1984) in a seminal paper as PAC-learning. Given oracle access to a convex function, there exists a myriad of algorithms to optimize this function. In some sense, optimization from samples synthesizes such learning and optimization by aiming to approximately minimize a convex function given sampled data of this function.

Information-theoretic lower bounds. Information-theoretic lower bounds have been shown in the value query model for various settings (Mirrokni et al., 2008; Vondrák, 2013; Ene et al., 2013). Informally, a main step of the argument of these lower bounds goes by contradiction and relies on the algorithm being able to make an additional query if it was able to optimize the function. In the samples model, it is not possible to make an additional query and novel frameworks to construct lower bounds are needed.

2. Impossibility of convex optimization from samples

In this section we show that there exists a class of convex functions that cannot be optimized from samples. As discussed above, the main idea is to create partitions of $[n]$ that can be used to define a class of functions whose optima crucially depend on the partitions. We begin by proving a general hardness lemma in Section 2.1. The lemma shows that when two specific technical conditions on a class of functions are met, then optimization from samples of such functions is impossible. In Section 2.2 we explicitly construct the class of functions and prove that the necessary conditions for the general hardness lemma are met. We then show that one can modify the construction so that functions in this class are strongly convex and Lipschitz continuous in Section 2.3.

2.1. A general hardness lemma

The hardness lemma shows that a family of functions $\mathcal{F}$ cannot be optimized from samples if there exists a *randomized* subfamily of functions $\mathcal{F}' \subseteq \mathcal{F}$ that satisfies two conditions. The first condition asserts that the functions in $\mathcal{F}'$ cannot be distinguished from samples with high probability. The second condition is that, with high probability, there exists no fixed solution $\mathbf{x}$ which performs well on all functions in $\mathcal{F}'$ simultaneously. By combining the two conditions, we show that for any algorithm, there exists at least one function $f \in \mathcal{F}$ such that the algorithm minimizes f poorly from samples.

Lemma 2 *Let $\mathcal{F}$ be a family of functions and $\mathcal{F}' = \{f_1, \ldots, f_m\} \subseteq \mathcal{F}$ be a* randomized *subfamily of these functions. Assume the following two conditions hold:*

1. ***Indistinguishability.*** *For all $\mathbf{x}$, with probability $1 - e^{-\Omega(n^{1/3})}$ over $\mathcal{F}'$, for all $f_i, f_j \in \mathcal{F}'$,*

$$f_i(\mathbf{x}) = f_j(\mathbf{x});$$

2. ***α-gap.*** *Let $\mathbf{x}_i^\star$ be a minimizer of f_i. With probability $1 - e^{-\Omega(n^{1/3})}$ over $\mathcal{F}'$, for all $\mathbf{x} \in [0,1]^n$,*

$$\mathop{\mathbb{E}}_{f_i \sim \mathcal{U}(\mathcal{F}')} \left[f_i(\mathbf{x}) - f_i\left(\mathbf{x}_i^\star\right) \right] \geq \alpha;$$

Then, $\mathcal{F}$ is not α-optimizable from strictly less than $e^{-\Omega(n^{1/3})}$ samples over any distribution $\mathcal{D}$.

Note that for the indistinguishability property to hold, we need to consider a randomized family of functions $\mathcal{F}'$. The ordering of the quantifiers for these two properties is crucial. They both hold with high probability over the randomization of $\mathcal{F}'$, but the gap is for all $\mathbf{x}$ simultaneously whereas the indistinguishability is for any individual $\mathbf{x}$.

Proof At a high level, by using the indistinguishability condition and switching from the randomization of $\mathcal{F}$ to the randomization of $\mathbf{x} \sim \mathcal{D}$, we claim that there exists $\mathcal{F}'$ such that $f^i(\mathbf{x}) = f^j(\mathbf{x})$ for all $f_i, f_j \in \mathcal{F}'$ with high probability over $\mathbf{x} \sim \mathcal{D}$. By a union bound this holds for all samples $\mathbf{x}$. Thus, for such a family of functions $\mathcal{F}' = \{f_1, \ldots, f_m\}$, the choices of an algorithm that is given samples from f_i for $i \in [m]$ are *independent* of i. By the α-gap condition, this implies that there exists at least one function f_i for which a solution $\mathbf{x}$ returned by the algorithm is at least α away from a minimizer of f_i.

We first claim that for any distribution $\mathcal{D}$, there exists a family of functions $F \subseteq \mathcal{F}$ such that

- with probability $1 - e^{-\Omega(n^{1/3})}$ over $\mathbf{x} \sim \mathcal{D}$, $f_i(\mathbf{x}) = f_j(\mathbf{x})$ for all $f_i, f_j \in F$, and
- for all $\mathbf{x} \in [0,1]^n$, $\mathbb{E}_{f_i \sim \mathcal{U}(F)} \left[f_i(\mathbf{x}) - f_i\left(\mathbf{x}_i^\star\right) \right] \geq \alpha$.

Let the indicator variable $\mathbb{1}_{I(\mathcal{F}',\mathbf{x})}$ indicate if $f_i(\mathbf{x}) = f_j(\mathbf{x})$ for all $f_i, f_j \in \mathcal{F}'$ and, for all $\mathbf{x}' \in [0,1]^n$, $\mathbb{E}_{f_i \sim \mathcal{U}(\mathcal{F}')} \left[f_i(\mathbf{x}') - f_i\left(\mathbf{x}_i^\star\right) \right] \geq \alpha$. Let $p = \inf_{\mathbf{x}} \mathbb{P}_{\mathcal{F}'}[I(\mathcal{F}', \mathbf{x})]$. Then, since $\sum_{F \subseteq \mathcal{F}} \mathbb{P}[\mathcal{F}' = F] \mathbb{1}_{I(F,\mathbf{x})} = \mathbb{P}_{\mathcal{F}'}[I(\mathcal{F}', \mathbf{x})]$, by linearity of integration:

$$\sum_{F \subseteq \mathcal{F}} \mathbb{P}[\mathcal{F}' = F] \int \cdots \int_{[0,1]^n} \mathbb{1}_{I(F,\mathbf{x})} \, \mathbb{P}[\mathbf{x} \sim \mathcal{D}] d\mathbf{x} = \int \cdots \int_{[0,1]^n} \mathbb{P}[\mathbf{x} \sim \mathcal{D}] \mathop{\mathbb{P}}_{\mathcal{F}'}[I(\mathcal{F}', \mathbf{x})] d\mathbf{x} \geq p.$$

Thus, there exists at least one $F = \{f_1, \ldots, f_m\}$ such that

$$\mathop{\mathbb{P}}_{\mathbf{x} \sim \mathcal{D}}[I(F, \mathbf{x})] = \int \cdots \int_{[0,1]^n} \mathbb{1}_{I(F,\mathbf{x})} \cdot \mathbb{P}[\mathbf{x} \sim \mathcal{D}] d\mathbf{x} \geq p.$$

Since $p = 1 - e^{-\Omega(n^{1/3})}$ by a union bound on the indistinguishability and gap properties, the claim holds for F. Then, by a union bound over the samples, $f_i(\mathbf{x}) = f_j(\mathbf{x})$ for all $f_i, f_j \in F$ and for all samples $\mathbf{x}$ with probability $1 - e^{-\Omega(n^{1/3})}$, and we assume this is the case, as well as the gap condition, for the remaining of the proof.

It follows that the choices of the algorithm given samples from f_i, $i \in [m]$, are independent of i. Pick $i \in [m]$ uniformly at random and consider the (possibly randomized) vector $\mathbf{x}$ returned by the algorithm. Since $\mathbf{x}$ is independent of i and by the α-gap condition,

$$\mathop{\mathbb{E}}_{i \sim \mathcal{U}(\{1,\ldots,m\})} \left[f_i(\mathbf{x}) - f_i\left(\mathbf{x}_i^\star\right) \right] \geq \alpha.$$

Thus, there exists at least one $f_i \in F$ such that for f_i, the algorithm is at least an additive factor α away from $f_i(\mathbf{x}_i^\star)$. ■

2.2. The construction

We now describe a class of functions $\mathcal{F}$ and a randomized subfamily $\mathcal{F}' \subset \mathcal{F}$ that respects the conditions of Lemma 2. Each function $f^P \in \mathcal{F}$ is defined in terms of a partition P of $[n]$. We first construct a randomized collection of partitions $\mathcal{P}$ and the convex function f^P corresponding to a partition $P \in \mathcal{P}$. We then show that the randomized family of functions $\mathcal{F}' := \{f^P : P \in \mathcal{P}\}$ satisfies the indistinguishability and α-gap conditions.

The partition. When considering suitable partitions $\mathcal{P}$, there is an inherent tension between the indistinguishability and α-gap conditions. On the one hand, $\mathcal{P}$ and $\mathcal{F}'$ cannot be too large to satisfy the indistinguishability property. On the other hand, they cannot be small if we want to obtain a meaningful gap. To reconcile this tension, we consider partitions of $[n]$ into four sets A_1, A_2, B_1, and B_2 of equal size and define the random collection of partitions $\mathcal{P}$ as follows:

$$\mathcal{P} := \{(A_{1,1}, A_{2,1}, B_1, B_2), \ldots, (A_{1,n}, A_{2,n}, B_1, B_2)\}$$

where $[n]$ is first partitioned into a set A of size $n/2$ and two sets B_1, B_2 of size $n/4$ uniformly at random. Then, we generate n i.i.d. uniformly random partitions of A into two sets $A_{1,i}, A_{2,i}$ of size $n/4$. The main idea of these partitions is that we construct functions f^P such that the algorithm may learn A, B_1, and B_2, but it cannot learn A_1 and A_2 from samples $f^P(\mathbf{x})$ with high probability, as illustrated in Figure 1.

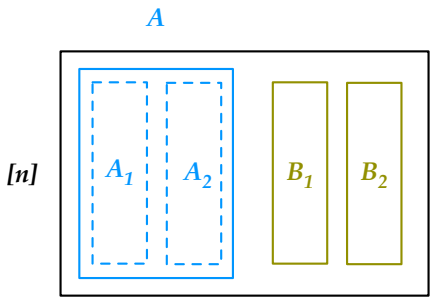

Figure 1: The partition P of the set $[n]$ into four sets A_1, A_2, B_1, B_2 of equal size. An algorithm cannot learn the parts that are dashed, A_1 and A_2, from samples.

The functions. For a partition $P = (A_1, A_2, B_1, B_2)$, the function f^P is defined as

$$f^P(\mathbf{x}) := \max\left(\mathbf{w}_A^\intercal \mathbf{x} - 2n^{-1/3}, \mathbf{w}_B^\intercal \mathbf{x}\right) + \frac{1}{2}$$

where $\mathbf{w}_A$ and $\mathbf{w}_B$ are defined such that

$$w_{A,i} := \begin{cases} \frac{2}{n} & \text{if } i \in A_1 \\ -\frac{2}{n} & \text{if } i \in A_2 \\ 0 & \text{otherwise} \end{cases} \qquad w_{B,i} := \begin{cases} \frac{2}{n} & \text{if } i \in B_1 \\ -\frac{2}{n} & \text{if } i \in B_2 \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that f^P is convex for all P since it is the maximum of two linear functions. The main idea of this construction is that with a random partition P, the values $\mathbf{w}_A^\intercal \mathbf{x}$ and $\mathbf{w}_B^\intercal \mathbf{x}$ concentrate around 0. Thus, the samples are likely to always be equal to $\mathbf{w}_B^\intercal \mathbf{x}$ due to the $-2n^{-1/3}$ term (Lemma 3). The algorithm therefore cannot learn the partition of A into A_1 and A_2. But if the algorithm does not learn A_1, A_2, then for the vector $\mathbf{x}$ returned by the algorithm, $\mathbf{w}_A^\intercal \mathbf{x}$ is likely to have value around 0 (Lemma 6), whereas $\mathbf{w}_A^\intercal \mathbf{x}^\star = -1/2$.

Indistinguishability and α-gap. Lemma 3 shows the indistinguishability condition and we obtain the gap condition by combining Lemma 5 and Lemma 6 for $\mathcal{F}' = \{f^P : P \in \mathcal{P}\}$. The main theorem then follows immediately from Lemma 2.

Lemma 3 *Fix a vector $\mathbf{x}$, then w.p. $1 - e^{-\Omega(n^{1/3})}$ over $\mathcal{P}$, for all $P \in \mathcal{P}$, $f^P(\mathbf{x}) = \mathbf{w}_B^\intercal \mathbf{x} + 1/2$.*

Proof We show that $\mathbf{w}_A^\intercal \mathbf{x}$ and $\mathbf{w}_B^\intercal \mathbf{x}$ are concentrated close to 0. Fix $\mathbf{x}$ and consider some partition $P \in \mathcal{P}$. By Hoeffding's inequality, $\mathbb{P}_P\left[\mathbf{w}_A^\intercal \mathbf{x} \geq n^{-1/3}\right] \leq 2e^{-n^{1/3}/3}$, so $\mathbf{w}_A^\intercal \mathbf{x} - 2n^{-1/3} \leq -n^{-1/3}$, with probability $1 - e^{-\Omega(n^{1/3})}$. By a union bound, this holds for all $P \in \mathcal{P}$ with probability $1 - e^{-\Omega(n^{1/3})}$. Similarly, $\mathbb{P}_P\left[\mathbf{w}_B^\intercal \mathbf{x} \leq -n^{-1/3}\right] \leq 2e^{-n^{1/3}/3}$ and $\mathbf{w}_B^\intercal \mathbf{x} \geq -n^{-1/3}$ for all $P \in \mathcal{P}$ with probability $1 - e^{-\Omega(n^{1/3})}$. Thus $\max\left(\mathbf{w}_A^\intercal \mathbf{x} - 2n^{-1/3}, \mathbf{w}_B^\intercal \mathbf{x}\right) + \frac{1}{2} = \mathbf{w}_B^\intercal \mathbf{x} + \frac{1}{2}$. ■

In order to show that the α-gap condition holds, we need an additional property C over $\mathcal{P}$ which we call *balanced*. This property requires that an index $i \in A$ is in A_1 for roughly half of the partitions P in $\mathcal{P}$. Informally, this implies that an algorithm which knows $\mathcal{P}$ but not $P \in \mathcal{P}$ does not know if an x_i, with $i \in A$, has a negative or positive contribution to the value of the function.

Definition 4 *$\mathcal{P}$ is β-*balanced *if for all $i \in A$, $(1-\beta)n/2 \leq |\{j \,:\, i \in A_{1,j}\}| \leq (1+\beta)n/2$.*

This assumption holds with high probability for $\beta = n^{-1/3}$ by a simple application of Chernoff bound and a union bound (see proof in Appendix B).

Lemma 5 *The random collection of partitions $\mathcal{P}$ is $n^{-1/3}$-balanced with probability $1 - e^{-\Omega(n^{1/3})}$.*

When $\mathcal{P}$ is $n^{-1/3}$-balanced, we obtain a $1/2 - o(1)$-gap.

Lemma 6 *Assume $\mathcal{P}$ is $n^{-1/3}$-balanced and $\mathbf{x}^{\star,P}$ is a minimizer of f^P. Then, for all $\mathbf{x}$,*

$$\mathop{\mathbb{E}}_{P\sim\mathcal{U}(\mathcal{P})}\left[f^P(\mathbf{x}) - f^P\left(\mathbf{x}^{\star,P}\right)\right] \geq \frac{1}{2}\left(1 - n^{-1/3}\right).$$

Proof At a high level, the balance condition implies that for all $i \in A$, $w_{A,i}x_i$ is close to 0 in expectation over P drawn uniformly at random from $\mathcal{P}$ while $w_{A,i}x_i^{\star,P} = -2/n$. The lemma then follows by linearity of expectation.

For $i \in A$, let $X_i = x_i$ if $i \in A_1$ and $X_i = 1 - x_i$ if $i \in A_2$. Observe that by the balance property (Lemma 5), with $P \sim \mathcal{U}(\mathcal{P})$,

$$\mathop{\mathbb{E}}_{P}[X_i] = \frac{1}{n}\left(\sum_{j\,:\,i\in A_{1,j}} x_i + \sum_{j\,:\,i\in A_{2,j}} (1-x_i)\right) \geq (1-n^{-1/3})\frac{1}{2}\left(x_i + (1-x_i)\right) = (1-n^{-1/3})\frac{1}{2}.$$

Thus, by linearity of expectation,

$$\mathop{\mathbb{E}}_{P\sim\mathcal{U}(\mathcal{P})}\left[\mathbf{w}_A^\intercal\mathbf{x}\right] = \frac{2}{n}\sum_{i\in A}\mathop{\mathbb{E}}_{P\sim\mathcal{U}(\mathcal{P})}[X_i] - \frac{1}{2} \geq \left(1-n^{-1/3}\right)\frac{1}{2} - \frac{1}{2},$$

and $\mathbb{E}_{P\sim\mathcal{U}(\mathcal{P})}\left[f^P(\mathbf{x})\right] \geq \left(1-n^{-1/3}\right)/2$ while $\mathbb{E}_{P\sim\mathcal{U}(\mathcal{P})}\left[f^P\left(\mathbf{x}^{\star,P}\right)\right] = 0$. ■

A tight lower bound. By combining Lemmas 2, 3, 5, and 6, we obtain the inapproximability result for any $\alpha \leq 1/2 - o(1)$. The above inapproximability result is tight since the center of the unit cube is a $1/2$ approximation, and can trivially be obtained without observing any samples.

Proposition 7 *Let $f : [0,1]^n \to [0,1]$ be convex. Then,*

$$f((1/2,\ldots,1/2)) - \min_{\mathbf{x}\in[0,1]^n} f(\mathbf{x}) \leq 1/2.$$

Proof Let $\mathbf{x}^\star$ be a minimizer of f. By convexity,

$$f\left((1/2,\ldots,1/2)\right) = f\left(\frac{1}{2}\mathbf{x}^\star + \frac{1}{2}(\mathbb{1}-\mathbf{x}^\star)\right) \leq \frac{1}{2}f(\mathbf{x}^\star) + \frac{1}{2}f(\mathbb{1}-\mathbf{x}^\star) \leq \frac{1}{2}f(\mathbf{x}^\star) + \frac{1}{2}.$$

Thus, $f\left((1/2,\ldots,1/2)\right) - f(\mathbf{x}^\star) \leq 1/2 - f(\mathbf{x}^\star)/2 \leq 1/2$. ■

Theorem 8 *The class of convex functions $\mathcal{F}$ is not $1/2 - o(1)$-optimizable from samples from any distribution $\mathcal{D}$ for the problem $\min_{\mathbf{x}} f(\mathbf{x})$. Furthermore, this bound is tight.*

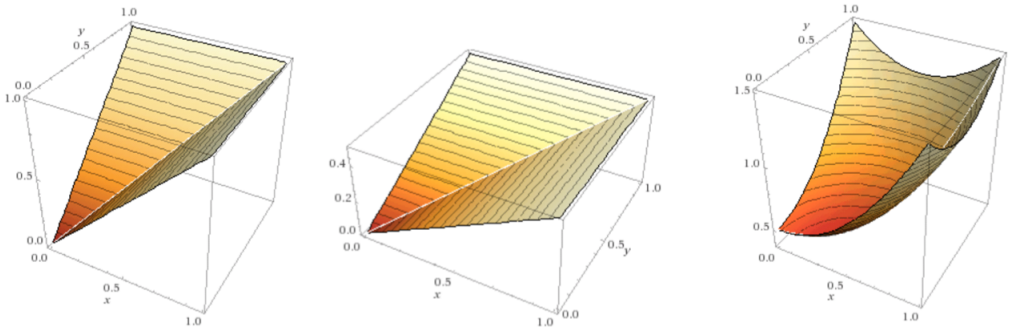

Figure 2: From left to right: $f^P((x,y))$, $f^P_\rho((x,y))$, and $f^P_\lambda((x,y))$, which are respectively convex, ρ-Lipschitz continuous convex, and λ-strongly convex, for the case $n = 2$.

2.3. Impossibility for strongly convex and Lipschitz continuous convex functions

A natural question is whether there are natural assumptions we can make on the class of convex functions that enable optimization from samples. Unfortunately, we show lower bounds even if we assume strong convexity and Lipschitz continuity. We complement these results with simple upper bounds obtained by the center of the unit cube $\mathbf{x} = (1/2, \ldots, 1/2)$. The inapproximability for such functions can be shown via simple adjustments to the functions f^P constructed in the previous section. These adjustments are illustrated in Figure 2. We use the same random collection of partition $\mathcal{P}$ for the analysis and show that the indistinguishability and α-gap conditions hold for the adjusted classes of functions for some new α. The proofs are deferred to Appendix B.2.

We adjust the family of functions $\mathcal{F}$ to obtain a family of λ strongly convex functions $\mathcal{F}_\lambda$ as follows, $f^P_\lambda(\mathbf{x}) := (1 - \lambda n/4) \cdot f^P(\mathbf{x}) + \lambda \|\mathbf{x} - (1/2, \ldots, 1/2)\|^2$.

Theorem 9 *The class of λ-strongly convex functions $\mathcal{F}_\lambda$ is not $(1-o(1))(1/2-3\lambda n/8)$-optimizable from samples from any distribution $\mathcal{D}$ for the problem $\min_{\mathbf{x}} f(\mathbf{x})$. In addition, let f be a λ-strongly convex function, then $f((1/2, \ldots, 1/2))$ is a $(1 - \lambda n/8)$-approximation to $\min_{\mathbf{x}} f(\mathbf{x})$.*

Similarly, we adjust the class of functions $\mathcal{F}$ to obtain a class of ρ-Lipschitz continuous convex functions $\mathcal{F}_\rho$ as follows, $f^P_\rho(\mathbf{x}) = \rho \cdot \sqrt{n/2} \cdot f^P(\mathbf{x})$ if $\rho \leq \sqrt{2/n}$ and $f^P_\rho(\mathbf{x}) = f^P(\mathbf{x})$ otherwise.

Theorem 10 *The class of ρ-Lipschitz convex functions $\mathcal{F}_\rho$ is not $(1 - o(1)) \min(1/2, \rho\sqrt{n}/2\sqrt{2})$-optimizable from samples from any distribution $\mathcal{D}$ for the problem $\min_{\mathbf{x}} f(\mathbf{x})$. In addition, let f be a ρ-Lipschitz function $f((1/2, \ldots, 1/2))$ is a $\min(1/2, \rho\sqrt{n}/2)$-approximation to $\min_{\mathbf{x}} f(\mathbf{x})$.*

3. Learnability of the hard functions

In this section, we show that learnability and convexity are not sufficient conditions for optimization from samples. We show this by designing a learning algorithm for the class of convex functions $\mathcal{F}$ constructed in the previous section. More precisely, we show that this class of functions is PAC learnable with the absolute loss function (or any other ρ-Lipschitz loss function, for bounded ρ).

Definition 11 (PAC learnability with absolute loss) *A class of functions $\mathcal{F}$ is PAC learnable if there exist a function $m_{\mathcal{F}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0,1)$, for every distribution $\mathcal{D}$, and every function $f \in \mathcal{F}$, when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. samples $(\mathbf{x}, f(\mathbf{x}))$ with $\mathbf{x} \sim \mathcal{D}$, the algorithm returns a function $\tilde{f}$ such that, with probability at least $1-\delta$ (over the choice of the m training examples),*

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}}\left[\left|\tilde{f}(\mathbf{x}) - f(\mathbf{x})\right|\right] \leq \epsilon.$$

Recall that the functions we wish to learn are $f(\mathbf{x}) = \max\left(\mathbf{w}_A^\intercal \mathbf{x} - 2n^{-1/3}, \mathbf{w}_B^\intercal \mathbf{x}\right) + 1/2$. Each term in the $\max$ operator is linear and thus trivially learnable (see Theorem 30). The difficulty with learning these functions is the $\max$ operator over the two linear functions $\mathbf{w}_A - 2n^{-1/3}$ and $\mathbf{w}_B$. The main idea behind our approach to learning with the $\max$ operator is to combine classification with linear regression. Intuitively, the classification step indicates which term of $\mathbf{w}_A^\intercal \mathbf{x}$ or $\mathbf{w}_B^\intercal \mathbf{x}$ is largest, and the linear regression step learns the two linear functions (Section 3.1). The challenge with using this approach is that the label of the samples is not known for the classification, i.e., given $(\mathbf{x}, f(\mathbf{x}))$, we do not know if $f(\mathbf{x}) = \mathbf{w}_A^\intercal \mathbf{x} - 2n^{-1/3} + 1/2$ or if $f(\mathbf{x}) = \mathbf{w}_B^\intercal \mathbf{x} + 1/2$. To overcome this, we develop two different approaches to deal with this labeling difficulty and discuss their tradeoffs (Section 3.2). The proofs in this section are deferred to Appendix C

3.1. Learning with labeled samples

We describe the learning algorithm assuming that we are given labeled samples and discuss how to obtain labeled samples in the next section. This learning algorithm, called CR and formally described in Algorithm 1, combines a classification step to learn the label of a fresh vector $\mathbf{x}$ with a linear regression step to learn two linear functions, one for each label. Let $\mathcal{A} = \{\mathbf{x} : f(\mathbf{x}) = \mathbf{w}_A^\intercal \mathbf{x} - 2n^{-1/3} + 1/2\}$ and $\mathcal{B} = \{\mathbf{x} : f(\mathbf{x}) = \mathbf{w}_B^\intercal \mathbf{x} + 1/2\}$. The classification we wish to learn is $C(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathcal{A}$ and $C(\mathbf{x}) = -1$ if $\mathbf{x} \in \mathcal{B}$. The linear regression tasks consist of learning $\mathbf{w}_A$ and $\mathbf{w}_B$ given m_A and m_B samples in $\mathcal{A}$ and $\mathcal{B}$.

Algorithm 1 CR, a learning algorithm for f given labeled samples which uses Classification and linear Regression.

Input: labeled samples $\mathcal{S} = \{(\mathbf{x}_i, f(\mathbf{x}_i), l_i)\}_{i=1}^m$, a fresh sample $\mathbf{x}$

if $|\{(\mathbf{x}_i, f(\mathbf{x}_i)) : i \in [m], l_i = 1\}| \leq m\delta\epsilon/8$ **then**
  $\tilde{C} \leftarrow \mathbf{x} \mapsto -1$ — Label all fresh samples $\mathbf{x}$ with $l = 1$
else if $|\{(\mathbf{x}_i, f(\mathbf{x}_i)) : i \in [m], l_i = -1\}| \leq m\delta\epsilon/8$ **then**
  $\tilde{C} \leftarrow \mathbf{x} \mapsto 1$ — Label all fresh samples $\mathbf{x}$ with $l = -1$
else
  $\tilde{C} \leftarrow \mathrm{ERM}_c(\{(\mathbf{x}_i, l_i)\}_{i=1}^m)$ — Train a classifier for labeling
end if
$\tilde{f}_A \leftarrow \mathrm{ERM}_{\|\mathbf{w}\|_1 \leq 2}(\{(\mathbf{x}_i, f(\mathbf{x}_i)) : i \in [m], l_i = 1\})$ — Linear regression on samples labeled 1
$\tilde{f}_B \leftarrow \mathrm{ERM}_{\|\mathbf{w}\|_1 \leq 2}(\{(\mathbf{x}_i, f(\mathbf{x}_i)) : i \in [m], l_i = -1\})$ — Linear regression on samples labeled -1
if $\tilde{C}(\mathbf{x}) = 1$ **then return** $\tilde{f}_A(\mathbf{x})$ **endif**
if $\tilde{C}(\mathbf{x}) = -1$ **then return** $\tilde{f}_B(\mathbf{x})$ **endif**

We restrict the linear regression to be over $\mathbf{w}$ with bounded ℓ_1 norm to obtain good generalization guarantees. If the number of samples m_A or m_B is too small to obtain guarantees on m_A

and m_B being close to their expected size in order to obtain guarantees on $\tilde{f}_A$ and $\tilde{f}_B$, then we either classify all fresh samples as -1 or 1 respectively. We denote by ERM_c and $\mathrm{ERM}_{\|\mathbf{w}\|_1 \leq 2}$ the empirical risk minimization algorithm for classification with halfspaces and linear regression over vectors with ℓ_1 norm bounded by 2.

The analysis of the sample complexity of Lemma 12 consists of three steps and is the main technical challenge in this section.

1. We begin by giving the sample complexities of the classification step (Lemma 31) and the linear regression step (Lemma 32) in isolation, which follow easily from known results using the VC-dimension and the Rademacher complexity, two of the most common tools to bound the generalization error of a learning algorithm, which we review in Appendix C.1;

2. Then, we argue that we either have enough samples m_A and m_B to obtain good guarantees on $\tilde{f}_A$ and $\tilde{f}_B$ or that we can use the classifier $\tilde{C}$ which always labels -1 or 1 if m_A or m_B, respectively, is small;

3. Finally, we derive the generalization error of our algorithm by combining four cases depending on the classification $C(\mathbf{x})$ and on the condition that the classification is correct.

Lemma 12 *For any $\epsilon, \delta \in (0,1]$, given $m = O\left(\epsilon^{-2}\delta^{-1}\left(n + \log\left(\delta^{-1}\right)\right)\right)$ labeled samples, the* CR *algorithm returns a function $\tilde{f}$ s.t. with probability $1-\delta$ over samples $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left|\tilde{f}(\mathbf{x}) - f(\mathbf{x})\right| \leq \epsilon$.*

3.2. Obtaining the labeled samples

The CR learning algorithm assumes that it is given labeled samples. Determining the label of a sample $(\mathbf{x}, f(\mathbf{x}))$ is a main conceptual component of this section. We describe two approaches with different tradeoffs.

$\mathcal{F}$ is PAC learnable. This approach, formally described in Algorithm 2, consists of searching for the "best" labeling $\mathcal{A}$ and $\mathcal{B}$ of the samples by partitioning the samples $\mathcal{S}$ into a training set $\mathcal{S}_1$ and a testing set $\mathcal{S}_2$. We train functions $\tilde{f}_{\mathcal{A},\mathcal{B}}$ over the training set for all possible labelings $\mathcal{A}, \mathcal{B}$ of the training set and evaluate which labeling $\mathcal{A}^\star, B^\star$ is best over the testing set and return $\tilde{f}_{\mathcal{A}^\star,\mathcal{B}^\star}$. This approach successfully shows that $\mathcal{F}$ is PAC learnable. In terms of computation, however, this learner is not efficient since it compares exponentially-many possible labelings.

Algorithm 2 A learning algorithm for $\mathcal{F}$.

Input: Samples $\mathcal{S} = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^m$, a fresh sample $\mathbf{x}$, a training set size m_1

$(\mathcal{S}_1, \mathcal{S}_2) \leftarrow (\{(\mathbf{x}_i, f(\mathbf{x}_i)) : i \leq m_1\}, \{(\mathbf{x}_i, f(\mathbf{x}_i)) : i > m_1\})$ training and testing set

$\mathcal{L} \leftarrow \{(\mathcal{A}, \mathcal{B}) : \mathcal{A}, \mathcal{B}$ partition $\mathcal{S}_1\}$ all labelings of training set

for $(\mathcal{A}, \mathcal{B}) \in \mathcal{L}$ **do** $\tilde{f}_{\mathcal{A},\mathcal{B}} \leftarrow \mathrm{CR}\left(\{(\mathbf{x}_i, f(\mathbf{x}_i), \mathbb{1}_{i\in\mathcal{A}}) : (\mathbf{x}_i, f(\mathbf{x}_i)) \in \mathcal{S}_1\}\right)$ **end for**

$\mathcal{A}^\star, \mathcal{B}^\star \leftarrow \mathrm{argmin}_{(\mathcal{A},\mathcal{B})\in\mathcal{L}} \sum_{(\mathbf{x}_i, f(\mathbf{x}_i))\in\mathcal{S}_2} \left|\tilde{f}_{\mathcal{A},\mathcal{B}}(\mathbf{x}_i) - f(\mathbf{x}_i)\right|$ best labeling

return $\tilde{f}_{\mathcal{A}^\star,\mathcal{B}^\star}(\mathbf{x})$

Theorem 13 *For any $\epsilon, \delta > 0$. Let $\tilde{f}$ be the function returned by Algorithm 2. Then, with probability $1-\delta$, $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left|\tilde{f}(\mathbf{x}) - f(\mathbf{x})\right| \leq \epsilon$ with sample complexity $O\left(\epsilon^{-4}\delta^{-1}\left(n + \log\left(\delta^{-1}\right)\right)\right)$.*

Efficient PAC learner for ϵ-convex functions. The second approach, described in Algorithm 3 holds for a slightly more general class of functions, which are ϵ-convex, i.e., almost convex. The idea is to simply encode the labeling of a sample $(\mathbf{x}, f(\mathbf{x}))$ into its value $f(\mathbf{x})$. This encoding can be done by setting the i-th decimal place of $f(\mathbf{x})$ to either 0 or 1. For example, assume $\max\left(\mathbf{w}_A^\intercal \mathbf{x} - 2n^{-1/3}, \mathbf{w}_B^\intercal \mathbf{x}\right) + \frac{1}{2} = 0.12345$. Then, with $i = 4$, $f(\mathbf{x}) = 0.12305$ if $\mathbf{x} \in \mathcal{A}$ and $f(\mathbf{x}) = 0.12315$ if $\mathbf{x} \in \mathcal{B}$. The decoding is then trivial. We obtain a slightly different class of functions $\mathcal{F}'$ with this encoding. This class of functions $\mathcal{F}'$ can be efficiently PAC learned with this approach, but it is only ϵ-convex, for arbitrarily small $\epsilon > 0$. The proof immediately follows by decoding the samples and then applying Lemma 12.

Algorithm 3 An efficient learning algorithm for $\mathcal{F}'$.

Input: Samples $\mathcal{S} = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^m$, a fresh sample $\mathbf{x}$
for $(\mathbf{x}_i, f(\mathbf{x}_i)) \in \mathcal{S}$ **do** $l_i \leftarrow \text{DECODE}(f(\mathbf{x}_i))$ **end for**
return $\text{CR}(\{(\mathbf{x}_i, f(\mathbf{x}_i), l_i)\}_{i=1}^m)(\mathbf{x})$

Theorem 14 *For any $\epsilon, \delta > 0$. Let $\tilde{f}$ be the function returned by Algorithm 3. Then, with probability $1-\delta$, $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left|\tilde{f}(\mathbf{x}) - f(\mathbf{x})\right| \leq \epsilon$ with sample complexity $O\left(\epsilon^{-2}\delta^{-1}\left(n + \log\left(\delta^{-1}\right)\right)\right)$.*

Note that the hardness result for optimization from samples still holds for $\mathcal{F}'$. Even if the algorithm knows the label of the samples, since all the samples are in $\mathcal{B}$ with high probability, the algorithm still cannot learn any information about the partition of A into A_1 and A_2. In addition, $\mathcal{F}'$ is also still optimizable in the value query model. Although in general ϵ-convex functions may be inapproximable (Singer and Vondrák, 2015), this class can be efficiently optimized when ϵ is sufficiently small ϵ (Belloni et al., 2015).

Learnability and optimization guarantees. To conclude, the results in this section imply that:

- There exists a class of convex functions that is PAC learnable but not optimizable from samples. The learning algorithms required to learn this class may be computationally inefficient;
- For any $\epsilon > 0$, there exists a class of ϵ-convex functions that is PAC learnable but not optimizable from samples. The learning algorithm that learns this class is computationally efficient.

4. Optimization of polynomials from samples

In Appendix D, we discuss the special case of multivariate polynomials and the notion of recoverability, i.e. approximating a function within arbitrarily good precision *everywhere* using poly-many samples. We give a simple condition on the samples for which recoverability, and thus optimization from samples, is possible for polynomials. We conclude by noting that recoverability is not a necessary condition for optimization from samples.

References

Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer, 2010.

Maria-Florina Balcan and Nicholas JA Harvey. Learning submodular functions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 793–802. ACM, 2011.

Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The power of optimization from samples. In *Advances in Neural Information Processing Systems*, pages 4017–4025, 2016.

Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The limitations of optimization from samples. *Proceedings of the Forty-Ninth Annual ACM on Symposium on Theory of Computing*, 2017.

Alexandre Belloni, Tengyuan Liang, Hariharan Narayanan, and Alexander Rakhlin. Escaping the local minima via simulated annealing: Optimization of approximately convex functions. In *COLT*, pages 240–265, 2015.

Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 301–309. ACM, 1988.

Sébastien Bubeck and Ronen Eldan. Multi-scale exploration of convex functions and bandit convex optimization. *CoRR, abs/1507.06580*, 2015.

Carl De Boor and Amos Ron. On multivariate polynomial interpolation. *Constructive Approximation*, 6(3):287–302, 1990.

Alina Ene, Jan Vondrák, and Yi Wu. Local distribution and the symmetry gap: Approximability of multiway partitioning problems. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 306–325. SIAM, 2013.

Vitaly Feldman. Generalization of erm in stochastic convex optimization: The dimension strikes back. In *Advances in Neural Information Processing Systems*, pages 3576–3584, 2016.

Vitaly Feldman and Pravesh Kothari. Learning coverage functions and private release of marginals. In *COLT*, pages 679–702, 2014.

Vitaly Feldman, Pravesh Kothari, and Jan Vondrák. Representation, approximation and learning of submodular functions using low-rank decision trees. In *COLT*, pages 711–740, 2013.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394. Society for Industrial and Applied Mathematics, 2005.

Mariano Gasca. Multivariate polynomial interpolation. In *Computation of curves and surfaces*, pages 215–236. Springer, 1990.

Avinatan Hassidim and Yaron Singer. Submodular optimization under noise. *arXiv preprint arXiv:1601.03095*, 2016.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2015.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

Vahab Mirrokni, Michael Schapira, and Jan Vondrák. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *Proceedings of the 9th ACM conference on Electronic commerce*, pages 70–77. ACM, 2008.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, 2009.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Yaron Singer and Jan Vondrák. Information-theoretic lower bounds for convex optimization with erroneous oracles. In *Advances in Neural Information Processing Systems*, pages 3204–3212, 2015.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1):265–304, 2013.

Appendix A. Concentration Bounds

Lemma 15 (Chernoff Bound) *Let $X_1, \ldots, X_n$ be independent indicator random variables such that $\mathbb{P}[X_i = 1] = 1/2$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbb{E}[X]$. For $0 < \delta < 1$,*

$$\mathbb{P}[|X - \mu| \geq \delta\mu] \leq 2e^{-\mu\delta^2/3}.$$

Lemma 16 (Hoeffding's inequality) *Let $X_1, \ldots, X_n$ be independent random variables with values in $[0, b]$. Let $X = \frac{1}{m}\sum_{i=1}^{m} X_i$ and $\mu = \mathbb{E}[X]$. Then for every $0 < \epsilon < 1$,*

$$\mathbb{P}\left[|X - \mu| \geq \epsilon\right] \leq 2e^{-2m\epsilon^2/b^2}.$$

Appendix B. Missing analysis from Section 2

B.1. Missing concentration bound in analysis from Section 2.1

Lemma 5 *The random collection of partitions $\mathcal{P}$ is $n^{-1/3}$-balanced with probability $1 - e^{-\Omega(n^{1/3})}$.*

Proof By Chernoff bound, with $\mu = n/2$ and $\delta = n^{-1/3}$,

$$\mathop{\mathbb{P}}_{\mathcal{P}}\left[\left||\{j \,:\, i \in A_{1,j}\}| - \frac{n}{2}\right| \geq n^{-1/3} \cdot \frac{n}{2}\right] \leq 2e^{-n^{1/3}/6}$$

By a union bound, the claim holds for all $i \in A$ with probability $1 - e^{-\Omega(n^{1/3})}$. ■

B.2. Missing analysis from Section 2.3

A natural question that arises from the hardness result of optimizing general convex functions from samples is if there exists well-behaved families of convex functions which are optimizable from samples. We provide additional hardness results for strongly convex functions and Lipschitz continuous convex functions. We complement these results with simple upper bounds obtained by the center of the unit cube $\mathbf{x} = (1/2, \ldots, 1/2)$.

Definition 17 *A function f is*

- ***λ-strongly convex**, if for all $\mathbf{x}, \mathbf{y}$ and $\alpha \in [0, 1]$,*

$$f(\alpha\mathbf{x} + (\mathbb{1} - \alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}) - \frac{\lambda}{2}\alpha(1 - \alpha)\|\mathbf{x} - \mathbf{y}\|^2;$$

- ***ρ-Lipschitz continuous**, if for all $\mathbf{x}, \mathbf{y}$, $|f(\mathbf{x}) - f(\mathbf{y})| \leq \rho\|\mathbf{x} - \mathbf{y}\|$.*

We obtain inapproximability for such functions with simple adjustments to the functions constructed for the main impossibility result. These adjustments are illustrated in Figure 2. We use the same partition $\mathcal{P}$ and show that the indistinguishable and gap α properties hold for the adjusted classes of functions for some new α.

B.2.1. STRONG CONVEXITY

In our setting, we have $\lambda \leq 8/n$ for λ-strongly convex functions (Lemma 20). We obtain a $\max((1-o(1))(1/2-3\lambda n/8), 0)$ lower bound (Theorem 9) and an $1-\lambda n/8$ upper bound (Lemma 20) We begin with useful simple known results for strongly convex functions.

Lemma 18 (Shalev-Shwartz and Ben-David (2014)) *The function $f(\mathbf{x}) = \lambda\|\mathbf{x}\|^2$ is 2λ-strongly convex. In addition, if f is λ-strongly convex and g is convex, then $f+g$ is λ-strongly convex.*

We adjust the family of functions $\mathcal{F}$ to a family of λ strongly convex functions $\mathcal{F}_\lambda$,

$$f_\lambda^P(\mathbf{x}) := \left(1 - \frac{\lambda n}{4}\right) \cdot f^P(\mathbf{x}) + \lambda\|\mathbf{x} - \mathbf{1/2}\|^2.$$

It is immediate from Lemma 18 that f_λ^P is λ strongly convex. We show that the indistinguishable and gap $\alpha = \max((1-o(1))(1/2 - 3\lambda n/8), 0)$ properties are satisfied.

Lemma 19 *The class of λ-strongly convex functions $\mathcal{F}_\lambda$ is not $\max((1-o(1))(1/2-3\lambda n/8), 0)$-optimizable from samples from any distribution $\mathcal{D}$ for the problem $\min_{\mathbf{x}} f(\mathbf{x})$.*

Proof The only dependence of $f_\lambda^P(\mathbf{x})$ on P is the $f^P(\mathbf{x})$ term. Thus, since $\mathcal{F}$ satisfies the indistinguishability condition (Lemma 3), so does $\mathcal{F}'$. For the gap condition, define $\mathbf{x}^P$ such that $x_i^P = 0$ if $i \in A_1 \cup B_1$ and $x_i^P = 1$ if $i \in A_2 \cup B_2$, so $f_\lambda^P\left(\mathbf{x}^P\right) = (1-\lambda n/4)\cdot 0 + \lambda n/4 = \lambda n/4$. Thus, by the gap condition for f^P (Lemma 6), for all $\mathbf{x}$, we get

$$\begin{aligned}
&\mathop{\mathbb{E}}_{P\sim\mathcal{U}(\mathcal{P})}\left[f_\lambda^P(\mathbf{x}) - f_\lambda^P\left(\mathbf{x}^P\right)\right] \\
\geq &\mathop{\mathbb{E}}_{P\sim\mathcal{U}(\mathcal{P})}\left[\left(1-\frac{\lambda n}{4}\right) f^P(\mathbf{x}) + \lambda\|\mathbf{x}-\mathbf{1/2}\|^2\right] - \frac{\lambda n}{4} \\
\geq &\left(1-\frac{\lambda n}{4}\right)\left(\left(1-n^{-1/3}\right)\frac{1}{2} - 2n^{-1/3}\right) - \frac{\lambda n}{4} \\
= &(1-o(1))\left(\frac{1}{2} - \frac{3\lambda n}{8}\right).
\end{aligned}$$

Combining the indistinguishable and $\max((1-o(1))(1/2-3\lambda n/8), 0)$-gap conditions with Lemma 2 completes the proof. ■

We complement the hardness result with a simple upper bound.

Proposition 20 *Let f be a λ-strongly convex function. Then, $f((1/2, \ldots, 1/2))$ is a $1-\lambda n/8$-approximation to $\min_{\mathbf{x}} f(\mathbf{x})$. In addition, it must be the case that $\lambda \leq 8/n$.*

Proof By the definition of λ-strongly convex, with $\mathbf{x} = (0, \ldots, 0)$, $\mathbf{y} = (1, \ldots, 1)$, and $\alpha = 1/2$,

$$f((1/2, \ldots, 1/2)) \leq \frac{1}{2}f(\mathbf{x}) + \frac{1}{2}f(\mathbf{y}) - \frac{\lambda}{2}\frac{1}{4}\|\mathbf{x}-\mathbf{y}\|^2 \leq 1 - \frac{\lambda n}{8}.$$

In addition, since $f((1/2, \ldots, 1/2)) \geq 0$, it must be the case that $\lambda \leq 8/n$ in this setting. ■

Theorem 9 *The class of λ-strongly convex functions $\mathcal{F}_\lambda$ is not $(1-o(1))(1/2-3\lambda n/8)$-optimizable from samples from any distribution $\mathcal{D}$ for the problem $\min_{\mathbf{x}} f(\mathbf{x})$. In addition, let f be a λ-strongly convex function, then $f((1/2, \ldots, 1/2))$ is a $(1-\lambda n/8)$-approximation to $\min_{\mathbf{x}} f(\mathbf{x})$.*

B.2.2. LIPSCHITZ CONTINUOUS

We turn our attention to ρ-Lipschitz continuous convex functions. The results for these functions are similar to the results for strongly convex functions, we obtain an $(1 - o(1)) \min(1/2, \rho\sqrt{n}/2\sqrt{2})$ lower bound and a $\min(1/2, \rho\sqrt{n}/2)$ upper bound. Again, we begin with simple results for Lipschitz continuous convex functions.

Lemma 21 *A linear function $f(\mathbf{x}) = \mathbf{w}^\intercal \mathbf{x} + b$ is $\|\mathbf{w}\|$-Lipschitz. In addition, let f_1 and f_2 be functions that are ρ-Lipschitz. Then, the functions $\max(f_1(\mathbf{x}), f_2(\mathbf{x}))$ and $\gamma f_1(\mathbf{x})$ are ρ-Lipschitz and $\gamma\rho$-Lipschitz respectively.*

Proof The result for linear functions is from (Shalev-Shwartz and Ben-David, 2014). By the definition of ρ-Lipschitz, observe that $|\max(f_1(\mathbf{x}), f_2(\mathbf{x})) - \max(f_1(\mathbf{y}), f_2(\mathbf{y}))| \leq \max_i |f_i(\mathbf{x}) - f_i(\mathbf{x})| \leq \rho\|\mathbf{x} - \mathbf{y}\|$ and that $|\gamma f_1(\mathbf{x}) - \gamma f_1(\mathbf{x})| \leq \gamma|f_1(\mathbf{x}) - f_1(\mathbf{y})| \leq \gamma\rho\|\mathbf{x} - \mathbf{y}\|$. ■

We adjust $\mathcal{F}$ to ρ Lipschitz continuous convex functions $\mathcal{F}_\rho$ as follows.

$$f_\rho^P(\mathbf{x}) = \begin{cases} \rho \cdot \sqrt{\frac{n}{2}} \cdot f^P(\mathbf{x}) & \text{if } \rho \leq \sqrt{\frac{2}{n}} \\ f^P(\mathbf{x}) & \text{otherwise} \end{cases}$$

It is immediate from Lemma 21 that f^P is $\sqrt{2/n}$-Lipschitz continuous convex and that f_ρ^P is ρ-Lipschitz continuous convex. The main result for Lipschitz functions follows from showing the indistinguisability and gap α conditions.

Lemma 22 *The class of ρ-Lipschitz convex functions $\mathcal{F}_\rho$ is not $(1 - o(1)) \min(1/2, \rho\sqrt{n}/2\sqrt{2})$-optimizable from samples from any distribution $\mathcal{D}$ for the problem $\min_{\mathbf{x}} f(\mathbf{x})$.*

Proof The only dependence of f_ρ^P on P is the f^P term. Thus, since $\mathcal{F}$ satisfies the indistinguishability condition (Lemma 3), so does $\mathcal{F}'$. If $\rho > \sqrt{2/n}$, the ρ-gap property follows immediately by the definition and the gap condition for f^P (Lemma 6). Otherwise, by Lemma 6, for all $\mathbf{x}$, we get

$$\begin{aligned} &\mathop{\mathbb{E}}_{P\sim\mathcal{U}(\mathcal{P})} \left[f_\rho^P(\mathbf{x}) - f_\rho^P(\mathbf{x}^{\star,P})\right] \\ \geq &\mathop{\mathbb{E}}_{P\sim\mathcal{U}(\mathcal{P})} \left[\rho \cdot \sqrt{\frac{n}{2}} \cdot f^P(\mathbf{x})\right] \\ \geq &\rho \cdot \sqrt{\frac{n}{2}} \cdot \left(\left(1 - n^{-1/3}\right)\frac{1}{2} - 2n^{-1/3}\right) \\ = &(1 - o(1))\frac{\rho\sqrt{n}}{2\sqrt{2}} \end{aligned}$$

Combining the indistinguishable and $(1-o(1)) \min(1/2, \rho\sqrt{n}/2\sqrt{2})$-gap conditions with Lemma 2 completes the proof. ■

We complement the hardness result for Lipschitz continuous functions with an upper bound.

Proposition 23 *Let f be a ρ-Lipschitz continuous function. Then, the center of the unit cube, $(1/2, \ldots, 1/2)$, is a $\min(1/2, \rho\sqrt{n}/2)$-approximation to $\min_{\mathbf{x}} f(\mathbf{x})$.*

Proof Let $\mathbf{x}^\star \in [0,1]^n$ be the optimal solution. Then, by definition of ρ-Lipschitz continuous,

$$f\left((1/2,\ldots,1/2)\right) - f(\mathbf{x}^*) \leq \rho \left\|(1/2,\ldots,1/2) - \mathbf{x}^\star\right\| \leq \rho \left\|(1/2,\ldots,1/2) - \mathbb{1}\right\| \leq \rho\frac{\sqrt{n}}{2}.$$

■

Theorem 10 *The class of ρ-Lipschitz convex functions $\mathcal{F}_\rho$ is not $(1-o(1))\min(1/2, \rho\sqrt{n}/2\sqrt{2})$-optimizable from samples from any distribution $\mathcal{D}$ for the problem $\min_{\mathbf{x}} f(\mathbf{x})$. In addition, let f be a ρ-Lipschitz function $f((1/2,\ldots,1/2))$ is a $\min(1/2, \rho\sqrt{n}/2)$-approximation to $\min_{\mathbf{x}} f(\mathbf{x})$.*

Appendix C. Missing analysis from Section 3

We begin by reviewing known results for classification and linear regression using the VC-dimension and the Rademacher complexity (Section C.1). We then construct a learning algorithm called CR which combines classification and regression and which assumes that it is given labeled samples (Section C.2). Finally, we describe the two possible approaches to labeling samples (Section C.3)

C.1. VC-dimension and Rademacher complexities essentials

We review learning results needed for the analysis. These results use the VC-dimension and the Radmeacher complexity, two of the most common tools to bound the generalization error of a learning algorithm. We formally define the VC-dimension and the Rademacher complexity using definitions from (Shalev-Shwartz and Ben-David, 2014). We begin with the VC-dimension, which is for classes of binary functions. We first define the concepts of restriction to a set and of shattering, which are useful to define the VC-dimension.

Definition 24 *(Restriction of $\mathcal{H}$ to A). Let $\mathcal{H}$ be a class of functions from $\mathcal{X}$ to $\{0,1\}$ and let $A = \{a_1,\ldots,a_m\} \subset \mathcal{X}$. The restriction of $\mathcal{H}$ to A is the set of functions from A to $\{0,1\}$ that can be derived from $\mathcal{H}$. That is,*

$$\mathcal{H}_A = \{(h(a_1),\ldots,h(a_m)) \,:\, h \in \mathcal{H}\},$$

where we represent each function from A to $\{0,1\}$ as a vector in $\{0,1\}^{|A|}$.

Definition 25 *(Shattering). A hypothesis class $\mathcal{H}$ shatters a finite set $A \subset \mathcal{X}$ if the restriction of $\mathcal{H}$ to A is the set of all functions from A to $\{0,1\}$. That is, $|\mathcal{H}_A| = 2^{|A|}$.*

Definition 26 *(VC-dimension). The VC-dimension of a hypothesis class $\mathcal{H}$ is the maximal size of a set $S \subset \mathcal{X}$ that can be shattered by $\mathcal{H}$. If $\mathcal{H}$ can shatter sets of arbitrarily large size we say that $\mathcal{H}$ has infinite VC-dimension.*

Next, we define the Rademacher complexity, which is for more complex classes of functions than binary functions, such as real-valued functions.

Definition 27 *(Rademacher complexity). Let σ be distributed i.i.d. according to $\Pr[\sigma_i = 1] = \Pr[\sigma_i = -1] = 1/2$. The Rademacher complexity $R(A)$ of set of vectors $A \subset \mathbb{R}^m$ is defined as*

$$R(A) := \frac{1}{m}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{a\in A}\sum_{i=1}^{m}\sigma_i a_i\right].$$

This first result bounds the generalization error of a class of binary functions in terms of the VC-dimension of these classifiers.

Theorem 28 (Shalev-Shwartz and Ben-David (2014)) *Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{-1, 1\}$ and $f : \mathcal{X} \mapsto \{-1, 1\}$ be some "correct" function. Assume that the VC-dimension of $\mathcal{H}$ is d. Then, there is an absolute constant C such that with $m \geq C(d + \log(1/\Delta))/\delta^2$ i.i.d. samples $\mathbf{x}^1, \ldots, \mathbf{x}^m \sim \mathcal{D}$,*

$$\left|\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}}[h(\mathbf{x}) \neq f(\mathbf{x})] - \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}_{h(\mathbf{x}_i)\neq f(\mathbf{x}_i)}\right| \leq \delta$$

for all $h \in \mathcal{H}$, with probability $1 - \Delta$ over the samples.

We use the class of halfspaces for classification, for which we know the VC-dimension.

Theorem 29 (Shalev-Shwartz and Ben-David (2014)) *The class of functions $\{\mathbf{x} \mapsto \mathrm{sign}(\mathbf{w}^\intercal\mathbf{x}) + b : \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}\}$ has VC dimension $n + 1$.*

The following result combines the generalization error of a class of functions in terms of its Rademacher complexity with the Rademacher complexity of linear functions over a ρ-Lipschitz loss function.

Theorem 30 (Shalev-Shwartz and Ben-David (2014)) *Suppose that $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$ such that with probability 1 we have that $\|\mathbf{x}\|_\infty \leq R$. Let $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|w\|_1 \leq B\}$ and let $\ell(\mathbf{w}, (\mathbf{x}, y)) = \phi(\mathbf{w}^\intercal\mathbf{x}, y)$ such that for all $y \in \mathcal{Y}$, $a \mapsto \phi(a, y)$ is an ρ-Lipschitz function and such that $\max_{a\in[-BR,BR]} |\phi(a, y)| \leq c$. Then, for any $\delta \in (0, 1)$, with probability of at least $1 - \delta$ over the choice of an i.i.d. sample of size m,*

$$L_{\mathcal{D}}(\mathbf{w}) \leq L_S(\mathbf{w}) + 2\rho BR\sqrt{\frac{2\log(2d)}{m}} + c\sqrt{\frac{2\log(2/\delta)}{m}}$$

for all $\mathbf{w} \in \mathcal{H}$.

C.2. Missing analysis from Section 3.1

We describe the learning algorithm assuming that we are given labeled samples and discuss how to obtain labeled samples in the next section. This learning algorithm combines a classification step to learn the label of a fresh vector $\mathbf{x}$ with a regression step to learn two linear functions, one for each label. The analyses of the classification step (Lemma 31) and the regression step (Lemma 32) in isolation are immediate from the above known results. The interesting aspect of this learning algorithm is how to combine them (Lemma 12). Recall that the class of functions we wish to learn,

$$f(\mathbf{x}) = \max\left(\mathbf{w}_A^\intercal \mathbf{x} - 2n^{-1/3}, \mathbf{w}_B^\intercal \mathbf{x}\right) + \frac{1}{2}.$$

Let $\mathcal{A} = \{\mathbf{x} : f(\mathbf{x}) = \mathbf{w}_A^\intercal \mathbf{x} - 2n^{-1/3} + 1/2\}$ and $\mathcal{B} = \{\mathbf{x} : f(\mathbf{x}) = \mathbf{w}_B^\intercal \mathbf{x} + 1/2\}$. The classification we wish to learn is $C(\mathbf{x}) = 1$ if $\mathbf{x} \in \mathcal{A}$ and $C(\mathbf{x}) = -1$ if $\mathbf{x} \in \mathcal{B}$.

Lemma 31 *Given a collection of labeled samples $\mathcal{S} = \{(\mathbf{x}_i, f(\mathbf{x}_i), l_i)\}_{i=1}^m$, consider the classifier $\tilde{C}$ such that*

$$(\tilde{\mathbf{w}}, \tilde{b}) = \min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} \sum_{i=1}^m \mathbb{1}_{\text{sign}(\mathbf{w}^\intercal \mathbf{x}_i + b) \neq l_i}$$

and $\tilde{C}(\mathbf{x}) = \text{sign}(\tilde{\mathbf{w}}^\intercal \mathbf{x} + \tilde{b})$. Then, there is an absolute constant c such that with $m \geq c(n + 1 + \log(1/\delta))/\Delta^2$ i.i.d. samples $\mathbf{x}^1, \ldots, \mathbf{x}^m \sim \mathcal{D}$, $\mathbb{P}_{\mathbf{x}\sim\mathcal{D}}\left[\tilde{C}(\mathbf{x}) \neq C(\mathbf{x})\right] \leq \Delta$ with probability $1 - \delta$ over the samples.

Proof Observe that with $\mathbf{w} = \mathbf{w}_A - \mathbf{w}_B$ and $b = -2n^{-1/3}$, we obtain the true classifier C with 0 loss. Thus, if $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ are minimizers, then $\sum_{i=1}^m \mathbb{1}_{\text{sign}(\mathbf{w}^\intercal \mathbf{x}_i + b) \neq C(\mathbf{x}_i)} = 0$. We then combine Theorems 28 and 29 to conclude the proof. ■

Next, we consider the linear regression tasks of learning $\mathbf{w}_A$ and $\mathbf{w}_B$ given m_A and m_B samples in $\mathcal{A}$ and $\mathcal{B}$. Define the distribution $\mathcal{D}_\mathcal{A}$ such that $\mathbb{P}[\mathbf{x} \sim \mathcal{D}_\mathcal{A}] = \mathbb{P}[\mathbf{x} \sim \mathcal{D} : \mathbf{x} \in \mathcal{A}]$ and define $\mathcal{D}_\mathcal{B}$ similarly. Define $\mathbf{x}' \in \mathbb{R}^{n+1}$ to be $x'_i = x_i$ for $i \leq n$ and $x'_{n+1} = 1$.

Lemma 32 *Let $\tilde{\mathbf{w}}_A \in \mathbb{R}^{n+1}$ be such that $\|\tilde{\mathbf{w}}_A\|_1 \leq 2$ and such that the hypothesis function $\tilde{\mathbf{w}}_A^\intercal \mathbf{x}'$ is a linear function that minimizes the empirical risk according to the absolute loss, i.e.,*

$$\tilde{\mathbf{w}}_A = \min_{\mathbf{w}\in\mathbb{R}^{n+1}} \sum_{\mathbf{x}\in\mathcal{S}_A} |\mathbf{w}^\intercal \mathbf{x}' - f(\mathbf{x})|,$$

where $\mathcal{S}_A$ is a collection of m_A i.i.d. samples drawn from $\mathcal{D}_\mathcal{A}$ then

$$\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_\mathcal{A}}\left[|\tilde{\mathbf{w}}_A^\intercal \mathbf{x}' - f(\mathbf{x})|\right] \leq 4\sqrt{\frac{2\log(2(n+1))}{m_A}} + 2\sqrt{\frac{2\log(2/\delta)}{m_A}}$$

Proof Observe that $\mathbf{w}'$ such that $w'_i = w_{A,i}$ for $i \leq n$ and $w'_{n+1} = 1/2 - 2n^{-1/3}$ has 0 empirical loss since it corresponds to the true underlying function. We then conclude the proof by applying Theorem 30 and noting that in this setting, $R = 1$, $B = 2$, $\rho = 1$ (absolute loss is 1-Lipschitz), $c = 2$, and $d = n + 1$ are sufficient. ■

We obtain identical guarantees for $\tilde{\mathbf{w}}_B \in \mathbb{R}^{n+1}$.

Lemma 12 *For any $\epsilon, \delta \in (0, 1]$, given $m = O\left(\epsilon^{-2}\delta^{-1}\left(n + \log\left(\delta^{-1}\right)\right)\right)$ labeled samples, the* CR *algorithm returns a function $\tilde{f}$ s.t. with probability $1-\delta$ over samples $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left|\tilde{f}(\mathbf{x}) - f(\mathbf{x})\right| \leq \epsilon$.*

Proof The proof consists of two steps. We begin by arguing that either the number of samples labeled $\mathcal{A}$ and $\mathcal{B}$ are concentrated close to their expected sizes or we can immediately bound ϵ or δ. We then derive the generalization error of our algorithm by combining four cases depending on the classification $C(\mathbf{x})$ of $\mathbf{x}$ into $\mathcal{A}$ or $\mathcal{B}$ and on if this classification is correct.

Divide the probability of failure δ over the samples into three components: the probability δ_S that the number of samples m_A and m_B are not satisfying, the probability δ_C that the guarantees for the classification algorithm do not hold (not to be mixed up with Δ from Lemma 31), and the probability δ_L that the guarantees for the regression algorithms do not hold. We bound each of these by $\delta/3$ and obtain a probability of failure δ by a union bound. Let p be the probability that $\mathbf{x} \in \mathcal{A}$ assuming $\mathbf{x} \sim \mathcal{D}$ and Δ be the probability of misclassification. We wish to obtain a number of samples m_A and m_B close to their mean, if this is not the case we bound the loss.

If $m_A < mp/2$ or $m_B < mp/2$. If the number of samples labeled $\mathcal{A}$ is small, $m_A \leq m\delta\epsilon/8$, then we classify all samples by B, i.e. $\tilde{C}(\mathbf{x}) = -1$ for all $\mathbf{x}$. There are then two cases, either p is small, $p \leq \epsilon\delta/4$ and we incur a small loss on ϵ or p is large and we consider this as failure and incur a small loss on δ. In the first case, $p \leq \epsilon\delta/4$, then get probability of misclassification $\Delta \leq \epsilon\delta/4$ of $\tilde{C}$ which is always -1. We will use this bound on Δ to bound ϵ in the next step. In the other case, $p \geq \epsilon\delta/4$, then we consider that failure of the choice of the samples and incur a loss on δ which we bound. By Chernoff bound, with $\mu = mp$ and $X_i = \mathbb{1}_{i \in \mathcal{A}}$ indicating if the ith sample is labeled $\mathcal{A}$,

$$\mathbb{P}\left[X \leq m\delta\epsilon/8\right] \leq \mathbb{P}\left[|\sum_i X_i - mp| \geq mp/2\right] \leq 2e^{-\frac{mp}{12}} \leq 2e^{-\frac{m\delta\epsilon}{48}}.$$

So with $m \geq \frac{48}{\delta\epsilon}\log\frac{6}{\delta}$, we can bound the probability of failure δ_S over the m samples with $\delta_S = \mathbb{P}\left[X \leq m\delta\epsilon/8\right] \leq \delta/3$. We treat the case $m_B \leq m\delta\epsilon/8$ similarly.

If $m_A \geq m\delta\epsilon/8$, we claim that $m_A \geq mp/2$ with small probability of failure. If $p \leq \delta\epsilon/4$, then $m_A \geq mp/2$. Otherwise, if $p \geq \delta\epsilon/4$, by Chernoff bound,

$$\mathbb{P}\left[X \geq mp/2\right] \leq \mathbb{P}\left[|\sum_i X_i - mp| \geq mp/2\right] \leq 2e^{-\frac{mp}{12}} \leq 2e^{-\frac{m\epsilon\delta}{48}}$$

So with $m \geq \frac{48}{\epsilon\delta}\log\frac{6}{\delta}$, we can bound the probability of failure δ_S over the m samples with $\delta_S = \mathbb{P}\left[X < mp/2\right] \leq \delta/3$. We treat the case $m_B \geq m\delta\epsilon/8$ similarly.

We conclude that if $m_A < mp/2$ or $m_B < mp/2$, then we have bounded the loss incurred by these cases either with ϵ or δ.

Combining classification and linear regression. Let Δ_A be the probability of misclassification for $\mathbf{x} \sim \mathcal{D} : \mathbf{x} \in \mathcal{A}$, i.e.

$$\Delta_A = \mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}:\mathbf{x}\in\mathcal{A}}\left[\tilde{C}(\mathbf{x}) = -1\right]$$

and define Δ_B similarly. Thus, $\Delta_A p + \Delta_B(1-p) = \Delta$. We decompose the performance of the learning algorithm into four cases depending on the classification $\tilde{C}(\mathbf{x})$ of $\mathbf{x}$ into $\mathcal{A}$ or $\mathcal{B}$ and on if this classification is correct.

- In the case where the classification is incorrect, we get

$$\begin{aligned}
&\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\in\mathcal{A}]\cdot\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{A}}}\left[\tilde{C}(\mathbf{x})=-1\right]\cdot\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{A}}:\tilde{C}(\mathbf{x})=-1}\left[|\tilde{f}(\mathbf{x})-f(\mathbf{x})|\right]\\
=&\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\in\mathcal{A}]\cdot\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{A}}}\left[\tilde{C}(\mathbf{x})=-1\right]\cdot\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{A}}:\tilde{C}(\mathbf{x})=-1}\left[|\tilde{\mathbf{w}}_B^{\intercal}\mathbf{x}'-f(\mathbf{x})|\right]\\
\leq&p\cdot\Delta_A\cdot 2
\end{aligned}$$

  since $\|\tilde{\mathbf{w}}_B\|_1\leq 2$. Similarly,

$$\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\in\mathcal{B}]\cdot\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{B}}}\left[\tilde{C}(\mathbf{x})=1\right]\cdot\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{B}}:\tilde{C}(\mathbf{x})=1}\left[|\tilde{f}(\mathbf{x})-f(\mathbf{x})|\right]\leq(1-p)\cdot\Delta_B\cdot 2.$$

- In the case where the classification is correct, we get

$$\begin{aligned}
&\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\in\mathcal{A}]\cdot\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{A}}}\left[\tilde{C}(\mathbf{x})=1\right]\cdot\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{A}}:\tilde{C}(\mathbf{x})=1}\left[|\tilde{f}(\mathbf{x})-f(\mathbf{x})|\right]\\
\leq&\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\in\mathcal{A}]\cdot\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{A}}}\left[|\tilde{\mathbf{w}}_A^{\intercal}\mathbf{x}'-f(\mathbf{x})|\right]\\
\leq&p\cdot\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{A}}}\left[|\tilde{\mathbf{w}}_A^{\intercal}\mathbf{x}'-f(\mathbf{x})|\right]
\end{aligned}$$

  and similarly,

$$\begin{aligned}
&\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x}\in\mathcal{B}]\cdot\mathop{\mathbb{P}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{B}}}\left[\tilde{C}(\mathbf{x})=-1\right]\cdot\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{B}}:\tilde{C}(\mathbf{x})=-1}\left[|\tilde{f}(\mathbf{x})-f(\mathbf{x})|\right]\\
\leq&(1-p)\cdot\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{B}}}\left[|\tilde{\mathbf{w}}_B^{\intercal}\mathbf{x}'-f(\mathbf{x})|\right]
\end{aligned}$$

Combining these four cases together, we obtain

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}}\left[\left|\tilde{f}(\mathbf{x})-f(\mathbf{x})\right|\right] &\le 2\Delta_A p + p\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_\mathcal{A}}\left[|\tilde{\mathbf{w}}_A^\intercal\mathbf{x}'-f(\mathbf{x})|\right] \\
&\quad + 2\Delta_B(1-p) + (1-p)\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}_\mathcal{B}}\left[|\tilde{\mathbf{w}}_B^\intercal\mathbf{x}'-f(\mathbf{x})|\right] \\
&\le 2\Delta + p\left(4\sqrt{\frac{2\log(2(n+1))}{m_A}}+2\sqrt{\frac{2\log(2/\delta_L)}{m_A}}\right) \\
&\quad + (1-p)\left(4\sqrt{\frac{2\log(2(n+1))}{m_B}}+2\sqrt{\frac{2\log(2/\delta_L)}{m_B}}\right) \\
&\le 2\Delta + p\left(4\sqrt{\frac{4\log(2(n+1))}{pm}}+2\sqrt{\frac{4\log(2/\delta_L)}{pm}}\right) \\
&\quad + (1-p)\left(4\sqrt{\frac{4\log(2(n+1))}{(1-p)m}}+2\sqrt{\frac{4\log(2/\delta_L)}{(1-p)m}}\right) \\
&\le 2\Delta + 8\sqrt{\frac{\log(2(n+1))}{m}}+4\sqrt{\frac{\log(2/\delta_L)}{m}} \\
&\quad + 8\sqrt{\frac{\log(2(n+1))}{m}}+4\sqrt{\frac{2\log(2/\delta_L)}{m}} \\
&\le 2\Delta + 16\sqrt{\frac{\log(2(n+1))}{m}}+8\sqrt{\frac{\log(2/\delta_L)}{m}}
\end{aligned}$$

with probability $1-\delta_L = 1-\delta/3$. The first inequality decomposes the error into the four cases discussed previously. The second inequality is since $\Delta_A p + \Delta_B(1-p) = \Delta$ and by Lemma 32. The third inequality is since by the previous argument that either $m_A \ge mp/2$, or $\tilde{C}$ never classifies a sample as $\mathcal{A}$, or we consider it as failure δ. The last two inequalities are simple algebraic manipulations. By Lemma 31, the probability Δ of misclassification can be such that $\Delta \le \epsilon/4$ with sample complexity $\frac{1}{\epsilon^2}\left(c(n+1+\log(2/\delta_C)\right)$. We get $16\sqrt{\frac{\log(2(n+1))}{m}}+8\sqrt{\frac{\log(2/\delta_L)}{m}} \le \epsilon/2$ with sample complexity $m \ge \frac{1}{\epsilon^2}\left(128\log(2(n+1))+32\log(6/\delta)\right)$. Thus, we obtain $\mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}}\left[\left|\tilde{f}(\mathbf{x})-f(\mathbf{x})\right|\right] \le \epsilon$ with probability $1-\delta$. ■

C.3. Missing analysis from Section 3.2

Define m_{te} and m_{tr} to be the sizes of the testing and training sets respectively. The empirical testing error of $\tilde{f}_{\mathcal{A},\mathcal{B}}$ is

$$L_{\mathcal{S}_{te}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}\right) = \frac{1}{m_{te}}\sum_{i=1}^{m_{te}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}(\mathbf{x})-f(\mathbf{x})\right)$$

and its expected error is

$$L_{\mathcal{D}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}\right) = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{D}}\left[\tilde{f}_{\mathcal{A},\mathcal{B}}(\mathbf{x})-f(\mathbf{x})\right]$$

We set the size of the training set to be $m_{tr} = c\epsilon^{-2}\delta^{-1}(n+\log(\delta^{-1}))$.

Lemma 33 *With probability* $1-\delta$*, for all* $(\mathcal{A},\mathcal{B})$ *partitioning* $\mathcal{S}_{tr}$*,*

$$\left|L_{\mathcal{D}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}\right)-L_{\mathcal{S}_{te}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}\right)\right|\leq\epsilon$$

with sample complexity

$$m_{te}\geq 8(m_{tr}+\log(1/\delta_{te})/\epsilon^2=O\left(\left(m_{tr}+\log\left(\frac{1}{\delta}\right)\right)\frac{1}{\epsilon^2}\right).$$

Proof Fix $\mathcal{A},\mathcal{B}$. By Hoeffding's inequality where $X_i=\tilde{f}_{\mathcal{A},\mathcal{B}}(\mathbf{x}_i)-f(\mathbf{x}_i)$, $m=m_{te}$, $\mu=L_{\mathcal{D}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}\right)$, and $b=2$.

$$\mathbb{P}\left[\left|L_{\mathcal{D}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}\right)-L_{\mathcal{S}_{te}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}\right)\right|>\epsilon\right]\leq 2e^{-m_{te}\epsilon^2/8}$$

By a union bound over all $2^{|m_{tr}|}$ pairs $(\mathcal{A},\mathcal{B})$, the above holds with probability at most $e^{m_{tr}-m_{te}\epsilon^2/8}$. Thus, we get the desired bound with $m_{te}\geq 8(m_{tr}+\log(1/\delta_{te})/\epsilon^2$. ■

Theorem 13 *For any* $\epsilon,\delta>0$*. Let* $\tilde{f}$ *be the function returned by Algorithm 2. Then, with probability* $1-\delta$*,* $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left|\tilde{f}(\mathbf{x})-f(\mathbf{x})\right|\leq\epsilon$ *with sample complexity* $O\left(\epsilon^{-4}\delta^{-1}\left(n+\log\left(\delta^{-1}\right)\right)\right)$*.*

Proof Note that,

$$\begin{aligned}
L_{\mathcal{D}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}(\mathbf{x})\right) &\leq L_{\mathcal{S}_{te}}\left(\tilde{f}_{\mathcal{A},\mathcal{B}}(\mathbf{x})\right)+\epsilon_1 && \text{Lemma 33}\\
&\leq L_{\mathcal{S}_{te}}\left(\tilde{f}_{\mathcal{A}^\star,\mathcal{B}^\star}(\mathbf{x})\right)+\epsilon_1 && \text{By algorithm}\\
&\leq L_{\mathcal{D}}\left(\tilde{f}_{\mathcal{A}^\star,\mathcal{B}^\star}(\mathbf{x})\right)+2\epsilon_1 && \text{Lemma 33}\\
&\leq 2\epsilon_1+\epsilon_2 && \text{Lemma 12}
\end{aligned}$$

and the sample complexity is $O(\epsilon^{-4}\delta^{-1}(n+\log(\delta^{-1})))$ by combining the sample complexities of Lemma 12 and 33. ■

Appendix D. Missing analysis from Section 4

D.1. Recoverability

In this section, we discuss the special case of multivariate polynomials and the notion of recoverability, i.e. approximating a function within arbitrarily good precision *everywhere* using poly-many samples. We characterize a condition on the samples for which recoverability, and thus optimization from samples, is possible for polynomials. We conclude the section by noting that recoverability is not a necessary condition for optimization from samples.

Definition 34 *We say that a function f is ϵ-recoverable over distribution $\mathcal{D}$ if given* $\mathrm{poly}(n, 1/\delta, 1/\epsilon)$ *i.i.d. samples $(\mathbf{x}, f(\mathbf{x}))$ with $\mathbf{x} \sim \mathcal{D}$, there exists a (not necessarily polynomial time) algorithm which outputs $\tilde{f}$ such that for* all $\mathbf{x}$*,*

$$|f(\mathbf{x}) - \tilde{f}(\mathbf{x})| < \epsilon$$

with probability $1 - \delta$ over the samples.

Recoverable functions are trivially optimizable from samples.

Proposition 35 *If a function f is $\epsilon/2$-recoverable given samples $\mathcal{S}$, then it is (not necessarily in polynomial time) ϵ-optimizable from samples $\mathcal{S}$.*

Proof A minimizer $\tilde{\mathbf{x}}^\star$ of any function $\tilde{f}$ which $\epsilon/2$-recovers f is an ϵ approximation to a minimizer $\mathbf{x}^\star$ of f:

$$f\left(\tilde{\mathbf{x}}^\star\right) \leq \tilde{f}\left(\tilde{\mathbf{x}}^\star\right) + \epsilon/2 \leq \tilde{f}\left(\mathbf{x}^\star\right) + \epsilon/2 \leq f\left(\mathbf{x}^\star\right) + \epsilon,$$

where the first and third inequalities are since $\tilde{f}$ $\epsilon/2$-recovers f and the second since $\tilde{\mathbf{x}}^\star$ if a minimizer of $\tilde{f}$. ■

D.2. Multivariate polynomials with full rank matrix of samples are optimizable

The class of multivariate polynomials of degree d is the class of functions such that

$$f(\mathbf{x}) = \sum_{\substack{(j_1,\ldots,j_k):\\ \sum_{k=1}^n j_k \leq d,\\ j_k \in \{0,\ldots,d\},\forall k}} \alpha_j \prod_{k=1}^n x_k^{j_k}$$

with $\alpha_j \in \mathbb{R}$ for all j. There exists an extensive literature on polynomial interpolation (De Boor and Ron, 1990; Gasca, 1990; Ben-Or and Tiwari, 1988) where the goal is to find a polynomial $\tilde{f}$ which "fits" a collection of m points $\mathcal{S} = \{(\mathbf{x}, f(\mathbf{x}))\}$ and to characterize when there is such a unique polynomial. An approach to that problem is to write the points in a system of linear equations $M\boldsymbol{\alpha} = \mathbf{f}$.

The *matrix of samples* M is an $m \times c$ matrix, where c is the number of terms in f, defined as $M_{ij} = \prod_{k=1}^n x_k^{j_k}$ for the ith sample $\mathbf{x}$. The vector $\boldsymbol{\alpha}$ is a c dimensional vector consisting of the c parameters α_j of the function f which we aim to learn. The vector $\mathbf{f}$ is the value of the ith sample $\mathbf{x}$, $f_i = f(\mathbf{x})$. Note that there always exists at least one solution $\boldsymbol{\alpha}$ to this system of linear equations, which corresponds to f, and that if M is invertible, then its solution α corresponding to f is unique, so f is 0-recoverable from $\mathcal{S}$. By Theorem 35, f is thus optimizable from samples $\mathcal{S}$. For a polynomial f which is optimizable in polynomial time in the value query model, we obtain that f is efficiently optimizable from samples.

Theorem 36 *Assume that the matrix M of samples $\mathcal{S}$ of a polynomial f is full rank, then a minimizer of f can be computed from samples $\mathcal{S}$.*

1	1	0	0	0	0	0	$\cdots$	0	0
1	0	1	0	0	0	0	$\cdots$	0	0
0	1	0	1	0	0	0	$\cdots$	0	0
1	0	0	0	1	0	0	$\cdots$	0	0
0	1	0	0	0	1	0	$\cdots$	0	0
				$\vdots$					
1	0	0	0	0	0	0	$\cdots$	0	1

Table 1: The matrix of samples M of rank $n-1$ for the hardness result for linear functions.

One might hope that if M is almost full rank then f might be approximately optimizable from $\mathcal{S}$. We show that this is not the case, even for linear functions, which are the special case of multivariate polynomials of the form $f(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j x_j$. Note that M is full rank it has rank n, We show that even if M has rank $n-1$, then we cannot obtain non-trivial guarantees for optimizing linear functions.

Theorem 37 *There exist a class of linear functions $\mathcal{F}$ and a collection of samples $\mathcal{S}$ such that the matrix M of samples has rank $n-1$ and such that $\mathcal{F}$ is not $1/2$-optimizable from samples $\mathcal{S}$.*

Proof We build two linear functions f_1 and f_2 which are indistinguishable from a collection of samples $\mathcal{S}$ and which have a $1/2$-gap. In addition, $\mathcal{S}$ has rank $n-1$.

Consider the following collection $\mathcal{S}$ of $n-1$ samples, with associated matrix M of samples illustrated in Table D.2. These samples are $\mathbf{x}^1 = (1, 1, 0, \ldots, 0)$, and for $i > 1$, $\mathbf{x}_i$ defined as $x_{i \,(\mathrm{mod}\ 2)+1} = 1$, $x_{i+1} = 1$ and $x_j = 0$ otherwise. It is easy to see that the matrix M of samples has rank $n-1$. We consider the following two functions $f_1(\mathbf{x}) = \mathbf{v}^1\mathbf{x} + 1/2$ and $f_2(\mathbf{x}) = \mathbf{v}^2\mathbf{x} + 1/2$,

$$v_j^1 = \begin{cases} 1/n & \text{if } j \ (\mathrm{mod}\ 2) = 0 \\ -1/n & \text{otherwise} \end{cases} \qquad v_j^2 = \begin{cases} -1/n & \text{if } j \ (\mathrm{mod}\ 2) = 0 \\ 1/n & \text{otherwise} \end{cases}$$

The samples all have value $1/2$ for both functions. Thus these two functions are indistinguishable from $\mathcal{S}$. Finally, note that for any $\mathbf{x}$, $f_1(\mathbf{x}) = 1 - f_2(\mathbf{x})$ and that both f_1 and f_2 have a minimizer of value 0. Thus any (possibly randomized) vector $\mathbf{x}$ returned by the algorithm is such that either $f_1(\mathbf{x}) \geq 1/2$ or $f_2(\mathbf{x}) \geq 1/2$. ■

We conclude by noting that this bound is (exactly) tight since the $1/2$ upper bound from Proposition 7 applies.

D.3. Recoverability is not necessary for optimization from samples

We have seen that recoverability is a condition that can be used successfully to show that multivariate polynomials can be optimized from samples. A natural follow-up question is whether recoverability is necessary for optimizable from samples. Similarly as for combinatorial optimization from samples (Balkanski et al., 2017), there exist functions that are not recoverable but are optimizable from samples.

Proposition 38 *There exist convex functions which are not $1/2$-recoverable over the uniform distribution $\mathcal{D}$, but that are such that a minimizer can always be computed from samples.*

Proof Consider the two functions from Section 1.1,

$$f_1(\mathbf{x}) = \max\left(0, 1 - \mathbb{1}_{[n/2]}^\intercal \mathbf{x}\right) \quad \text{and} \quad f_2(\mathbf{x}) = \max\left(0, 1 - \mathbb{1}_{\{n/2+1,\ldots,n\}}^\intercal \mathbf{x}\right)$$

The vector $(1, \ldots, 1)$ is always a minimizer for these two functions. With probability at least $1 - e^{-\Omega(n^{1/3})}$, $f_1(\mathbf{x}) = f_2(\mathbf{x}) = 0$ for all samples $\mathbf{x}$ drawn from the uniform distribution by a Chernoff bound. Thus f_1 and f_2 are indistinguishable from samples from $\mathcal{D}$. Consider $\mathbf{x} = \mathbb{1}_{[n/2]}$. Note that $f_1(\mathbf{x}) = 0$ and $f_2(\mathbf{x}) = 1$. Thus for any learned function $\tilde{f}$, either $|\tilde{f}(\mathbf{x}) - f_1(\mathbf{x})| \geq 1/2$ or $|\tilde{f}(\mathbf{x}) - f_2(\mathbf{x})| \geq 1/2$, and $\{f_1, f_2\}$ is not $1/2$-recoverable. ■

Thus, recoverability is not necessary for optimization from samples.