On Convex Optimization, Fat Shattering and Learning

Nathan Srebro
Toyota Technological Institute at Chicago
nati@ttic.edu

Karthik Sridharan
University of Pennsylvania
skarthik@wharton.upenn.edu

Abstract

Oracle complexity of the problem under the oracle based optimization model introduced by Nemirovski & Yudin (1978) is considered. We show that the oracle complexity can be lower bounded by fat-shattering dimension introduced by Kearns & Schapire (1990), a key tool in learning theory. Using this result, we proceed to establish upper bounds on learning rates for agnostic PAC learning with linear predictors in terms of oracle complexity thus showing an inherent relationship between learning and convex optimization.

1 Introduction

In the problem of convex optimization, we are interested in minimizing a given convex function f over a feasible convex set $\mathcal{W} \subset \mathcal{B}$ (where $\mathcal{B}$ is some real vector space). We say that a point $\mathbf{w} \in \mathcal{W}$ is an β sub-optimal solution if

$$f(\mathbf{w}) - \inf_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) \leq \beta$$

Given a set $\mathcal{F}$ of convex functions over the feasible set $\mathcal{W}$ we are interested in the question of efficiency of optimization procedures to produce β sub-optimal solutions while minimizing some convex function from $\mathcal{F}$. A typical convex optimization procedure initially picks some point in the feasible convex set $\mathcal{W}$ and iteratively updates these points based on some local information about the function it calculates around these successive points. Examples of these type of procedures are gradient descent methods that uses first order gradient information, the ellipsoid method, newton's method that uses second order hessian information, interior point methods and so on (see Nemirovski (2005); Nemirovski & Yudin (1978) and Boyd & Vandenberghe (2004) for survey). In fact most procedures one can think of for optimization are based on iteratively updating based on some local information about the function at the points queried so far. In general computing the exact computational complexity of these methods is cumbersome and may not even be possible in full generality. Hence to capture efficiency of an optimization procedure, number of function evaluations, number of gradient calculations etc required by the procedure to ensure sub optimality smaller than a target error rate have been used instead. In fact to capture efficiency of optimization procedures in a general way, this idea can be generalized by considering the problem of oracle based optimization problem as one by Nemirovski & Yudin (1978). The basic idea is that optimization procedures only get information about function to optimize by querying an oracle on local information about the function on the locality of the query point. The number of oracle queries made by the procedure to ensure β sub-optimality is then used for measuring efficiency of the optimization procedure and is referred to as the oracle complexity of the procedure. In this paper we are interested in giving generic lower bounds on oracle complexity of large class of commonly encountered optimization problems.

The main result of this paper is a lower bound on the oracle complexity for the problem of optimizing convex function over the feasible set $\mathcal{W}$ when the sub-gradients of the function to optimize lie in some arbitrary centrally symmetric set $\mathcal{X} \subset \mathcal{B}^\star$. We show that the oracle complexity of this optimization problem is lower bounded by the so called "fat-shattering" dimension of a linear class specified by sets $\mathcal{W}$ and $\mathcal{X}$. The fat-shattering dimension, a real valued analog of the Vapnik-Chervonenkis dimension has been an important tool used in analyzing and providing learning rates for agnostic PAC learning problems Alon *et al.* (1993). As a straightforward consequence of the main result in the paper we also provide upper bounds on learning rates of supervised learning problems with linear predictors in terms of oracle efficiency with which we can solve convex optimization problems in the appropriate domain. We further conclude that when we are interested in learning problems with convex losses, then if we can optimize efficiently we can also learn efficiently and further the learning rate can be upper bounded in terms f rate of optimization. Overall we establish some inherent connections between learning and convex optimization.

The rest of the paper is organized as follows: Section 2 describes the oracle based optimization problem and defines the oracle complexity of an optimization problem. Subsection 2.1 provides the formal description of the convex optimization problems we consider in this paper. Subsection 2.2 describes convex optimization problems commonly encountered in machine learning applications and how the setting we consider applies to these problems. Section 3 provides the main result of the paper which is the lower bound on oracle complexity in terms of fat-shattering dimension. Section 4 shows how the main result can be used to provide upper bounds on sample complexity of agnostic learning problems using oracle complexity and shows how optimization and learning are connected.

2 Oracle-Based Convex Optimization

We now proceed to define an Oracle as in Nemirovski & Yudin (1978). However before we proceed, let us first provide the general setting for the optimization problem. We start with a **convex** feasible set $\mathcal{W} \subset \mathcal{B}$ where $\mathcal{B}$ is a topological vector space over reals. The convex function we are to optimize belongs to some given set $\mathcal{F}$ of convex functions over $\mathcal{W}$. We use the term oracle to refer to any mapping $\mathcal{O} : \mathcal{W} \times \mathcal{F} \mapsto \mathcal{I}$, where set $\mathcal{I}$ is some arbitrary information set $\mathcal{I}$. However at this generality note that the information set $\mathcal{I}$ could be all of $\mathcal{F}$ and Oracle $\mathcal{O}$ could be the identity mapping. This would defeat the purpose of introducing oracle models and associated oracle complexity. To address this issue we use the notion of local oracle defined by Nemirovski and Yudin Nemirovski & Yudin (1978) and whenever we use the term oracle we mean local oracle.

Definition 1. A (local) oracle $\mathcal{O} : \mathcal{W} \times \mathcal{F} \mapsto \mathcal{I}$ is a mapping, which, given any query point $\mathbf{w} \in \mathcal{W}$ and instances $f, f' \in \mathcal{F}$ such that $f \equiv f'$ in some neighborhood of $\mathbf{w}$ (ie. functions are identical in the neighborhood), outputs answers that satisfy the equality $\mathcal{O}(\mathbf{w}, f) = \mathcal{O}(\mathbf{w}', f)$.

Typically if we had some metric on $\mathcal{W}$, then the neighborhood above can be taken to be open ball around query point $\mathbf{w}$.

At a high level, the definition basically says if two convex functions f and f' are indistinguishable about some neighborhood of a point $\mathbf{w} \in \mathcal{W}$, then querying an oracle about the two instances at this point $\mathbf{w}$ leads to the same answer. For a query with instance $\mathbf{w} \in \mathcal{W}$ at instance $f \in \mathcal{F}$, the oracle only provides information about local structure of the function f around the point of query $\mathbf{w}$. To make the concept clearer we provide the following examples of (local) oracles commonly used in practice. Figure 1 illustrates the concept of a local oracle. Examples of local oracles are the zero'th order oracle or function evaluations

which gives rise to bandit optimization problem, first order oracles that provide sub-gradient information which includes methods like gradient descent, the ellipsoid method, accelerated gradient method etc., the second order oracle that provides hessian information and includes methods like newton's method etc.

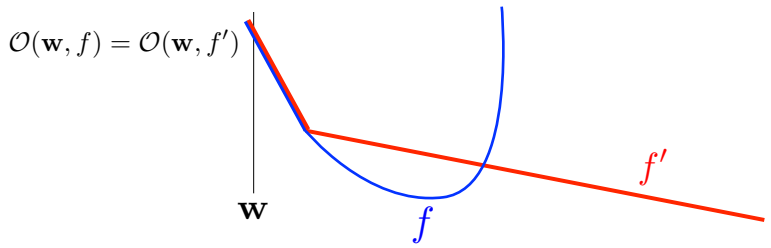

Figure 1: *In the neighborhood of* $\mathbf{w}$, $f \equiv f'$ *and so oracle queries about the two functions at* $\mathbf{w}$ *match.*

We now describe the oracle-based offline convex optimization protocol. Given convex function f, the oracle-based optimization procedure is as follows :

Oracle-based Optimization Protocol :
for $t = 1$ **to** m
Pick $\mathbf{w}_t \in \mathcal{W}$ for query
Oracle provides answer $\mathrm{I}_t = \mathcal{O}(\mathbf{w}_t, f)$
end for

At the end of the m iterations, the sub-optimality of the procedure is given by $f(\mathbf{w}_m) - \inf_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w})$.

Definition 2. For a given Oracle $\mathcal{O}$, an "Oracle Based Optimization Algorithm", $\mathbf{A}^{\mathcal{O}} : \bigcup_{n\in\mathbb{N}} \mathcal{I}^n \to \mathcal{W}$ is a mapping from a sequences of oracle answers in $\mathcal{I}$ to an element of the feasible set $\mathcal{W}$.

Given an Oracle-based optimization algorithm, $\mathbf{A}^{\mathcal{O}}$ we shall use the short-hand, $\mathrm{I}_1 = \mathcal{O}(\mathbf{w}_1, f)$ and further iteratively, use the notation $\mathrm{I}_t = \mathcal{O}(\mathbf{w}_t, f)$ where of course each query point $\mathbf{w}_t = \mathbf{A}^{\mathcal{O}}(\mathrm{I}_1, \ldots, \mathrm{I}_{t-1})$ is picked using the Oracle-based optimization algorithm. We now define oracle complexity of an optimization problem using oracle $\mathcal{O}$.

Definition 3. For a given class of convex functions $\mathcal{F}$ over the set $\mathcal{W}$ and some Oracle $\mathcal{O}$, the offline oracle complexity of a given "Oracle based optimization algorithm", $\mathbf{A}^{\mathcal{O}}$, is defined as

$$m(\beta, \mathcal{F}, \mathbf{A}^{\mathcal{O}}) = \inf \left\{ m \in \mathbb{N} \;\middle|\; \sup_{f\in\mathcal{F}} \left\{ f(\mathbf{A}^{\mathcal{O}}(\mathrm{I}_1, \ldots, \mathrm{I}_m)) - \inf_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) \right\} \leq \beta \right\}$$

Further, for the given oracle $\mathcal{O}$, the $\mathcal{O}$-oracle complexity of the given offline optimization problem is defined as $m(\beta, \mathcal{O}, \mathcal{F}) = \inf_{\mathbf{A}^{\mathcal{O}}} m(\beta, \mathcal{F}, \mathbf{A}^{\mathcal{O}})$. Finally the oracle complexity of the given optimization problem is defined as

$$m(\beta, \mathcal{F}) = \inf_{\mathcal{O}} m(\beta, \mathcal{F}, \mathcal{O}) \; .$$

where the infimum above is over local oracles.

Roughly speaking, given $\beta > 0$, the oracle complexity of an algorithm is the minimum number of oracle answers needed by the algorithm to guarantee sub-optimality smaller than β against any convex objective $f \in \mathcal{F}$. Further, the above definition basically implies that for any β, there exists an Oracle Based

Optimization Algorithm, $\mathbf{A}^{\mathcal{O}}$ that provides an β sub-optimality for any $f \in \mathcal{F}$ within $m(\beta, \mathcal{F}, \mathcal{O})$ steps (or oracle queries). Finally the definition of oracle complexity of the optimization problem, $m(\beta, \mathcal{F})$ tells us that there exists some local oracle $\mathcal{O}$ and an associated oracle based optimization algorithm, $\mathbf{A}^{\mathcal{O}}$ that provides an β sub-optimality for any $f \in \mathcal{F}$ within $m(\beta, \mathcal{F})$ steps.

2.1 Convex Lipschitz Optimization

Perhaps the most commonly studied convex optimization problem is the optimization of convex function over domain $\mathcal{W}$ that are Lipschitz w.r.t. some underlying norm on $\mathcal{W}$. That is functions $f \in \mathcal{F}$ considered are convex and for any $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$,

$$|f(\mathbf{w}) - f(\mathbf{w}')| \leq L \left\|\mathbf{w} - \mathbf{w}'\right\|$$

for some given norm $\|\cdot\|$. Notice that a convex function f is L-Lipschitz w.r.t. a norm $\|\cdot\|$ if and only if for any $\mathbf{w} \in \mathcal{W}$, the sub-gradient of the function at $\mathbf{w}$, that is $\nabla f(\mathbf{w}) \in \mathcal{B}^\star$ ($\mathcal{B}^\star$ is the dual of vector space $\mathcal{B}$) belongs to the ball w.r.t. the dual norm $\|\cdot\|_*$ of radius at most L. In this paper we consider a generalization of such convex optimization problem where we consider a set $\mathcal{X} \subset \mathcal{B}^\star$ assumed to be a **centrally symmetric** subset of the dual space $\mathcal{B}^\star$. We consider optimization of convex functions belonging to the class of convex functions specified as :

$$\mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X}) = \{f : \mathcal{W} \mapsto \mathbb{R} \mid f \text{ is convex and } \forall \mathbf{w} \in \mathcal{W},\ \nabla f(\mathbf{w}) \in \mathcal{X}\}$$

Notice that when $\mathcal{X}$ is L times the unit ball of the dual norm $\|\cdot\|_*$ (which is a dual norm of some given norm $\|\cdot\|$) then this problem coincides with convex L-Lipschitz optimization problem over $\mathcal{W}$ where the functions are L-Lipschitz w.r.t. norm $\|\cdot\|$. However for the results we provide in the paper we only require set $\mathcal{X}$ to be centrally symmetric meaning that if $\mathbf{x} \in \mathcal{X}$ then so is $-\mathbf{x}$.

Commonly considered examples are the case when $\mathcal{W}$ is unit ball of some norm $\|\cdot\|$ and $\mathcal{X}$ is the unit ball (or appropriately scaled by some other radius) of the dual norm $\|\cdot\|_*$. We list below a couple of such commonly encountered examples.

Example 1 (ℓ_2/ℓ_2 dual case)**.** In this example the set $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_2 \leq 1\}$ the unit ball ℓ_2 ball and $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_2 \leq 1\}$ the dual ball. In this case, when dimensionality d is small compared to $1/\beta^2$, then $m(\beta, \mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})) \approx d\log(1/\beta)$. For large scale problems when dimensionality is large compared to $1/\beta^2$, $m(\beta, \mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})) \approx 1/\beta^2$. Further the former bound when dimensionality is small can be achieved using cutting plane methods like center of gravity method (and more efficiently for example with ellipsoid method). The later bound can be attained using gradient descent method.

Example 2 (ℓ_1/ℓ_∞ dual case)**.** In this example the set $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_1 \leq 1\}$ the unit ball ℓ_1 ball and $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_\infty \leq 1\}$ the dual ball. In this case again as in the previous example, when dimensionality d is small (say constant) w.r.t. $1/\beta$ then using cutting plane method one gets $m(\beta, \mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})) \approx d\log(1/\beta)$ and for large scale problems (when dimensionality is large), $m(\beta, \mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})) \approx \log(d)/\beta^2$ and this bound can be attained using appropriate mirror descent method.

Example 3 (ℓ_1/ℓ_1 non-dual case)**.** In this example the set $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_1 \leq 1\}$ the unit ball ℓ_1 ball and $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\|_1 \leq 1\}$ (not the dual set). In this case again as in the previous example, when dimensionality d is small (say constant) w.r.t. $1/\beta$ then using cutting plane method one gets $m(\beta, \mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})) \approx d\log(1/\beta)$. For large scale problems (when dimensionality is large), $m(\beta, \mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})) \approx 1/\beta^2$ and this bound can be attained basically using gradient descent with appropriate step size.

In all of this for low dimensional problems (dimension low compared to required accuracy) ellipsoid method gives more or less the right rate. However in many machine learning applications we deal with large scale problems and the required accuracy for the optimization problem is not as small (we don't need to be more accurate than the noise in estimation error). For more examples of rates for non-dual and dual cases in high dimension one can refer to Srebro *et al.* (2011).

2.2 Optimization for Learning Problems

Optimization plays a central role in machine learning problems. Most learning algorithms can be seen as simply solving an optimization problem of minimizing some empirical objective (w.r.t. training sample) over set of feasible hypothesis. As our interest here is also in the connections between optimization and learning, we consider the optimization problems arising from machine learning applications. A typical approach in machine learning is to seek a predictor minimizing the training error (empirical risk) over the training sample. Consider the problem of supervised learning where the learner is provided with a training data set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ of instance label pairs and say the input instances $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are represented as vectors from some set $\mathcal{X} \subset \mathcal{B}^\star$ and labels $y_1, \ldots, y_n \in \mathcal{Y}$ are from arbitrary label space. Now say we are interested in using the linear hypothesis class for our prediction problem where the linear predictors we use are from a set $\mathcal{W} \subset \mathcal{B}$. Given an instance $(\mathbf{x}, y)$ the loss suffered by using linear predictor $\mathbf{w}$ for this instance is given by $\ell(\langle \mathbf{w}, \mathbf{x} \rangle, y)$ where $\ell : \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}$ [1]. In this case the empirical risk minimization algorithm that is commonly used amounts to solving the optimization problems

$$\min_{\mathbf{w} \in \mathcal{W}} \widehat{L}(\mathbf{w}) := \min_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \ell(\langle \mathbf{w}, \mathbf{x}_i \rangle, y_i)$$

We further consider the case when the loss ℓ used is convex in its first argument and is such that $\forall y \in \mathcal{Y}$, $|\partial_{\hat{y}} \ell(\hat{y}, y)| \leq 1$ (or in other words is 1-Lipschitz). This is the case for e.g. when we consider the absolute loss, hinge loss (used in SVMs), the logistic loss or even the squared loss when the domain is bounded. (of course one can have a Lipschitz constant other than 1 and in this case all results hold with appropriate rescaling). In this setting, note that the function $\widehat{L}$ is convex over the domain $\mathcal{W}$ and further, for any $\mathbf{w} \in \mathcal{W}$

$$\nabla \widehat{L}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \partial \ell(\langle \mathbf{w}, \mathbf{x}_i \rangle, y_i) \mathbf{x}_i$$

Hence we can conclude that $\nabla \widehat{L}(\mathbf{w}) \in \operatorname{abscond}(\mathcal{X})$ or if $\mathcal{X}$ is convex and centrally symmetric then simply that for any $\mathbf{w} \in \mathcal{W}$, $\nabla \widehat{L}(\mathbf{w}) \in \mathcal{X}$. Hence we see that the empirical risk minimization problem lies in $\mathcal{F}_{\mathrm{Lip}}(\mathcal{W}, \mathcal{X})$ and one can use optimization procedures for optimization over class $\mathcal{F}_{\mathrm{Lip}}(\mathcal{W}, \mathcal{X})$ to solve the learning problems.

3 Lower Bound On Oracle Complexity

We are now ready to present our lower bound on the oracle complexity of optimizing convex functions from function class $\mathcal{F}_{\mathrm{Lip}}(\mathcal{X}, \mathcal{W})$. Our lower bound is given in terms of the fat-shattering dimension of the linear class

$$\operatorname{lin}(\mathcal{W}, \mathcal{X}) = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle : \mathbf{w} \in \mathcal{W}\}$$

[1] We use the notation $\langle \mathbf{w}, \mathbf{x} \rangle$ to represent the linear functional $\mathbf{x}$ of the dual space being applied to $\mathbf{w}$ and is not necessarily the inner product. For example it is not symmetric and $\mathbf{w}$ and $\mathbf{x}$ are not from the same space

The fat-shattering is a combinatorial measure generalizing the VC dimension to real-valued functions, and sensitive to the scale of the functions. The fat-shattering dimension was introduced byKearns & Schapire (1990) and is key concept in learning theory. In Alon *et al.* (1993) it has been shown that finiteness of fat-shattering dimension characterizes agnostic learnability of the function class. The fat-shattering dimension is defined as follows :

Definition 4. Set of points $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ is said to be β-shattered by function class $\text{lin}(\mathcal{W}, \mathcal{X})$ if there exists scalars $s_1, \ldots, s_m \in \mathbb{R}$ such that for every sign pattern $(\epsilon_1, \ldots, \epsilon_m) \in \{\pm 1\}^m$, there exists $\mathbf{w} \in \mathcal{W}$ such that

$$\forall i \in [m],\ \epsilon_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - s_i) > \beta/2$$

Further the fat shattering dimension of the function class $\text{lin}(\mathcal{W}, \mathcal{X})$ at scale β is defined as the largest number of points that can be β-shattered by $\text{lin}(\mathcal{W}, \mathcal{X})$.

We are now ready to provide the main lower bound result.

Theorem 1. *For any convex set $\mathcal{W} \subset \mathcal{B}$ any centrally symmetric set $\mathcal{X} \subset \mathcal{B}^\star$ and any $\beta > 0$:*

$$m(\beta, \mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})) \geq \text{fat}_{2\beta}(\text{lin}(\mathcal{W}, \mathcal{X}))\ .$$

Proof. The proof uses ideas from Nemirovski & Yudin (1978) (Section 4.4.2) and basically proceeds by constructing piecewise linear convex function. Note that for any piecewise linear function, the only local information present at any point are the function value and the set of sub-gradients at the point. Hence it is enough to only show lower bounds w.r.t. algorithms that only have access to function value and sub-gradients at query point.
To prove the lower bound, we first start by picking $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ and $s_1, \ldots, s_m \in \mathbb{R}$. Now the functions we will construct after the m steps will have a form of

$$f_\epsilon(\mathbf{w}; (\mathbf{x}_1, s_1), \ldots, (\mathbf{x}_m, s_m)) = \max_{i \in [m]} \epsilon_i(\langle \mathbf{w}, -\mathbf{x}_i \rangle + s_i)$$

where $\epsilon \in \{\pm 1\}^m$. Notice that each $f_\epsilon \in \mathcal{F}_{\text{Lip}}(\mathcal{X})$. Note also that for any $\epsilon \in \{\pm 1\}^m$,

$$-\inf_{\mathbf{w} \in \mathcal{W}} f_\epsilon(\mathbf{w}; (\mathbf{x}_1, s_1), \ldots, (\mathbf{x}_m, s_m)) = -\inf_{\mathbf{w} \in \mathcal{W}} \max_{i \in [m]} \epsilon_i(\langle \mathbf{w}, -\mathbf{x}_i \rangle + s_i) = \sup_{\mathbf{w} \in \mathcal{W}} \min_{i \in [m]} \epsilon_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - s_i) \quad (1)$$

Recall that we want to show that for any $\mathbf{A}^{\mathcal{O}}$ there exists a function that requires at least m calls to some Oracle $\mathcal{O}$ to ensure sub-optimality less than $\beta > 0$. As mentioned earlier, since the family of functions we consider are piece wise linear, any local oracle can give no more information than function value and sub-gradients at point any query point. Now given an Optimization algorithm $\mathbf{A}^{\mathcal{O}}$ the exact function we shall use for the lower bound will be constructed in m steps based on the algorithm and the choosen $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ add $s_1, \ldots, s_m \in \mathbb{R}$. The procedure for constructing the function is given below :

Initialize $A_1 = [m]$
for $t = 1$ **to** m
$\mathbf{A}^{\mathcal{O}}$ *picks* $\mathbf{w}_t \in \mathcal{W}$ *for query*
$i(t) = \underset{i \in A_t}{\operatorname{argmax}} \{|\langle \mathbf{w}_t, -\mathbf{x}_i \rangle + s_i|\}$
$\epsilon_t = \begin{cases} +1 & \text{if } (\langle \mathbf{w}_t, -\mathbf{x}_{i(t)} \rangle + s_{i(t)}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$
$A_{t+1} = A_t \setminus \{i(t)\}$
$f^t(\mathbf{w}) = \max_{j \in [t]} \left\{ \epsilon_j (\langle \mathbf{w}, -\mathbf{x}_{i(j)} \rangle + s_{i(j)}) \right\}$
Oracle return answer $\mathrm{I}_t = \mathcal{O}(\mathbf{w}_t, f^t)$.
end for

The first thing we notice about f^m is that it is of the form :

$$f^m(\cdot) = f_\epsilon(\cdot; (\mathbf{x}_{i(1)}, s_{i(1)}), \ldots, (\mathbf{x}_{i(m)}, s_{i(m)})) \tag{2}$$

where $\epsilon_1, \ldots, \epsilon_m$ are given by the procedure above. Next, f^m is such that, for any $i \in [m]$ and any local oracle $\mathcal{O}$,

$$\mathcal{O}(\mathbf{w}_i, f^m) = \mathcal{O}(\mathbf{w}_i, f^i)$$

hence the $\mathbf{w}_1, \ldots, \mathbf{w}_m$ returned by the algorithm when it is presented with function f^m is the same as the corresponding ones in the above procedure. Or in other words it is as if to begin with, the optimization algorithm was given function f^m to be optimized. Finally, by the way the functions are constructed (specifically the way ϵ_t is picked),

$$f^m(\mathbf{w}_m) \geq 0$$

Hence we conclude that

$$\begin{aligned} f^m(\mathbf{w}_m) - \inf_{\mathbf{w} \in \mathcal{W}} f^m(\mathbf{w}) &\geq - \inf_{\mathbf{w} \in \mathcal{W}} f^m(\mathbf{w}) \\ &= - \inf_{\mathbf{w} \in \mathcal{W}} z_\epsilon(\mathbf{w}; (\mathbf{x}_{i(1)}, s_{i(1)}), \ldots, (\mathbf{x}_{i(m)}, s_{i(m)})) && \text{(by Eq. 4)} \\ &= \sup_{\mathbf{w} \in \mathcal{W}} \min_{i \in [m]} \epsilon_i (\langle \mathbf{w}, \mathbf{x}_i \rangle - s_i) && \text{(by Eq. 1)} \\ &\geq \inf_{\epsilon \in \{\pm 1\}^m} \sup_{\mathbf{w} \in \mathcal{W}} \min_{i \in [m]} \epsilon_i (\langle \mathbf{w}, \mathbf{x}_i \rangle - s_i) \ . \end{aligned}$$

Furthermore note that the choice of $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ and $s_1, \ldots, s_m$ are arbitrary. Hence we can conclude that for any $\beta > 0$, if for any m there exists $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ and $s_1, \ldots, s_m \in \mathbb{R}$ such that

$$\inf_{\epsilon \in \{\pm 1\}^m} \sup_{\mathbf{w} \in \mathcal{W}} \min_{i \in [m]} \epsilon_i (\langle \mathbf{w}, \mathbf{x}_i \rangle - s_i) > \beta \ ,$$

then no oracle based algorithm can achieve sub-optimality smaller than β in m or less steps. However note that this is exactly the definition of fat-shattering dimension at scale 2β (Definition 4) for the linear class $\mathrm{lin}(\mathcal{W}, \mathcal{X})$. Hence we conclude the theorem statement. □

The lower bound is a lower bound w.r.t. any local oracle and so can be viewed also as a lower bound on number of gradient evaluations needed by any gradient based algorithm for instance.

3.1 Uniformly Convex Programming

In this subsection we shall consider convex optimization problem where the functions we optimize are uniformly convex. More specifically, we would like to optimize over convex set $\mathcal{W}$ as before. However we assume that the set $\mathcal{X}$ is the unit ball w.r.t. some norm $\|\cdot\|_{\mathcal{X}}$ and the functions we are interested in optimizing are ones whose sub-gradients lie in $\mathcal{X}$ and are (λ, q)-uniformly convex w.r.t. norm $\|\cdot\|_{\mathcal{X}^*}$ where $\|\cdot\|_{\mathcal{X}^*}$ is the dual norm of $\|\cdot\|_{\mathcal{X}}$. Throughout this sub-section we also assume that $\forall \mathbf{w} \in \mathcal{W}, \|\mathbf{w}\|_{\mathcal{X}^*} \leq 1$

Recall that a function $f : \mathcal{W} \mapsto \mathbb{R}$ is said to be (λ, q)-uniformly convex w.r.t. norm $\|\cdot\|$ if for any $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$ and any $\alpha \in [0, 1]$,

$$f\left(\alpha\mathbf{w} + (1-\alpha)\mathbf{w}'\right) \leq \alpha f(\mathbf{w}) + (1-\alpha) f(\mathbf{w}') - \frac{\lambda\alpha(1-\alpha)}{q}\left\|\mathbf{w} - \mathbf{w}'\right\|^q$$

Note that $q = 2$ is the special case of strong convexity. Before we state the main result of this section it is important to note that for a given vector space $\mathcal{B}$ and norm $\|\cdot\|_{\mathcal{X}^*}$ on it, if we fix some arbitrary $\lambda > 0$, it might not even be possible to find a function that is 1-Lipschitz w.r.t. $\|\cdot\|_{\mathcal{X}^*}$ and (λ, q)-uniformly convex w.r.t. the same norm. In fact for every Banach space $(\mathcal{B}, \|\cdot\|_{\mathcal{X}})$ there exists a minimum $q^* \geq 2$ associated with the space for which one can find a finite λ such that there exists a function that is (λ, q^*)-uniformly convex. Let us define

$$\lambda_q(\mathcal{X}) = \inf\left\{\lambda : \exists f : \mathcal{W} \mapsto \mathbb{R} \text{ that is } (\lambda, q)\text{-uniformly convex w.r.t. } \|\cdot\|_{\mathcal{X}^*} \text{ and is 1-Lipschitz w.r.t. } \|\cdot\|_{\mathcal{X}^*}\right\}$$

Theorem 2. *For any convex set $\mathcal{W} \subset \mathcal{B}$ any set $\mathcal{X} \subset \mathcal{B}^\star$, unit ball of some norm $\|\cdot\|_{\mathcal{X}}$, any $q \geq 2, \lambda < \lambda_q(\mathcal{X})$ and any $\beta > 0$:*

$$m(\beta, \mathcal{F}_{\text{ucvx}(q,\lambda)}(\mathcal{W}, \mathcal{X})) \geq \text{fat}_{\frac{4\lambda\beta^{1/p}}{(\lambda_q(\mathcal{X})-\lambda)}}(\text{lin}(\mathcal{W}, \mathcal{X})) \ .$$

Proof. We start by noting that by definition of $\lambda_q(\mathcal{X})$ for any $\lambda' < \lambda_q(\mathcal{X})$, there exists a function $\phi : \mathcal{W} \mapsto \mathbb{R}$ that is lower semi-continuous on $\mathcal{W}$, continuous at 0, and is (λ', q)-uniformly convex w.r.t. norm $\|\cdot\|_{\mathcal{X}^*}$ and 1-Lipschitz w.r.t. the same norm. By **?** (see Theorem 2.4), using this function, one can define a norm $\|\cdot\|$ on $\mathcal{B}$ such that $\|\cdot\| \leq \|\cdot\|_{\mathcal{X}^*} \leq \frac{1}{\lambda'^{1/q}}\|\cdot\|$ (i.e.. equivalent to $\|\cdot\|_{\mathcal{X}^*}$) further this norm is such that the function $\|\cdot\|^q/q$ is $(1, q)$-uniformly convex w.r.t. $\|\cdot\|$. Note that since $\|\cdot\|_{\mathcal{X}^*} \leq \frac{1}{\lambda'^{1/q}}\|\cdot\|$, the function $\|\cdot\|^q/q$ is (λ', q) uniformly convex w.r.t. $\|\cdot\|_{\mathcal{X}^*}$ and further since $\|\cdot\| \leq \|\cdot\|_{\mathcal{X}^*}$ the function is also 1-Lipschitz w.r.t. norm $\|\cdot\|_{\mathcal{X}^*}$. Now for any $\lambda < \lambda_q(\mathcal{X})$ we shall pick λ' (to be specified later) such that $\lambda < \lambda' < \lambda_q(\mathcal{X})$. As mentioned we can find a norm $\|\cdot\|$ such that $\|\cdot\| \leq \|\cdot\|_{\mathcal{X}^*} \leq \frac{1}{\lambda'^{1/q}}\|\cdot\|$ and the function $\|\cdot\|^q/q$ is (λ', q) uniformly convex w.r.t. $\|\cdot\|_{\mathcal{X}^*}$ and is 1-Lipschitz. Now note that the function $\frac{\lambda}{\lambda'}\|\cdot\|$ is (λ, q) uniformly convex w.r.t. $\|\cdot\|_{\mathcal{X}^*}$ and is $\frac{\lambda}{\lambda'}$-Lipschitz w.r.t. the same norm.

Now to prove the lower bound just as in the convex Lipschitz programming case, we first start by picking $\mathbf{x}_1, \ldots, \mathbf{x}_m \in (1 - \frac{\lambda}{\lambda'})\mathcal{X}$ and $s_1, \ldots, s_m \in \mathbb{R}$. Now the functions we will construct after the m steps will have a form of

$$f_\epsilon(\mathbf{w}; (\mathbf{x}_1, s_1), \ldots, (\mathbf{x}_m, s_m)) = \max_{i\in[m]} \epsilon_i(\langle \mathbf{w}, -\mathbf{x}_i\rangle + s_i) + \frac{\lambda}{\lambda'}\|\mathbf{w}\|^q$$

where $\epsilon \in \{\pm 1\}^m$. Notice that since $\frac{\lambda}{\lambda'}\|\mathbf{w}\|^q$ is (λ, q) uniformly convex w.r.t. $\|\cdot\|_{\mathcal{X}^*}$, the function f_ϵ is also (λ, q) uniformly convex. Also note that $\frac{\lambda}{\lambda'}\|\mathbf{w}\|^q$ is $\frac{\lambda}{\lambda'}$-Lipschitz and each $\mathbf{x}_i \in (1 - \frac{\lambda}{\lambda'})\mathcal{X}$. Hence we can conclude that the functions f_ϵ are $\frac{\lambda}{\lambda'} + (1 - \frac{\lambda}{\lambda'}) = 1$-Lipschitz as required. Effectively we give

a lower bound by only considering a subset of functions of the form of piece-wise linear + fixed function $\frac{\lambda}{\lambda'}\|\cdot\|^q$. Now consider what information a local oracle can provide at any point for such a function. Given that before we start the optimization algorithm already has knowledge of function $\frac{\lambda}{\lambda'}\|\cdot\|^q$ the only remaining local information the oracle can provide on each round is the local information about the piece-wise linear function. More formally, around any point $\mathbf{w}$, since $\frac{\lambda}{\lambda'}\|\cdot\|^q$ is a fixed function, two functions f, f' from the family of functions consisting of piecewise linear function + the fixed function are equivalent around every neighborhood if and only if the corresponding piecewise linear functions are equivalent. Thus beyond the fixed function $\frac{\lambda}{\lambda'}\|\cdot\|^q$ the local information the oracle can provide at any point is only the function value of the piecewise linear part and the corresponding sub-gradients. Now note that for each $\epsilon \in \{\pm 1\}^m$ we have,

$$\begin{aligned} -\inf_{\mathbf{w}\in\mathcal{W}} f_\epsilon(\mathbf{w}; (\mathbf{x}_1, s_1), \ldots, (\mathbf{x}_m, s_m)) &= -\inf_{\mathbf{w}\in\mathcal{W}} \left\{ \max_{i\in[m]} \epsilon_i(\langle \mathbf{w}, -\mathbf{x}_i\rangle + s_i) + \frac{\lambda}{\lambda'}\|\mathbf{w}\|^q \right\} \\ &= \sup_{\mathbf{w}\in\mathcal{W}} \left\{ \min_{i\in[m]} \epsilon_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - s_i) - \frac{\lambda}{\lambda'}\|\mathbf{w}\|^q \right\} \end{aligned} \tag{3}$$

Recall that we want to show that for any $\mathbf{A}^{\mathcal{O}}$ there exists a function that requires at least m calls to some Oracle $\mathcal{O}$ to ensure sub-optimality less than $\beta > 0$. As mentioned earlier, since the family of functions we consider are piece wise linear + the fixed function $\frac{\lambda}{\lambda'}\|\cdot\|^q$, any local oracle can give no more information than function value and sub-gradients at point any query point. Now given an Optimization algorithm $\mathbf{A}^{\mathcal{O}}$ the exact function we shall use for the lower bound will be constructed in m steps based on the algorithm and the choosen $\mathbf{x}_1, \ldots, \mathbf{x}_m \in (1 - \frac{\lambda}{\lambda'})\mathcal{X}$ add $s_1, \ldots, s_m \in \mathbb{R}$. The procedure for constructing the function is given below :

Initialize $A_1 = [m]$
for $t = 1$ **to** m
  $\mathbf{A}^{\mathcal{O}}$ *picks* $\mathbf{w}_t \in \mathcal{W}$ *for query*
  $i(t) = \underset{i\in A_t}{\operatorname{argmax}} \{|\langle \mathbf{w}_t, -\mathbf{x}_i\rangle + s_i|\}$
  $\epsilon_t = \begin{cases} +1 & \text{if } (\langle \mathbf{w}_t, -\mathbf{x}_{i(t)}\rangle + s_{i(t)}) \geq 0 \\ -1 & \text{otherwise} \end{cases}$
  $A_{t+1} = A_t \setminus \{i(t)\}$
  $f^t(\mathbf{w}) = \max_{j\in[t]} \{\epsilon_j(\langle \mathbf{w}, -\mathbf{x}_{i(j)}\rangle + s_{i(j)})\} + \lambda\|\mathbf{w}\|^q$
  Oracle return answer $\mathrm{I}_t = \mathcal{O}(\mathbf{w}_t, f^t)$.
end for

The first thing we notice about f^m is that it is of the form :

$$f^m(\cdot) = f_\epsilon(\cdot; (\mathbf{x}_{i(1)}, s_{i(1)}), \ldots, (\mathbf{x}_{i(m)}, s_{i(m)})) \tag{4}$$

where $\epsilon_1, \ldots, \epsilon_m$ are given by the procedure above. Next, f^m is such that, for any $i \in [m]$ and any local oracle $\mathcal{O}$,

$$\mathcal{O}(\mathbf{w}_i, f^m) = \mathcal{O}(\mathbf{w}_i, f^i)$$

hence the $\mathbf{w}_1, \ldots, \mathbf{w}_m$ returned by the algorithm when it is presented with function f^m is the same as the corresponding ones in the above procedure. Or in other words it is as if to begin with, the optimization algorithm was given function f^m to be optimized. Finally, by the way the functions are constructed (specifically the way ϵ_t is picked),

$$f^m(\mathbf{w}_m) \geq 0$$

Hence we conclude that

$$
\begin{aligned}
f^m(\mathbf{w}_m) - \inf_{\mathbf{w}\in\mathcal{W}} f^m(\mathbf{w}) &\geq - \inf_{\mathbf{w}\in\mathcal{W}} f^m(\mathbf{w}) \\
&= - \inf_{\mathbf{w}\in\mathcal{W}} f_\epsilon(\mathbf{w}; (\mathbf{x}_{i(1)}, s_{i(1)}), \ldots, (\mathbf{x}_{i(m)}, s_{i(m)})) && \text{(by Eq. 4)} \\
&= \sup_{\mathbf{w}\in\mathcal{W}} \left\{ \min_{i\in[m]} \epsilon_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - s_i) - \frac{\lambda}{\lambda'} \|\mathbf{w}\|^q \right\} && \text{(by Eq. 3)} \\
&\geq \inf_{\epsilon\in\{\pm1\}^m} \sup_{\mathbf{w}\in\mathcal{W}} \left\{ \min_{i\in[m]} \epsilon_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - s_i) - \frac{\lambda}{\lambda'} \|\mathbf{w}\|^q \right\} \\
&\geq \inf_{\epsilon\in\{\pm1\}^m} \sup_{\mathbf{w}\in\mathcal{W}} \left\{ \min_{i\in[m]} \epsilon_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - s_i) - \frac{\lambda}{\lambda'} \|\mathbf{w}\|_{\mathcal{X}^*}^q \right\}
\end{aligned}
$$

Now for any β such that $\beta \leq \left(\frac{\lambda}{\lambda'}\right)^q$

$$
\geq \inf_{\epsilon\in\{\pm1\}^m} \sup_{\mathbf{w}\in\frac{\lambda'\beta^{1/q}}{\lambda}\mathcal{W}} \left\{ \min_{i\in[m]} \epsilon_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - s_i) - \frac{\lambda}{\lambda'} \|\mathbf{w}\|_{\mathcal{X}^*}^q \right\}
$$

since $\forall \mathbf{w} \in \mathcal{W}$, $\|\mathbf{w}\|_{\mathcal{X}^*} \leq 1$ we can conclude that for all $\mathbf{w} \in \frac{\lambda'\beta^{1/q}}{\lambda}\mathcal{W}$, we have $\|\mathbf{w}\|_{\mathcal{X}^*} \leq \frac{\lambda'\beta^{1/q}}{\lambda}$ and so

$$
\begin{aligned}
&\geq \inf_{\epsilon\in\{\pm1\}^m} \sup_{\mathbf{w}\in\frac{\lambda'\beta^{1/q}}{\lambda}\mathcal{W}} \left\{ \min_{i\in[m]} \epsilon_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - s_i) \right\} - \beta \\
&= \inf_{\epsilon\in\{\pm1\}^m} \sup_{\mathbf{w}\in\mathcal{W}} \left\{ \min_{i\in[m]} \epsilon_i(\tfrac{\lambda'\beta^{1/q}}{\lambda} \langle \mathbf{w}, \mathbf{x}_i\rangle - s_i) \right\} - \beta \\
&= \inf_{\epsilon\in\{\pm1\}^m} \sup_{\mathbf{w}\in\mathcal{W}} \left\{ \min_{i\in[m]} \epsilon_i(\tfrac{\lambda'\beta^{1/q}}{\lambda}(1-\tfrac{\lambda}{\lambda'}) \left\langle \mathbf{w}, \frac{\mathbf{x}_i}{(1-\frac{\lambda}{\lambda'})} \right\rangle - s_i) \right\} - \beta \\
&= \beta^{1/q}(\tfrac{\lambda'}{\lambda} - 1) \inf_{\epsilon\in\{\pm1\}^m} \sup_{\mathbf{w}\in\mathcal{W}} \left\{ \min_{i\in[m]} \epsilon_i(\langle \mathbf{w}, \tilde{\mathbf{x}}_i\rangle - \tfrac{s_i}{\beta^{1/q}(\frac{\lambda'}{\lambda}-1)}) \right\} - \beta \ .
\end{aligned}
$$

where $\tilde{\mathbf{x}}_i = \frac{\mathbf{x}_i}{(1-\frac{\lambda}{\lambda'})}$. Furthermore note that the choice of $\mathbf{x}_1, \ldots, \mathbf{x}_m \in (1-\frac{\lambda}{\lambda'})\mathcal{X}$ and $s_1, \ldots, s_m$ are arbitrary. Also note that $\tilde{\mathbf{x}}_i \in \mathcal{X}$. Hence we can conclude that for any $\beta > 0$, if for any m there exists $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{X}$ and $s_1, \ldots, s_m \in \mathbb{R}$ such that

$$
\inf_{\epsilon\in\{\pm1\}^m} \sup_{\mathbf{w}\in\mathcal{W}} \min_{i\in[m]} \epsilon_i(\langle \mathbf{w}, \mathbf{x}_i\rangle - s_i) > \frac{2\beta}{\beta^{1/q}(\frac{\lambda'}{\lambda} - 1)} = \frac{2\beta^{1/p}}{(\frac{\lambda'}{\lambda} - 1)} \ ,
$$

then no oracle based algorithm can achieve sub-optimality smaller than β in m or less steps. However note that this is exactly the definition of fat-shattering dimension at scale $\frac{4\beta^{1/p}}{(\frac{\lambda'}{\lambda}-1)}$ (Definition 4) for the linear class $\mathrm{lin}(\mathcal{W}, \mathcal{X})$. We conclude that

$$
m(\beta, \mathcal{F}_{\mathrm{ucvx}(q,\lambda)}(\mathcal{W}, \mathcal{X})) \geq \mathrm{fat}_{\frac{4\beta^{1/p}}{(\frac{\lambda'}{\lambda}-1)}}(\mathrm{lin}(\mathcal{W}, \mathcal{X})) \ .
$$

However note that we can pick λ' as close to $\lambda_q(\mathcal{X})$ as we would like, taking the limit of $\lambda' \to \lambda_q(\mathcal{X})$ we conclude that

$$m(\beta, \mathcal{F}_{\text{ucvx}(q,\lambda)}(\mathcal{W}, \mathcal{X})) \geq \text{fat}_{\frac{4\lambda\beta^{1/p}}{(\lambda_q(\mathcal{X})-\lambda)}}(\text{lin}(\mathcal{W}, \mathcal{X})) \ .$$

□

4 Agnostic PAC Learning With Linear Predictors

In this section are interested in the problem of agnostic PAC learning Haussler (1992) with linear predictors. In this statistical learning problem, training sample $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ are drawn iid from some unknown but fixed distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. The goal is to use the training sample to output a linear predictor in $\mathcal{W}$ that minimizes expected loss over future instances drawn from the distribution $\mathcal{D}$. The loss $\ell : \mathbb{R} \times \mathcal{Y}$ we consider is assumed to be is 1-Lipschitz in its first argument for any $y \in \mathcal{Y}$ and is bounded by a constant B. We stress that we don't assume that the loss is convex. Throughout this section we also assume that $\sup_{\mathbf{w}\in\mathcal{W}} \sup_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{w}, \mathbf{x} \rangle \leq 1$ and this assumption is rather for convenience[2].

The following theorem proved in the appendix bounds the learning rate of the empirical risk minimizer in terms of the fat-shattering dimension of the linear class. The result follows by combining results from Bartlett & Mendelson (2002), Alon *et al.* (1993) and Srebro & Sridharan (2010).

Theorem 3. *Let ℓ be a 1-Lipschitz loss function bounded by B. For any $\delta > 0$, with probability at least $1 - \delta$ over draw of training sample $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n) \sim \mathcal{D}$ we have that the excess risk is bounded as :*

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell(\langle \widehat{\mathbf{w}}_n, \mathbf{x}\rangle, y)\right] - \inf_{\mathbf{w}\in\mathcal{W}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell(\langle \mathbf{w}, \mathbf{x}\rangle, y)\right]$$
$$\leq \inf_{\alpha > 1/n} \left\{ 4\alpha + \frac{12\,\log(2en^2)}{\sqrt{n}} \int_\alpha^1 \sqrt{\text{fat}_\theta(\text{lin}(\mathcal{W}, \mathcal{X}))}\, d\theta \right\} + \sqrt{\frac{B^2 \log\left(\frac{1}{\delta}\right)}{n}}$$

where $\widehat{\mathbf{w}}_n = \operatorname{argmin}_{\mathbf{w}\in\mathcal{W}} \sum_{i=1}^n \ell(\langle \mathbf{w}, \mathbf{x}_i\rangle, y_i)$ is the Empirical Risk Minimizer (ERM).

The above theorem shows that learning rate can be bounded in terms of fat shattering dimension. In fact in can be shows that the above bound is tight upto logarithmic factors when one considers the absolute loss and so one cannot hope to improve the above bound further. In the previous section we saw that the fat shattering dimension can in turn be bounded by the oracle complexity of $\mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})$. We are now in a position to combine the lower bound from the previous section with the above theorem to provide a bound on sample complexity of the learning problem we consider in terms of oracle complexity of the optimization problem over $\mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})$.

Corollary 4. *If there exists some $V > 0$ and $q \in (0, \infty)$ such that for any $\beta > 0$,*

$$m(\beta, \mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})) \leq \left(\frac{V}{\beta}\right)^q$$

then for any agnostic learning problem with 1-Lipschitz loss ℓ bounded by B, any $\beta > 0$ and $\delta > 0$, the number of samples n, need to ensure that with probability at least $1 - \delta$, the excess risk of the empirical risk minimizer, $\mathbb{E}\left[\ell(\langle \widehat{\mathbf{w}}_n, \mathbf{x}\rangle, y)\right] - \inf_{\mathbf{w}\in\mathcal{W}} \mathbb{E}\left[\ell(\langle \mathbf{w}, \mathbf{x}\rangle, y)\right]$, is smaller than β is bounded as :

[2]The bound of 1 can be replaced by any other constant with appropriate re-scaling of results.

1. *In case* $q > 2$ *:* $\quad n \leq \max\left\{ \left(\frac{448\ V\ \log\left(\frac{2eV}{\beta}\right)}{\beta} \right)^q, \left(\frac{2B}{\beta}\right)^2 \log\left(\frac{1}{\delta}\right) \right\}$

2. *In case* $q \leq 2$ *:* $\quad n \leq \max\left\{ \left(\frac{128\ V^{q/2} \log^2(\frac{2eV^q}{\beta})}{\beta} \right)^2, \left(\frac{2B}{\beta}\right)^2 \log\left(\frac{1}{\delta}\right) \right\}$

Proof. We have that for all $\beta > 0$, $\quad m(\beta, \mathcal{F}_{\mathrm{Lip}}(\mathcal{W}, \mathcal{X})) \leq \left(\frac{V}{\beta}\right)^q$. Hence by Theorem 1 we conclude that :

$$\mathrm{fat}_\beta(\mathrm{lin}(\mathcal{W}, \mathcal{X})) \leq m(\beta/2, \mathcal{F}_{\mathrm{Lip}}(\mathcal{W}, \mathcal{X})) \leq \left(\frac{2V}{\beta}\right)^q \quad .$$

Using the above with Theorem 3 we can conclude that for any $\delta > 0$, with probability at least $1 - \delta$:

$$\begin{aligned} &\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell(\langle \widehat{\mathbf{w}}_n, \mathbf{x}\rangle, y)\right] - \inf_{\mathbf{w}\in\mathcal{W}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell(\langle \widehat{\mathbf{w}}_n, \mathbf{x}\rangle, y)\right] \\ &\qquad \leq 2 \inf_{\alpha > 1/n} \left\{ 4\alpha + \frac{12(2V)^{q/2}\log(2en^2)}{\sqrt{n}} \int_\alpha^1 \theta^{-\frac{q}{2}}\, d\theta \right\} + \sqrt{\frac{B^2 \log(1/\delta)}{n}} \end{aligned} \tag{5}$$

Now to bound the first term in the RHS above we divide the analysis into two cases. First where $q \in (2, \infty)$ and next where $q \in (0, 2]$. We start for the case when $q \in (2, \infty)$. In this case,

$$\begin{aligned} 2 \inf_{\alpha > 1/n} \left\{ 4\alpha + \frac{12(2V)^{q/2}\log(2en^2)}{\sqrt{n}} \int_\alpha^1 \theta^{-\frac{q}{2}}\, d\theta \right\} &= 2 \inf_{\alpha > 1/n} \left\{ 4\alpha + \frac{12(2V)^{q/2}\log(2en^2)}{\sqrt{n}} \left[\frac{\theta^{1-\frac{q}{2}}}{1-\frac{q}{2}} \right]_\alpha^1 \right\} \\ &\leq 2 \inf_{\alpha > 1/n} \left\{ 4\alpha + \frac{24(2V)^{q/2}\log(2en^2)}{(q-2)\sqrt{n}} \frac{1}{\alpha^{\frac{q}{2}-1}} \right\} \\ &\leq \frac{224\ V\ \log(\sqrt{2e}n)}{(q-2)\, n^{1/q}} \end{aligned} \tag{6}$$

where in the last step above we used the value $\alpha = \frac{2V}{(q-2)^{2/q}\, n^{1/q}}$. Now for the case when $q \in (0, 2]$ we have,

$$\begin{aligned} 2 \inf_{\alpha > 1/n} \left\{ 4\alpha + \frac{12(2V)^{q/2}\log(2en^2)}{\sqrt{n}} \int_\alpha^1 \theta^{-\frac{q}{2}}\, d\theta \right\} &\leq 2 \inf_{\alpha > 1/n} \left\{ 4\alpha + \frac{12(2V)^{q/2}\log(2en^2)}{\sqrt{n}} \int_\alpha^1 \theta^{-1}\, d\theta \right\} \\ &\leq 2 \inf_{\alpha > 1/n} \left\{ 4\alpha + \frac{12(2V)^{q/2}\log(2en^2)}{\sqrt{n}} \log(1/\alpha) \right\} \\ &\leq 2 \left(\frac{4}{\sqrt{n}} + \frac{12(2V)^{q/2}\log^2(\sqrt{2e}n)}{\sqrt{n}} \right) \\ &\leq \frac{64\ V^{q/2}\ \log^2(\sqrt{2e}n)}{\sqrt{n}} \end{aligned} \tag{7}$$

Pugging the bounds for the two cases above back into Equation 5 we get the bounds on learning rate of ERM for the two cases in terms of sample size n. Now we need to ensure that the n is large enough so that the bound on excess risk is lesser than β. To ensure this it is enough to ensure that

$$\max\left\{ 2 \inf_{\alpha > 1/n} \left\{ 4\alpha + \frac{12(2V)^{q/2}\log(2en^2)}{\sqrt{n}} \int_\alpha^1 \theta^{-\frac{q}{2}}\, d\theta \right\}, \sqrt{\frac{B^2 \log(1/\delta)}{n}} \right\} \leq \beta/2$$

Plugging in the bound for the case when $q > 2$ from Equation 6 and for the case when $q \in (0, 2]$ from Equation 7 and solving for an n that satisfies the above inequality gives the bound in the theorem. □

Typically on the bound on sample complexity above, the second term in the max is dominated by the first. Further if we only seek bound on expected excess risk be less than β instead of the high probability version, then we are only left with the first term. So the way to parse the bound on number of samples required is as :

$$n \leq \begin{cases} \left(\frac{448\ V\ \log\left(\frac{2eV}{\beta}\right)}{\beta} \right)^q & \text{if } q > 2 \\ V^q \left(\frac{128 \log^2(\frac{2eV^q}{\beta})}{\beta} \right)^2 & \text{when } q \in (0, 2] \end{cases}$$

Notice that when $q \geq 2$, then sample complexity if upper bounded by oracle complexity to within log factors. When $q < 2$, in general the sample complexity still scales only as $1/\beta^2$ though oracle complexity is much smaller. For instance for low dimensional convex optimization problems, oracle complexity can be bounded as $d \log(1/\beta)$ (for instance using center of gravity method). However sample complexity can at best be order $1/\beta^2$ and this is unavoidable in general.

It is also interesting to note in both the cases of $q > 2$ and $q \leq 2$, the dependence on V is same as its dependence in the oracle complexity (upto log factor). Note that the constant V is problem dependent and could be dependent on dimensionality or some other structure of the problem. The result shows that the dependence on V is preserved when we bound the sample complexity using the oracle complexity bound.

In the above theorem we did not require the loss ℓ to be a convex loss function. However if it turns out that ℓ is indeed a convex loss function and the set $\mathcal{X}$ is also convex apart from being centrally symmetric, then, as we noticed in Section 2.2, the problem of optimizing empirical risk also belongs to $\mathcal{F}_{\text{Lip}}(\mathcal{W}, \mathcal{X})$ and so the bound on oracle complexity implies that we can find a β approximate solution to the empirical risk minimization problem with number of oracle calls bounded by $\left(\frac{V}{\beta}\right)^q$. Combining this with the sample complexity bound as above, we can conclude that the problem is oracle efficiently learnable with number of oracle calls bounded by $O\left(\left(\frac{V}{\beta}\right)^q\right)$ and number of samples required bounded as shown in theorem.

5 Disussion

In this paper we proved a general lower bound on oracle complexity of convex optimization problem in terms of fat-shattering dimension. Note that fat-shattering dimension is a quantity introduced and studied in the statistical learning field. In fact it is shown in Alon *et al.* (1993) that finiteness of fat-shattering dimension charecterizes learnability. While on the onset the question we were interested was in pure convex optimization. The result shows an inherent connection between learning and optimization.

In this work we mainly only dealt with lower bounding oracle complexity. For low dimensional problems (or when dimensionality is small compared to required accuracy), the cutting plane method (or for computational efficiency specifically the ellipsoid method) is effectively optimal. Also note that since fat-shattering dimension is upper bounded by dimension, for low dimensional cases effectively oracle complexity can be upper bounded in terms of fat-shattering dimension. For really large scale problems when dimensionality is larger, results from Sridharan (2010) show that for most reasonable cases (for instance in "Banach lattices"), the Mirror Descent technique, a generalization of gradient descent, is near optimal for convex optimization. In these cases it can also be shown that oracle complexity can be upper bounded using fat-shattering dimen-

sion. This opens the question/conjecture of whether one can in general provide an upper bound on oracle complexity of the optimization problem over $\mathcal{F}_{\mathrm{Lip}}(\mathcal{W}, \mathcal{X})$ in terms of the fat-shattering dimension.

Appendix

Proof of Theorem 3. By standard Rademacher complexity bound (see for instance Bartlett & Mendelson (2002)), we have that for any $\delta > 0$, with probability at least $1 - \delta$ over training sample, the excess risk of the empirical risk minimizer can be bounded in terms of the Rademacher complexity as

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell(\langle\widehat{\mathbf{w}}_n,\mathbf{x}\rangle,y)\right] - \inf_{\mathbf{w}\in\mathcal{W}}\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell(\langle\widehat{\mathbf{w}}_n,\mathbf{x}\rangle,y)\right] \leq 2\,\mathcal{R}_n(\ell\circ\lin(\mathcal{W},\mathcal{X})) + \sqrt{\frac{B^2\log(1/\delta)}{n}} \tag{8}$$

where

$$\mathcal{R}_n(\ell\circ\lin(\mathcal{W},\mathcal{X})) = \frac{1}{n}\sup_{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\in\mathcal{X}\times\mathcal{Y}}\mathbb{E}_{\epsilon_1,\ldots,\epsilon_n\sim\mathrm{Unif}\{\pm1\}}\left[\sup_{\mathbf{w}\in\mathcal{W}}\sum_{i=1}^{n}\epsilon_i\ell(\langle\mathbf{w},\mathbf{x}_i\rangle,y_i)\right]$$

where in the above $\epsilon_1,\ldots,\epsilon_n \in \{\pm1\}$ are Rademacher random variables each taking on value $+1$ with probability $1/2$ and -1 with probability $1/2$ independently. However since we assumed that the loss function ℓ is 1-Lipschitz in its first argument, using Lipschitz contraction lemma for Rademacher complexity (see for instance Theorem 12 of Bartlett & Mendelson (2002)), we have that

$$\begin{aligned}\mathcal{R}_n(\ell\circ\lin(\mathcal{W},\mathcal{X})) &\leq \mathcal{R}_n(\lin(\mathcal{W},\mathcal{X})) \\ &= \frac{1}{n}\sup_{(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_n,y_n)\in\mathcal{X}\times\mathcal{Y}}\mathbb{E}_{\epsilon_1,\ldots,\epsilon_n\sim\mathrm{Unif}\{\pm1\}}\left[\sup_{\mathbf{w}\in\mathcal{W}}\sum_{i=1}^{n}\epsilon_i\ell(\langle\mathbf{w},\mathbf{x}_i\rangle,y_i)\right]\end{aligned} \tag{9}$$

Now we can further upper bound the Rademacher complexity by the refined Dudley integral bound (see for instance Srebro & Sridharan (2010)) as :

$$\begin{aligned}\mathcal{R}_n(\lin(\mathcal{W},\mathcal{X})) &\leq \inf_{\alpha>0}\left\{4\alpha + 12\int_\alpha^1\sqrt{\frac{\log\mathcal{N}_2(\theta,\lin(\mathcal{W},\mathcal{X}))}{n}}d\theta\right\} \\ &\leq \inf_{\alpha>0}\left\{4\alpha + 12\int_\alpha^1\sqrt{\frac{\mathrm{fat}_\theta(\lin(\mathcal{W},\mathcal{X}))\ \log^2(2en/\theta)}{n}}d\theta\right\} \\ &\leq \inf_{\alpha>1/n}\left\{4\alpha + 12\int_\alpha^1\sqrt{\frac{\mathrm{fat}_\theta(\lin(\mathcal{W},\mathcal{X}))\ \log^2(2en^2)}{n}}d\theta\right\} \\ &\leq \inf_{\alpha>1/n}\left\{4\alpha + \frac{12\log(2en^2)}{\sqrt{n}}\int_\alpha^1\sqrt{\mathrm{fat}_\theta(\lin(\mathcal{W},\mathcal{X}))}d\theta\right\}\end{aligned} \tag{10}$$

Where in the above $\mathcal{N}_2(\theta,\lin(\mathcal{W},\mathcal{X}))$ is the ℓ_2 covering number of the linear function class $\lin(\mathcal{W},\mathcal{X})$ at scale θ. The second inequality above of bounding the ℓ_2 covering number in terms of fat-shattering dimension which can for instance be found for instance in Alon *et al.* (1993) (ℓ_2 covering number is bounded by ℓ_∞ covering number which and bound on that in terms of fat-shattering dimension is provided by Alon *et al.* (1993)). □

References

Alon, N., Ben-David, S., Cesa-Bianchi, N., & Haussler, D. 1993. Scale-sensitive dimensions, uniform convergence, and learnability. *Focs*, 292–301.

Bartlett, P. L., & Mendelson, S. 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. *Jmlr*, **3**, 463–482.

Boyd, S., & Vandenberghe, L. 2004. *Convex optimization*. Cambridge University Press.

Haussler, D. 1992. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and computation*, **100**(1), 78–150.

Kearns, Michael J., & Schapire, Robert E. 1990 (Oct.). Efficient distribution-free learning of probabilistic concepts. *Pages 382–391 of: 31st annual symposium on foundations of computer science*. To appear, *Journal of Computer and System Sciences*.

Nemirovski, A., & Yudin, D. 1978. *Problem complexity and method efficiency in optimization*. Nauka Publishers, Moscow.

Nemirovski, Arkadi. 2005. Efficient Methods in Convex Programming.

Srebro, N., & Sridharan, K. 2010. Note on refined dudley integral covering number bound. *In: http://ttic.uchicago.edu/ karthik/dudley.pdf*.

Srebro, Nati, Sridharan, Karthik, & Tewari, Ambuj. 2011. On the universality of online mirror descent. *Pages 2645–2653 of:* Shawe-Taylor, J., Zemel, R.S., Bartlett, P., Pereira, F.C.N., & Weinberger, K.Q. (eds), *Advances in neural information processing systems 24*.

Sridharan, Karthik. 2010. Learning from an optimization viewpoint. *In: Phd thesis*.