# Optimization for Machine Learning

Reading group, Fall 2017

November 1, 2017

## Contents

# 1 Introduction

*Speaker: Patrick Rebeschini, 12/10/2017.*

Many problems in statistics and machine learning can be formulated as the problem of computing a solution to:

$$\begin{aligned} \text{minimize} \quad & r(x) := \mathbf{E}\ell(x^T\Phi(W), Y) \\ \text{subject to} \quad & x \in \mathcal{X}, \end{aligned} \tag{1}$$

given only data in the form of $m$ i.i.d. samples $(W_1, Y_1), \dots, (W_m, Y_m) \sim (W, Y) \in \mathcal{W} \times \mathcal{Y}$, i.e., without knowledge of the distribution of $(W, Y)$ (in particular, without knowledge of the function $r$ that we want to minimize!). Here, the function $\Phi : \mathcal{W} \to \mathbb{R}^n$ (known) is a feature map, the function $x \in \mathcal{X} \subseteq \mathbb{R}^n \to x^T\Phi(W)$ is a linear predictor for the label $Y$ (the domain/constraint set $\mathcal{X}$ is known), the function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ (known) is a loss function characterizing the penalty in committing a wrong prediction, and $r(x)$ is the *test cost* or *expected risk* associated to the parameter $x$. In this setting, one wants to find $x^\star$ that yields the best linear predictor over unseen data using the data in the training set, under the assumption that the unseen data share the same (unknown) distribution of observed samples.

In binary classification, one has $\mathcal{Y} = \{-1, 1\}$ and loss functions are typically of the form $\ell(z, y) = \varphi(-zy)$ for a given $\varphi : \mathbb{R} \to \mathbb{R}$. A few examples are:

- Zero-One loss (a.k.a. the true loss): $\varphi(u) = \mathbf{1}_{u \geq 0}$.
- Exponential loss: $\varphi(u) = \exp(u)$.
- Hinge loss: $\varphi(u) = \max\{0, 1 + u\}$ (giving SVM).
- Logistic loss: $\varphi(u) = \log_2(1 + \exp(u))$.

In regression, one has $\mathcal{Y} = \mathbb{R}$, and the typical loss is:

- Least-squares loss: $\ell(u, y) = (u - y)^2$.

Perhaps the most intuitive paradigm for solving problem (1) is the *empirical risk minimization*, that is, the idea to minimize the *training cost* or *empirical risk*:

$$\begin{aligned} \text{minimize} \quad & R(x) := \frac{1}{m}\sum_{i=1}^{m} R_i(x) \\ \text{subject to} \quad & x \in \mathcal{X}, \end{aligned} \tag{2}$$

where $R_i(x) := \ell(x^T\Phi(W_i), Y_i)$. This approach is motivated by the fact that for each $x \in \mathcal{X}$ the law of large number gives $r(x) \approx R(x)$ so that one might expect that $x^\star \approx X^\star$, where $X^\star \in \arg\min_{x \in \mathcal{X}} R(x)$. This intuition can be made precise, and Proposition 1 below shows that the estimation error $r(X^\star) - r(x^\star) \geq 0$ is controlled by suprema of random processes.

**Proposition 1.** *We have*

$$r(X^\star) - r(x^\star) \leq \sup_{x \in \mathcal{X}}(r(x) - R(x)) + \sup_{x \in \mathcal{X}}(R(x) - r(x)).$$

*Proof.* By adding and subtracting terms, we get

$$r(X^\star) - r(x^\star) = r(X^\star) - R(X^\star) + R(X^\star) - R(x^\star) + R(x^\star) - r(x^\star).$$

As $X^\star$ is a minimizer of $R$, we have $R(X^\star) - R(x^\star) \leq 0$. The proof follows by taking the supremum over $x \in \mathcal{X}$. □

This analysis assumes that one can actually compute an exact minimizer $X^\star$ for the training cost $R$. With a finite amount of computational resources available, however, typically one can only compute an approximate minimizer. Let $\hat{X}_t \in \mathcal{X}$ denote an approximate solution to $\arg\min_{x\in\mathcal{X}} R(x)$ that is computed after $t$ iterations of a given algorithmic procedure. Proceeding as above we have the following result.

**Proposition 2.** *We have*

$$r(\hat{X}_t) - r(x^\star) \leq \underbrace{\sup_{x\in\mathcal{X}}(r(x) - R(x)) + \sup_{x\in\mathcal{X}}(R(x) - r(x))}_{STATISTICS} + \underbrace{R(\hat{X}_t) - R(X^\star)}_{OPTIMIZATION}. \tag{3}$$

*Proof.* By adding and subtracting terms, we get

$$r(\hat{X}_t) - r(x^\star) = r(\hat{X}_t) - R(\hat{X}_t) + R(\hat{X}_t) - R(X^\star) + R(X^\star) - R(x^\star) + R(x^\star) - r(x^\star).$$

As $X^\star$ is a minimizer of $R$, we have $R(X^\star) - R(x^\star) \leq 0$. The proof follows by taking the supremum over $x \in \mathcal{X}$. □

The problem is now to estimate how close the expected risk of the approximate empirical risk minimizer $r(\hat{X}_t)$ is to the minimal expected risk $r(x^\star)$ in terms of the number of training samples $m$, the dimension of the feature map $n$, the number of iterates $t$ for the optimization routine, and the other parameters that define the model at hand (for example, the radius of the parameter set $\mathcal{X}$, properties of the loss function $\ell$, and of the feature map $\Phi$). To this end, as Proposition 2 shows, we need to control two random terms: the *STATISTICS* term $\sup_{x\in\mathcal{X}}(r(x) - R(x)) + \sup_{x\in\mathcal{X}}(R(x) - r(x))$ and the *OPTIMIZATION* term $R(\hat{X}_t) - R(X^\star)$.

For the most part this reading group will focus on algorithms that can be used to approximately minimize the empirical risk and hence control the *OPTIMIZATION* term in (3), making various assumptions for the loss function $\ell$ and the parameter space $\mathcal{X}$, starting from the fruitful assumption of convexity.[1] Along with convexity, we state some of the main properties that a generic function $f$ can have that are used to assess the quality of algorithms to minimize $f$. Here we assume that $f$ is differentiable, but analogous notions can be defined for non-differentiable functions using the notion of subgradients. We also mention in brackets equivalent (local) definitions, which apart from $L$-Lipschitz hold when $f$ is twice-differentiable.

- Convexity: $f(x) - f(y) \leq \nabla f(x)^T(x - y)$ for any $x, y \in \mathbb{R}^n$ ($\nabla^2 f(x) \succcurlyeq 0$ for any $x \in \mathbb{R}^n$).
- $L$-Lipschitz: $|f(x) - f(y)| \leq L\|x - y\|_2$ for any $x, y \in \mathbb{R}^n$ ($\|\nabla f(x)\|_2 \leq L$ for any $x \in \mathbb{R}^n$).
- $\beta$-Smoothness: $\|\nabla f(x) - \nabla f(y)\|_2 \leq \beta\|x - y\|_2$ for any $x, y \in \mathbb{R}^n$ ($\nabla^2 f(x) \preccurlyeq \beta I$ for any $x \in \mathbb{R}^n$).
- $\alpha$-Strong convexity: $f(x) - f(y) \leq \nabla f(x)^T(x - y) - \frac{\alpha}{2}\|x - y\|_2^2$ for any $x, y \in \mathbb{R}^n$ ($\nabla^2 f(x) \succcurlyeq \alpha I$ for any $x \in \mathbb{R}^n$).

Going back to the losses defined above, it is easy to check that the convex losses below satisfy the following:

- Exponential loss: $\varphi(u) = \exp(u)$: Lipschitz NO, Smooth NO, strongly-convex NO.
- Hinge loss: $\varphi(u) = \max\{0, 1 + u\}$: Lipschitz YES, Smooth NO, strongly-convex NO.
- Logistic loss: $\varphi(u) = \log_2(1 + \exp(u))$: Lipschitz YES, Smooth YES, strongly-convex NO.
- Least-squares loss: $\ell(u, y) = (u - y)^2$: Lipschitz NO, Smooth YES, strongly-convex YES.

[1] Convexity is only needed to perform the analysis of the algorithms that we introduce, but it is not needed to run the algorithms themselves; in fact, in practice the algorithms that we define are run also on non-convex problems.

Note that as far as the empirical risk minimization goes, what we care about are the (almost sure) properties of the function $x \in \mathcal{X} \to R(x) := \frac{1}{m}\sum_{i=1}^m R_i(x) = \frac{1}{m}\sum_{i=1}^m \ell(x^T\Phi(W_i), Y_i)$. In particular, if $\mathcal{X}$ and $\Phi$ are bounded, then $x^T\Phi(W_i)$ is bounded and we can restrict the definitions above to hold on a bounded set. Henceforth, we say that a function $x \in \mathrm{Dom}(f) \subseteq \mathbb{R}^n \to f(x)$ is $L$-Lipschitz if $|f(x) - f(y)| \leq L\|x - y\|_2$ for any $x, y \in \mathrm{Dom}(f)$ ($\|\nabla f(x)\|_2 \leq L$ for any $x \in \mathrm{Dom}(f)$). Assuming boundedness provides extra flexibility, allowing one to directly use these properties in setting where otherwise they would not apply (take the exponential loss, for instance, which is only Lipschitz on a bounded interval).

Even if we assume boundedness, assuming that the empirical risk $R$ is strongly convex is typically a strong assumption. In fact, note that *for any* $x \in \mathcal{X}$ (that is, regardless of the properties of $\mathcal{X}$) the Hessian

$$\nabla^2 R(x) = \frac{1}{m}\sum_{i=1}^m \frac{\partial^2}{\partial z^2}\ell(x^T\Phi(W_i), Y_i)\,\Phi(W_i)\Phi(W_i)^T \in \mathbb{R}^{n\times n}$$

is not invertible if $m < n$, being a sum of $m < n$ rank-1 matrices. So in this case $x \to \nabla^2 R(x)$ can not be strongly convex. In applications where $m \geq n$, the empirical covariance can be invertible, but it is typically the case that the strong convexity parameter $\alpha$ is very small, of order $1/m$, effectively $\alpha \approx 0$ (or close to machine precision) for applications involving large datasets. For this reason, in this reading group we will only focus on results about Lipschitz and smooth functions, which are more natural assumptions for the empirical risk $R$, as we now show.

First, note that if we assume that $\ell$ is $L_{\text{loss}}$-Lipschitz in the first coordinate, namely, $z \to \ell(z, y)$ is $L_\ell$-Lipschitz for any $y \in \mathcal{Y}$, and that the feature map $\Phi$ is bounded in $\ell_2$, namely, $\|\Phi\|_2 \leq G$, then $R$ is $GL_\ell$-Lipschitz:

$$|R(x) - R(y)| \leq \frac{1}{m}\sum_{i=1}^m |\ell(x^T\Phi(W_i), Y_i) - \ell(y^T\Phi(W_i), Y_i)| \leq \frac{L_\ell}{m}\sum_{i=1}^m \|(x-y)^T\Phi(W_i))\|_2 \leq GL_\ell\|x - y\|_2,$$

where for the last inequality we used Cauchy-Schwarz. Second, note that if we assume that $\ell$ is $\beta_\ell$-Lipschitz in the first coordinate, namely, $z \to \ell(z, y)$ is $\beta_{\text{loss}}$-Lipschitz for any $y \in \mathcal{Y}$, and that $\|\Phi\|_2 \leq G$, then $R$ is $G^2\beta_{\text{loss}}$-smooth:

$$\begin{aligned}\|\nabla R(x) - \nabla R(y)\|_2 &\leq \frac{1}{m}\sum_{i=1}^m \left\| \left( \frac{\partial}{\partial z}\ell(x^T\Phi(W_i), Y_i) - \frac{\partial}{\partial z}\ell(y^T\Phi(W_i), Y_i) \right)\Phi(W_i) \right\|_2 \\ &\leq \frac{1}{m}\sum_{i=1}^m \beta_\ell |(x-y)^T\Phi(W_i)| \|\Phi(W_i)\|_2 \leq G^2\beta_\ell\|x - y\|_2.\end{aligned}$$

Before we embark on the adventure of minimizing the empirical risk and bound the *OPTIMIZATION* term in (3), we should try to develop some basic understanding of what to expect from the behavior of the *STATISTICS* term as well. This understanding will give us insights on the type of precision that we should strive for to control the *OPTIMIZATION* term. To this end, we now give a brief detour on statistical learning theory and derive an explicit bound in the case when the loss function $\ell$ is Lipschitz (no assumption on convexity is made for this result), the feature map is bounded, and also the parameter space $\mathcal{X}$ is bounded. The proof is instructive and relies on classical machinery of concentration inequalities and Rademacher complexity.

## 1.1 Statistics term

Statistical learning theory focuses on understanding how the estimation error $r(X^\star) - r(x^\star)$ converges to zero as a function of the sample size $m$, the loss function $\ell$, the parameter space $\mathcal{X}$, and possibly some properties of the distribution of $\Phi(W)$ and $Y$ (recall that the distribution of the data is not known). We will prove the following results, inspired by the presentation in [2].

**Proposition 3** (Mean)**.** *Let $z \to \ell(z, y)$ be $L_\ell$-Lipschitz for any $y \in \mathcal{Y}$, $\sup_{x\in\mathcal{X}}\|x\|_2 \leq B$ and $\|\Phi(W)\|_2 \leq G$ a.s.. Then,*

$$\mathbf{E}[\mathit{STATISTICS}] = \mathbf{E}\sup_{x\in\mathcal{X}}(r(x) - R(x)) + \mathbf{E}\sup_{x\in\mathcal{X}}(R(x) - r(x)) \leq 4\frac{BGL_\ell}{\sqrt{m}}.$$

**Proposition 4** (High probability)**.** *Let $z \to \ell(z, y)$ be $L_\ell$-Lipschitz for any $y \in \mathcal{Y}$, $\sup_{x\in\mathcal{X}} \|x\|_2 \le B$ and $\|\Phi(W)\|_2 \le G$ a.s.. Then, with probability at least $1-\delta$ we have*

$$STATISTICS = \sup_{x\in\mathcal{X}}(r(x) - R(x)) + \sup_{x\in\mathcal{X}}(R(x) - r(x)) \le 2\frac{\ell_0 + BGL_\ell}{\sqrt{m}}\left(2 + \sqrt{2\log\frac{1}{\delta}}\right),$$

*where $\ell_0 := \sup_{y\in\mathcal{Y}} |\ell(0, y)|$.*

The quality of the bound in Proposition 4 is typical in statistical learning theory. In particular, it is typically the case that the *slow* rate $O(1/\sqrt{m})$ can not be improved unless other assumptions are taken into consideration that allow to prove the *fast* rate $O(1/m)$. These upper bounds suggest[2] that we only need to approximate the *OPTIMIZATION* term up to precision $O(1/\sqrt{m})$ (resp. $O(1/m)$).

### 1.1.1 Bound on the mean

We prove Proposition 3. An important tool to bound the mean of a supremum of a random process is the Rademacher complexity. The Rademacher complexity of a set is defined as follows.

**Definition 1.** *The Rademacher complexity of a set $T \subseteq \mathbb{R}^m$ is*

$$\mathcal{R}(T) := \mathbf{E}\sup_{t\in T}\frac{1}{m}\sum_{i=1}^m \epsilon_i t_i,$$

*where $\epsilon_1, \dots, \epsilon_m$ are i.i.d. random variables uniform in $\{-1, 1\}$.*

The quantity $\mathcal{R}(T)$ is a measure of how complex the set $T$ is, as $\sup_{t\in T}\sum_{i=1}^m \epsilon_i t_i$ describes how well elements in $T$ can replicate the sign pattern of a random signal $\epsilon = (\epsilon_1, \dots, \epsilon_m)$. One way of seeing this is to restrict to the case $T \subseteq [-1, 1]^n$. If $T = [-1, 1]^n$ then $\mathcal{R}(T) = 1$, as for any realization of the signal $\epsilon$ we can find $t \in T$ that has its same sign pattern. More generally, if $T \subseteq [-1, 1]^n$ with $k$-sparse components, then $\mathcal{R}(T) = k/m$.

One reason why the Rademacher complexity is a useful notion is given by the following lemma, that shows how this quantity behaves with respect to composition with Lipschitz functions.

**Lemma 1** (Contraction property)**.** *For each $i \in \{1, \dots, m\}$, let $\varphi_i : \mathbb{R} \to \mathbb{R}$ be a $L$-Lipschitz function. Let $(\varphi_1, \dots, \varphi_m) \circ T = \{(\varphi_1(t_1), \dots, \varphi_m(t_m))^T : t \in T\}$. Then,*

$$\mathcal{R}((\varphi_1, \dots, \varphi_m) \circ T) \le L\mathcal{R}(T),$$

*or, explicitly,*

$$\mathbf{E}\sup_{t\in T}\sum_{i=1}^m \epsilon_i\varphi_i(t_i) \le L\mathbf{E}\sup_{t\in T}\sum_{i=1}^m \epsilon_i t_i.$$

*Proof.* First, note that for $S \subseteq \mathbb{R}^2$ and $\varphi : \mathbb{R} \to \mathbb{R}$ 1-Lipschitz we have

$$\begin{aligned}
\sup_{s\in S}(s_1 + \varphi(s_2)) + \sup_{s\in S}(s_1 - \varphi(s_2)) &= \sup_{s,s'\in S}(s_1 + s_1' + \varphi(s_2) - \varphi(s_2')) = \sup_{s,s'\in S}(s_1 + s_1' + |\varphi(s_2) - \varphi(s_2')|) \\
&\le \sup_{s,s'\in S}(s_1 + s_1' + |s_2 - s_2'|) = \sup_{s,s'\in S}(s_1 + s_1' + s_2 - s_2') \\
&= \sup_{s\in S}(s_1 + s_2) + \sup_{s\in S}(s_1 - s_2).
\end{aligned}$$

[2]Caveats are in order: after all the result in (3) is only an upper-bound, and it is often the case that in practice one does not know the constants sitting in front of the error bounds that one can prove for both the *OPTIMIZATION* term and the *STATISTICS* term, i.e., the constants $G, L, R$ in our case.

Then, if the functions $\varphi_i$'s are 1-Lipschitz we have, for any $k \in \{1, \dots, m\}$,

$$\begin{aligned}
\mathbf{E} \sup_{t\in T} \sum_{i=1}^m \epsilon_i \varphi_i(t_i) &= \mathbf{E}\mathbf{E}\left[ \sup_{t\in T} \sum_{i=1}^m \epsilon_i \varphi_i(t_i) \middle| \epsilon_1, \dots, \epsilon_{i-1}, \epsilon_{i+1}, \dots, \epsilon_m \right] \\
&= \frac{1}{2}\mathbf{E}\left[ \sup_{t\in T} \left( \sum_{i\neq k} \epsilon_i \varphi_i(t_i) + \varphi_k(t_k) \right) + \sup_{t\in T} \left( \sum_{i\neq k} \epsilon_i \varphi_i(t_i) - \varphi_k(t_k) \right) \right] \\
&\leq \frac{1}{2}\mathbf{E}\left[ \sup_{t\in T} \left( \sum_{i\neq k} \epsilon_i \varphi_i(t_i) + t_k \right) + \sup_{t\in T} \left( \sum_{i\neq k} \epsilon_i \varphi_i(t_i) - t_k \right) \right] \\
&= \mathbf{E} \sup_{t\in T} \left( \sum_{i\neq k} \epsilon_i \varphi_i(t_i) + \epsilon_k t_k \right),
\end{aligned}$$

and by iterating one finds

$$\mathbf{E} \sup_{t\in T} \sum_{i=1}^m \epsilon_i \varphi_i(t_i) \leq \mathbf{E} \sup_{t\in T} \sum_{i=1}^m \epsilon_i t_i.$$

If the functions $\varphi_i$'s are $L$-Lipschitz, then clearly,

$$\mathbf{E} \sup_{t\in T} \sum_{i=1}^m \epsilon_i \varphi_i(t_i) = L\mathbf{E} \sup_{t\in T} \sum_{i=1}^m \epsilon_i \frac{\varphi_i(t_i)}{L} \leq L\mathbf{E} \sup_{t\in T} \sum_{i=1}^m \epsilon_i t_i.$$

□

Given the contraction property for the Rademacher complexity of a set, we can derive the following result.

**Proposition 5.** *Let $z \to \ell(z, y)$ be $L_\ell$-Lipschitz for any $y \in \mathcal{Y}$. Then,*

$$\mathbf{E} \sup_{x\in\mathcal{X}} (r(x) - R(x)) \leq 2\mathbf{E} \sup_{x\in\mathcal{X}} \frac{1}{m} \sum_{i=1}^m \epsilon_i R_i(x) \leq 2L_\ell \mathbf{E} \sup_{x\in\mathcal{X}} \frac{1}{m} \sum_{i=1}^m \epsilon_i x^T \Phi(W_i),$$

*where $\epsilon_1, \dots, \epsilon_n$ are i.i.d. random variables uniform in $\{-1, 1\}$, independent of $(W_1, Y_1), \dots, (W_m, Y_m)$.*

*Proof.* The proof uses the standard machinery of symmetrization. Let $Z_i = (W_i, Y_i)$ for any $i \in \{1, \dots, m\}$. Let $Z'_1, \dots, Z'_m$ be an independent copy of the data $Z_1, \dots, Z_m$, and let $R'_i(x) := \ell(x^T \Phi(W'_i), Y'_i)$ for each $i \in \{1, \dots, m\}$. Using that $r(x) = \mathbf{E}R'_i(x) = \mathbf{E}[R'_i(x)|Z_1, \dots, Z_m]$, we have

$$\begin{aligned}
\mathbf{E} \sup_{x\in\mathcal{X}} \left( r(x) - \frac{1}{m} \sum_{i=1}^m R_i(x) \right) &= \mathbf{E} \sup_{x\in\mathcal{X}} \frac{1}{m} \sum_{i=1}^m \mathbf{E}[R'_i(x) - R_i(x))|Z_1, \dots, Z_m] \\
&\leq \mathbf{E}\mathbf{E}\left[ \sup_{x\in\mathcal{X}} \frac{1}{m} \sum_{i=1}^m (R'_i(x) - R_i(x)) \middle| Z_1, \dots, Z_m \right] \\
&= \mathbf{E} \sup_{x\in\mathcal{X}} \frac{1}{m} \sum_{i=1}^m (R'_i(x) - R_i(x)) \\
&= \mathbf{E} \sup_{x\in\mathcal{X}} \frac{1}{m} \sum_{i=1}^m \epsilon_i (R'_i(x) - R_i(x)) \quad \text{by symmetrization} \\
&\leq 2\mathbf{E} \sup_{x\in\mathcal{X}} \frac{1}{m} \sum_{i=1}^m \epsilon_i R_i(x) = 2\mathbf{E} \sup_{x\in\mathcal{X}} \frac{1}{m} \sum_{i=1}^m \epsilon_i \ell(x^T \Phi(W_i), Y_i).
\end{aligned}$$

Conditioning on $Z_1, \ldots, Z_m$ we can apply the contraction property for the Rademacher complexity of a set, applying Lemma 1 with the choice $\varphi_i(z) = \ell(z, Y_i)$ and $T = \{(x^T\Phi(W_1), \ldots, x^T\Phi(W_m))^T : x \in \mathcal{X}\}$:

$$\mathbf{E}\left[\sup_{x\in\mathcal{X}} \frac{1}{m}\sum_{i=1}^m \epsilon_i \ell(x^T\Phi(W_i), Y_i)\middle| Z_1, \ldots, Z_m\right] \leq L_\ell \mathbf{E}\left[\sup_{x\in\mathcal{X}} \frac{1}{m}\sum_{i=1}^m \epsilon_i x^T\Phi(W_i)\middle| Z_1, \ldots, Z_m\right], \tag{4}$$

and the result follows by taking the expectation. □

**Remark 1** (Emipirical Rademacher complexity)**.** *The quantity*

$$\mathbf{E}\sup_{x\in\mathcal{X}} \frac{1}{m}\sum_{i=1}^m \epsilon_i x^T\Phi(W_i)$$

*on the right hand side of the bound in Proposition 5 can be interpreted as the expected value of the Rademacher complexity of a random set. In fact, by conditioning on the data $W_1, \ldots, W_m$ we get*

$$\mathbf{E}\sup_{x\in\mathcal{X}} \frac{1}{m}\sum_{i=1}^m \epsilon_i x^T\Phi(W_i) = \mathbf{E}\mathbf{E}\left[\sup_{x\in\mathcal{X}} \frac{1}{m}\sum_{i=1}^m \epsilon_i x^T\Phi(W_i)\middle| W_1, \ldots, W_m\right] = \mathbf{E}\mathbf{E}\left[\sup_{t\in T_{W_1,\ldots,W_m}} \frac{1}{m}\sum_{i=1}^m \epsilon_i t_i\middle| W_1, \ldots, W_m\right],$$

*where $T_{W_1,\ldots,W_m} := \{(x^T\Phi(W_1), \ldots, x^T\Phi(W_m))^T : x \in \mathcal{X}\}$. The Rademacher complexity that we get when we condition on the data, as at the end of the proof of Proposition 5, is typically referred to as the* empirical *Rademacher complexity.*

We now present a result to control the Rademacher complexity when we have boundedness assumptions with respect to the Euclidean norm.

**Proposition 6.** *Let $\sup_{x\in\mathcal{X}} \|x\|_2 \leq B$ and $\|\Phi(W)\|_2 \leq G$ a.s.. Then,*

$$\mathbf{E}\sup_{x\in\mathcal{X}} \frac{1}{m}\sum_{i=1}^m \epsilon_i x^T\Phi(W_i) \leq \frac{BG}{\sqrt{m}}.$$

*Proof.* We have

$$\begin{aligned}
\mathbf{E}\sup_{x\in\mathcal{X}} \frac{1}{m}\sum_{i=1}^m \epsilon_i x^T\Phi(W_i) &\leq \sup_{x\in\mathcal{X}} \|x\|_2 \mathbf{E}\left\|\frac{1}{m}\sum_{i=1}^m \epsilon_i\Phi(W_i)\right\|_2 \quad \text{by Cauchy-Schwarz} \\
&\leq B\mathbf{E}\sqrt{\left\|\frac{1}{m}\sum_{i=1}^m \epsilon_i\Phi(W_i)\right\|_2^2} \leq B\sqrt{\mathbf{E}\left\|\frac{1}{m}\sum_{i=1}^m \epsilon_i\Phi(W_i)\right\|_2^2} \quad \text{by Jensen's inequality} \\
&= \frac{B}{m}\sqrt{\mathbf{E}\sum_{j=1}^n \left(\sum_{i=1}^m \epsilon_i\Phi(W_i)_j\right)^2} = \frac{B}{m}\sqrt{\mathbf{E}\sum_{j=1}^n\sum_{i=1}^m (\epsilon_i\Phi(W_i)_j)^2} \quad \text{by the independence of the } \epsilon\text{'s} \\
&= \frac{B}{m}\sqrt{\mathbf{E}\sum_{i=1}^m \|\Phi(W_i)\|_2^2} \leq \frac{GB}{\sqrt{m}} \quad \text{as } \|\Phi(W)\|_2 \leq G \text{ a.s.}
\end{aligned}$$

□

Proposition 5 and Proposition 6 immediately yields the following result, from which Proposition 3 follows immediately.

**Proposition 7.** *Let $z \to \ell(z, y)$ be $L_\ell$-Lipschitz for any $y \in \mathcal{Y}$, $\sup_{x\in\mathcal{X}} \|x\|_2 \leq B$ and $\|\Phi(W)\|_2 \leq G$ a.s.. Then,*

$$\mathbf{E}\sup_{x\in\mathcal{X}}(r(x) - R(x)) \leq 2\frac{BGL_\ell}{\sqrt{m}}.$$

### 1.1.2 Bound with high probability

We prove Proposition 4. Let us consider the following quantity:

$$h(Z_1, \dots, Z_m) := \sup_{x \in \mathcal{X}} (r(x) - R(x)), \tag{5}$$

where we use the notation $Z_i := (W_i, Y_i)$. Proposition 7 yields a bound on $\mathbf{E}h(Z_1, \dots, Z_m)$. A classical way to derive bounds in high probability is to use concentration inequalities, which are tools in probability theory to bound the deviation of a function of many i.i.d. random variables from its mean. One of the classical concentration inequalities is the Bounded Difference, of which we now state a simple version.

**Proposition 8** (Bounded Difference Inequality)**.** *Let $Z_1, \dots, Z_m \in \mathcal{Z}$ be $m$ i.i.d. random variables, and let $h : \mathcal{Z}^m \to \mathbb{R}$ be a function satisfying, for every $k \in \{1, \dots, m\}$ and for all $z_1, \dots, z_m, z_k' \in \mathcal{Z}$:*

$$|h(z_1, \dots, z_{k-1}, z_k, z_{k+1}, \dots, z_m) - h(z_1, \dots, z_{k-1}, z_k', z_{k+1}, \dots, z_m)| \le c.$$

*Then,*

$$\mathbf{P}(h(Z_1, \dots, Z_m) \ge \mathbf{E}h(Z_1, \dots, Z_m) + t) \le \exp\bigg( - \frac{2t^2}{mc^2} \bigg),$$

*or, equivalently, with probability at least $1 - \delta$ we have*

$$h(Z_1, \dots, Z_m) \le \mathbf{E}h(Z_1, \dots, Z_m) + c\sqrt{\frac{m}{2} \log \frac{1}{\delta}}.$$

To apply the Bounded Difference inequality to the function $h$ defined in (5), we need to find $c$ that satisfy the assumption of Proposition 8 to control the magnitude of the difference in the function $h$ upon changing only one coordinate. The next proposition shows how to do this we have boundedness assumptions with respect to the Euclidean norm.

**Proposition 9.** *Let $h$ be the function defined in* (5)*. Let $z \to \ell(z, y)$ be $L_\ell$-Lipschitz for any $y \in \mathcal{Y}$, $\sup_{x \in \mathcal{X}} \|x\|_2 \le B$ and $\|\Phi(W)\|_2 \le G$ a.s.. Then, $c \in \mathbb{R}$ satisfying the requirement of the Bounded Difference inequality is given by*

$$c = \frac{2}{m} \bigg( \ell_0 + BGL_\ell \bigg),$$

*where $\ell_0 := \sup_{y \in \mathcal{Y}} |\ell(0, y)|$.*

*Proof.* Fix $k \in \{1, \dots, m\}$ and let $z = (z_1, \dots, z_{k-1}, z_k, z_{k+1}, \dots, z_m)$ and $z' = (z_1, \dots, z_{k-1}, z_k', z_{k+1}, \dots, z_m)$. Then,

$$|h(z) - h(z')| = \bigg| \sup_{x \in \mathcal{X}} \bigg( r(x) - \frac{1}{m} \sum_{i=1}^m R_i^z(x) \bigg) - \sup_{x \in \mathcal{X}} \bigg( r(x) - \frac{1}{m} \sum_{i=1}^m R_i^{z'}(x) \bigg) \bigg|,$$

where $R_i^z(x) := \ell(x^T \Phi(w_i), y_i)$. If $h(z) - h(z') \ge 0$ and we let $\tilde{x} \in \mathcal{X}$ be the maximizer of $\sup_{x \in \mathcal{X}} (r(x) - \frac{1}{m} \sum_{i=1}^m R_i^z(x))$ (note that the supremum is attained by the Extreme Value Theorem), we have

$$\begin{aligned} h(z) - h(z') &= \bigg( r(\tilde{x}) - \frac{1}{m} \sum_{i=1}^m R_i^z(\tilde{x}) \bigg) - \sup_{x \in \mathcal{X}} \bigg( r(x) - \frac{1}{m} \sum_{i=1}^m R_i^{z'}(x) \bigg) \\ &\le \bigg( r(\tilde{x}) - \frac{1}{m} \sum_{i=1}^m R_i^z(\tilde{x}) \bigg) - \bigg( r(\tilde{x}) - \frac{1}{m} \sum_{i=1}^m R_i^{z'}(\tilde{x}) \bigg) \\ &= \frac{1}{m} (R_k^z(\tilde{x}) - R_k^{z'}(\tilde{x})) \le \frac{2}{m} \sup_{x \in \mathcal{X}, z \in \mathcal{Z}} |R_k^z(x)|. \end{aligned}$$

Proceeding analogously in the case $h(z) - h(z') \leq 0$, we finally find

$$|h(z) - h(z')| \leq \frac{2}{m} \sup_{x \in \mathcal{X}, z \in \mathcal{Z}} |R_k^z(x)| = \frac{2}{m} \sup_{x \in \mathcal{X}, w \in \mathcal{W}, y \in \mathcal{Y}} |\ell(x^T \Phi(w), y)|.$$

Using, that $z \to \ell(z, y)$ is $L_\ell$-Lipschitz for any $y \in \mathcal{Y}$ and Cauchy-Schwarz we get

$$|\ell(x^T \Phi(w), y)| \leq |\ell(0, y)| + |\ell(x^T \Phi(w), y) - \ell(0, y)| \leq |\ell(0, y)| + L_\ell \|x^T \Phi(w)\|_2 \leq |\ell(0, y)| + L_\ell \|x\|_2 \|\Phi(w)\|_2.$$

As $\|\Phi(W)\|_2 \leq G$ a.s. and $\sup_{x \in \mathcal{X}} \|x\|_2 \leq B$ we have $|\ell(x^T \Phi(w), y)| \leq |\ell(0, y)| + BGL_\ell$, so that

$$|h(z) - h(z')| \leq \frac{2}{m} \left( \sup_{y \in \mathcal{Y}} |\ell(0, y)| + BGL_\ell \right).$$

□

The proof of Proposition 4 follows by putting everything together, namely, Proposition 8, Proposition 9, and Proposition 7, noting that the bounds in high probability that we have obtained hold *simultaneously* for $\sup_{x \in \mathcal{X}}(r(x) - R(x))$ and $\sup_{x \in \mathcal{X}}(R(x) - r(x))$. Here are the details.

*Proof of Proposition 4.* Using, respectively, Proposition 8 and Proposition 7, we have that with probability at least $1 - \delta$ the following holds:

$$\sup_{x \in \mathcal{X}} (r(x) - R(x)) \leq \mathbf{E}\left[ \sup_{x \in \mathcal{X}} (r(x) - R(x)) \right] + c\sqrt{\frac{m}{2} \log \frac{1}{\delta}} \leq \frac{2BGL_\ell}{\sqrt{m}} + c\sqrt{\frac{m}{2} \log \frac{1}{\delta}},$$

with $c = \frac{2}{m}(\ell_0 + BGL_\ell)$ by Proposition 9, which yields

$$\mathbf{P}\left( \sup_{x \in \mathcal{X}} (r(x) - R(x)) \leq c' \right) \geq 1 - \delta,$$

where $c' := (\ell_0 + BGL_\ell)(2 + \sqrt{2 \log(1/\delta)})/\sqrt{m}$. As the bounds we have derived holds also for $\sup_{x \in \mathcal{X}}(R(x) - r(x))$, we have

$$\mathbf{P}\left( \sup_{x \in \mathcal{X}} (r(x) - R(x)) + \sup_{x \in \mathcal{X}} (R(x) - r(x)) \leq 2c' \right) \geq \mathbf{P}\left( \left\{ \sup_{x \in \mathcal{X}} (r(x) - R(x)) \leq c' \right\} \bigcap \left\{ \sup_{x \in \mathcal{X}} (R(x) - r(x)) \leq c' \right\} \right) \geq 1 - \delta.$$

□

## 1.2 Overview of the reading group

Given the analysis of the previous section, we can now describe the plan for the reading group this term. While we motivated our interest in optimization techniques by looking at the classical setting of supervised machine learning and empirical risk minimization, only in Week 8 (hopefully!) we will see how to develop specialized algorithms to minimize $R$ (that is, algorithms that can take advantage of the specific structure of $R$). In fact, most of our time will be devoted to reviewing classical algorithms to optimize a *general* convex function $f$ over a convex set $\mathcal{X}$, namely, to solve:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{X}. \end{aligned}$$

Below is the current plan of what we aim to cover, along with references to the corresponding literature.

**IMPORTANT NOTE FOR SPEAKERS.**
The results that are quoted in bold are the ones we will be focusing on. When presenting your section and typing up notes, focus only on covering these results, along with introducing the quantities/proofs the are needed. You might comment on the other results if you have time, or just to give an overview of the bigger picture. In other words, instead of trying to cover as much as possible content-wise, try to cover as little as possible with as much explanation as possible (pictures, intuition, etc.).
As we move along in the reading group, the plan below might change. Please make sure that you have the latest version of these lecture notes before you prepare your section.

**Week 2: Convexity. Black-box model. Projected gradient descent methods.**

Readings: Parts of Chapter 1 and Chapter 3 in [3]. Details are below.

We introduce convexity, and some of its basic properties, such as the existence of subgradients (**Proposition 1.1 in [3]**) and the first order optimality condition (**Proposition 1.3 in [3]**).

We introduce the black-box model of computation, where we assume that the constraint set $\mathcal{X} \subseteq \mathbb{R}^n$ is known and the objective function $f$ is unknown but can be accessed through queries to first order oracles: given $x \in \mathcal{X}$, a first order oracle yields back a subgradient of $f$ at $x$.

We study how the different assumptions of $L$-Lipschitz, $\beta$-Smoothness, and $\alpha$-Strong convexity, lead to different convergence rates for projected subgradient descent methods. Upon different conditions, different projected subgradient descent methods achieve the following rates:

| | $L$-Lipschitz | $\beta$-smooth |
|---|---|---|
| Convex | $O(BL/\sqrt{t})$ | $O(B^2\beta/t)$ |
| $\alpha$-strongly convex | $O(L^2/(\alpha t))$ | $O(B^2 e^{-t\alpha/\beta})$ |

where $B$ is the radius of the Euclidean ball that contains $\mathcal{X}$, namely, $\sup_{x\in\mathcal{X}} \|x\|_2 \leq B$. As we discuss earlier, for applications in machine learning the assumption of $\alpha$-strong convexity is typically too strong. This is why we only prove the rates for the convex case, both for $L$-Lipschitz and for $\beta$-Smooth. For completeness, here is where to find all the results mentioned above:

- Convex & $L$-Lipschitz: $O(BL/\sqrt{t})$ (**Theorem 3.2 in [3]**).
- Convex & $\beta$-Smoothness: $O(B^2\beta/t)$ (**Theorem 3.7 and in [3]**).
- $\alpha$-Strong convexity & $L$-Lipschitz: $O(L^2/(\alpha t))$ (Theorem 3.9 and in [3]).
- $\alpha$-Strong convexity & $\beta$-Smoothness: $O(B^2 \exp(-t\alpha/\beta))$ (Theorem 3.10 and in [3]).

These results are sometimes called *dimension-free*, since as presented the rates do not depend explicitly on the ambient dimension $n$. However, the dependence on the dimension enters implicit in the constants: note that $B$ is the radius of the constraint set $\mathcal{X} \subseteq \mathbb{R}^n$, and that the parameters $L$, $\alpha$, and $\beta$, are defined in terms of the Euclidean norm in $\mathbb{R}^n$. The terminology dimension-free is used in [3] to differentiate the rates achieved by subgradient descent methods from the rates obtained by ellipsoid methods (Chapter 2 in [3]), which we will not cover in this reading group.

We can also phrase these results in terms of *oracle complexity*, namely, the number of queries to the oracle that are *sufficient* to find an $\epsilon$-approximate minima of a convex function:

| | $L$-Lipschitz | $\beta$-smooth |
|---|---|---|
| Convex | $O(B^2L^2/\epsilon^2)$ | $O(B^2\beta/\epsilon)$ |
| $\alpha$-strongly convex | $O(L^2/(\alpha\epsilon))$ | $O((\beta/\alpha)\log(B^2/\epsilon))$ |

**Remark 2.** *To conclude, let us go back to our motivating example of minimizing the empirical risk function* $R(x) = \frac{1}{m}\sum_{i=1}^{m} \ell(x^T\Phi(W_i), Y_i)$. *Under the assumptions of Proposition 3,* $R$ *is* $GL_\ell$*-Lipschitz (see Section 1). Assuming that the loss function* $\ell$ *is convex, the subgradient descent method then yields:*

$$\textit{OPTIMIZATION} := R(\hat{X}_t) - R(X^\star) \leq \frac{BGL_\ell}{\sqrt{t}}.$$

*Here the constants are explicit, and they match the ones in Proposition 3 and Proposition 4 for the STATISTICS term. In particular, following the rationale of optimizing the* OPTIMIZATION *term in* (3) *up to the accuracy given by the* STATISTICS *term in* (3)*, one finds that it suffices to run the algorithm for* $t \sim m$ *number of steps.*

**Week 3: Lower bounds for oracle complexity.**

Readings: Parts of Chapter 3 in [3]. Details are below.

We see that within the first order oracle model one can prove lower-bounds for the amount of calls to the oracle that are *needed* to achieve a certain accuracy $\epsilon$. In the non-smooth case we show that the rates achieved by subgradient descent methods, i.e., $O(BL/\sqrt{t})$ for convex functions and $O(L^2/(\alpha t))$ for $\alpha$-strongly convex functions, can not be improved (**Theorem 3.13 in [3]; we will only cover the proof of the first statement for convex $L$-Lipschitz functions**). On the other hand, in the smooth case we prove oracle-complexity lower bounds that are better than the ones achieved by subgradient methods, namely, $O(D^2\beta/t^2)$ for smooth functions (**Theorem 3.14 in [3]**) and $O(D^2\exp(-t\sqrt{\alpha/\beta}))$ for smooth and strongly convex functions (Theorem 3.15 in [3]), where $D$ is the diameter of $\mathcal{X}$, i.e., $D := \max_{x,y\in\mathcal{X}} \|x-y\|_2$. To recap, the optimal rates are:

| | $L$-Lipschitz | $\beta$-smooth |
|---|---|---|
| Convex | $O(BL/\sqrt{t})$ | $O(D^2\beta/t^2)$ |
| $\alpha$-strongly convex | $O(L^2/(\alpha t))$ | $O(D^2 e^{-t\sqrt{\alpha/\beta}})$ |

**Week 4: Application: Boosting.**

Reading: Parts of Part II, Section 1.4 and Part II, Section 2.2 in [6].

We apply the projected subgradient descent algorithm to solve an important example in machine learning: Boosting. For a given loss function $\varphi$ (recall, $\ell(z,y) = \varphi(-zy)$ in classification), Boosting can be written as the problem:

$$\begin{aligned} \text{minimize} \quad & r(x) = \mathbf{E}\varphi(-Yx^T\Phi(W)) \\ \text{subject to} \quad & x \in \Delta_n, \end{aligned}$$

where $\Delta_n := \{x \in [0,1]^n : \sum_{k=1}^n x_k = 1\}$ is the $n$-dimensional probability simplex. Here the feature map $\Phi : \mathcal{W} \to \{-1,1\}^n$ encodes the prediction of the $n$ base classifiers on the given data point it is applied to, and $x^T\Phi$ is a convex combination of these classifiers. Following the error decomposition given in Proposition 2, we show that in the case of Boosting the *STATISTICS* term is $O(\log n/\sqrt{m})$, which only grows logarithmically with the number of base classifiers $n$. This is a nice feature in applications where the number $n$ of base classifiers is very large. Hence, one would hope to be able to design an algorithmic procedure that only uses $\log n$ calls to the first order oracle in order to match the accuracy of the *STATISTICAL* term to find an approximate solution to the empirical risk minimization problem:

$$\begin{aligned} \text{minimize} \quad & R(x) = \frac{1}{m}\sum_{i=1}^{m} \varphi(-Y_i x^T\Phi(W_i)) \\ \text{subject to} \quad & x \in \Delta_m. \end{aligned}$$

Note that if $\varphi$ is $L_\varphi$-Lipschitz on $[-1,1]$, then the empirical risk $R$ is $L_\varphi\sqrt{n}$-Lipschitz on $[-1,1]$:

$$|R(x)-R(y)| \le \frac{1}{m}\sum_{i=1}^m |\varphi(-Y_i x^T\Phi(W_i)) - \varphi(-Y_i y^T\Phi(W_i))| \le \frac{1}{m}\sum_{i=1}^m L_\varphi \|Y_i(x-y)^T\Phi(W_i)\|_2$$
$$\le L_\varphi \|\Phi(W_i)\|_2 \|x-y\|_2 \le \sqrt{n} L_\varphi \|x-y\|_2.$$

Hence, projected subgradient descent yields a rate $O(L_\varphi\sqrt{n/t})$. Imposing the condition $L_\varphi\sqrt{n/t} \lesssim \log n/\sqrt{m}$, we find that $t \gtrsim L_\varphi^2 nm/\log n^2$, which does *not* scale logarithmically with $n$ as we hoped for!

Boosting is one example where one would like to apply subgradient descent methods on a non-Euclidean space, i.e., on the probability simplex $\Delta_n$, but these methods are not suited for this type of problems as they are really designed for the Euclidean geometry. Note, in fact, that all the definitions we gave for $L$-Lipschitz, $\beta$-smoothness, and $\alpha$-strong convexity are expressed in terms of the Euclidean norm $\|\cdot\|_2$. This motivates the design of a new class of algorithms that can adapt to the geometry of the problem at hand, as we will see next.

**Week 5: Non-Euclidean setting: mirror descent.**

Readings: Section 4.1, 4.2, and 4.3 in [3]. Details are below.

We introduce the mirror descent algorithm and show that this algorithm adapts also to non-Euclidean geometries. We say that $f$ is $L$-Lipschitz in some norm $\|\cdot\|$ if $|f(x)-f(y)| \le L\|x-y\|$ for any $x,y \in \mathbb{R}^n$. We show that for such convex functions mirror descent yields rates that scale like $O(LB/\sqrt{t})$ (**Theorem 4.2 in [3]**). This allows mirror descent to achieve a rate $O(\sqrt{\log n/t})$ in the example of boosting, where one wants to minimize a function with subgradient bounded in the $\ell_\infty$-norm over the probability simplex (**Section 4.3 in [3]**), in contrast to the rate $O(\sqrt{n/t})$ achieved by subgradient descent, as shown last time. We also show that in the case of the Euclidean norm, the algorithm is equivalent to projected subgradient descent.

**Week 6: Acceleration by coupling gradient descent and mirror descent.**

Readings: [1].

We show that mirror descent can be coupled with gradient descent to yield an *accelerated* algorithm that can achieve the lower bound for smooth functions that we proved in Week 3. The acceleration technique is linear coupling, from the paper [1].

**Week 7: Non-Euclidean setting: Frank-Wolfe.**

Readings: Section 3.3 in [3]. Details are below.

In many applications, the computational bottleneck in gradient descent methods is given by the projection step on the constraint set $\mathcal{X}$. To address this issue, we study the Frank-Wolfe algorithm (a.k.a. conditional gradient descent), which replaces the projection step with a linear optimization over $\mathcal{X}$, which in some cases can be a much simpler problem. For smooth objective functions, the Frank-Wolfe algorithm achieves rate $O(B^2\beta/t)$ (**Theorem 3.8 in [3]**). While this rare is slow and it does not match the oracle complexity lower bound $O(D^2\beta/t^2)$, the saving in the computational complexity makes this algorithm convenient in some applications. Other advantages of this algorithm over gradient descent methods is that this algorithm can apply to smoothness in any norm ($f$ is $\beta$-smooth in some norm $\|\cdot\|$ if $\|\nabla f(x) - \nabla f(y)\|_* \le \beta\|x-y\|$ for any $x,y \in \mathbb{R}^n$, where the dual norm $\|\cdot\|_*$ is defined as $\|g\|_* = \sup_{x\in\mathbb{R}^n:\|x\|\le 1} g^T x$), and that the algorithm computes sparse iterates. We will discuss a concrete application where these benefits are substantials, the least square regression with structured sparsity (**in Section 3.3 in [3]**).

**Week 8: Stochastic oracle model.**

Readings: Section 6.1, 6.2, and 6.3 in [3].

The first order oracle model that we have investigated so far had allowed us to produce a complete theory of convex optimization, in the sense that for various classes of convex functions we can design algorithms with an *oracle complexity* that matches the lower bounds. However, the black-box model does not tell us anything about the *computational complexity*, namely, the number of elementary computations that an algorithm needs to do to solve the problem (indeed, to address this question we should "open" the black-box). Going back to the original problem in machine learning that we set to solve, as described in Section 1, we note that the first order oracle model is too general for our needs, and that there might be computational savings to be gained in working with a model of computation that is more fined-tuned to our problem. This consideration motivates us to consider the *stochastic* first order oracle model, where for any point $x \in \mathcal{X}$ the oracle gives back an unbiased estimator of the gradient at $x$. Within this framework, two approaches have attracted a lot of attention lately, due to the computational savings they yield.

Multiple passes over the data. Within the setting of empirical risk minimization, one notices that we know the *global* structure of the function we want to minimize, namely, $R = \frac{1}{m}\sum_{i=1}^{m} R_i$. Hence, for instance, given $x \in \mathcal{X}$ we can have access to $\nabla R_i(x)$ for a specific $i \in \{1, \ldots, m\}$, which is something that is not allowed in the oracle model previously discussed where only access to $\nabla R(x)$ is granted. Note that computing $\nabla R(x)$ costs $O(m)$ operations as $R$ is the sum of $m$ terms, while computing $\nabla R_i(x)$ costs $O(1)$.

Single pass over the data. The empirical risk minimization approach described in Section 1 has allowed us to break the original problem we care about, namely, problem (1), into a *STATISTICAL* component and an *OPTIMIZATION* component, see Proposition 2. We show that within the stochastic oracle model there is a way to directly address problem (1) and yield a bound for $r(\hat{X}_t) - r(x^\star)$, *combining* both statistics and optimization. Recall that we do not know the function $r$ as we assume that we do not know the distribution of the random variables $W$ and $Y$; as a consequence, we also do not know the gradient of $r$, so we can not minimize $r$ within the classical first order oracle model. On the other hand, we can treat the $\nabla R_i(x)$'s as unbiased estimators of $\nabla r$.

# 2 Convexity. Black-box model. Projected gradient descent methods.

*Speaker: Anthony Caterini, 19/10/2017.*

We want to focus first on introducing the notion of convexity, which will admit some very nice properties for optimization. In particular, convex functions always have subgradients, and any local minimum is also a global minimum. We will also introduce projected subgradient descent methods and prove convergence rates in the cases where $f$ is convex and either $L$-Lipschitz continuous or $\beta$-smooth. In these two cases, the essence of the method is the same, although the particulars are slightly different.

The methods in this section relate to controlling the *OPTIMIZATION* term in empirical risk minimization. We note that although the empirical risk $R$ may not be convex, we can still run the algorithms that we discuss to find local optima.

Note that in this section, we rely heavily on [3], although the extension of projected gradient descent to time-varying step sizes comes from [6, Lecture 11, Page 5].

## 2.1 Convexity

We begin by recalling the definition of a convex function and convex set.

**Definition 2** (Convex function and set)**.** *A function $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$ is* convex *if, for all $(x, y, \gamma) \in \mathcal{X} \times \mathcal{X} \times [0, 1]$,*

$$f((1-\gamma)x + \gamma y) \leq (1-\gamma)f(x) + \gamma f(y).$$

*In other words, $f$ always lies below its chords. Similarly, a* set *$\mathcal{X} \subset \mathbb{R}^n$ is* convex *if, for all $(x, y, \gamma) \in \mathcal{X} \times \mathcal{X} \times [0, 1]$,*

$$(1-\gamma)x + \gamma y \in \mathcal{X}.$$

*In other words, $\mathcal{X}$ contains all of its line segments.*

### 2.1.1 Existence of subgradients

We will also define the notion of a subgradient – useful when $f$ is not differentiable – and show that a convex function always admits subgradients. This will be important when we introduce the projected subgradient method, as we will not assume differentiability of $f$. We will instead assume that $f$ is convex, and that we have access to a *first order oracle* that can give us a subgradient of $f$ at each point.

**Definition 3** (Subgradient)**.** *Let $\mathcal{X} \subset \mathbb{R}^n$ and $f : \mathcal{X} \to \mathbb{R}$. Then, $g \in \mathbb{R}^n$ is a* subgradient *of $f$ at $x \in \mathcal{X}$ if, for any $y \in \mathcal{X}$, one has*

$$f(x) - f(y) \leq g^T(x - y).$$

*The set of subgradients of $f$ at $x$ is denoted $\partial f(x)$.*

Note that we can rewrite the above inequality as $f(y) \geq f(x) + g^T(y - x)$. Thus, each subgradient defines a plane that supports $f$. Also notice the set of subgradients at a point at which $f$ is non-differentiable can be infinite. Conversely, we will see that, for convex and differentiable functions, the standard gradient is also a subgradient. We first mention the supporting hyperplane theorem and the definition of the epigraph before showing this.

**Theorem 1** (Supporting Hyperplane Theorem)**.** *Let $\mathcal{X}$ be a convex set, and $x_0 \in \partial\mathcal{X}$ (the boundary of $\mathcal{X}$). Then, there exists $w \in \mathbb{R}^n$, $w \neq 0$, such that*

$$\forall x \in \mathcal{X}, w^T x \geq w^T x_0.$$

**Definition 4** (Epigraph)**.** *The* epigraph *of a function $f : \mathcal{X} \to \mathbb{R}$ is*

$$epi(f) = \{(x, t) \in \mathcal{X} \times \mathbb{R} : t \geq f(x)\}.$$

*Also, a function is convex if and only if its epigraph is convex.*

**Proposition 10** (Existence of subgradients)**.** *Let $\mathcal{X}$ be convex and $f : \mathcal{X} \to \mathbb{R}$. If $\forall x \in \mathcal{X}, \partial f(x) \neq \emptyset$, then $f$ is convex. Conversely, if $f$ is convex, then for any $x \in int(\mathcal{X}), \partial f(x) \neq \emptyset$. Furthermore, if $f$ is convex and differentiable at $x$, then $\nabla f(x) \in \partial f(x)$.*

*Proof.* For the first claim, let $g \in \partial f((1-\gamma)x + \gamma y)$, for any $x, y \in \mathcal{X}$ and $\gamma \in [0, 1]$. Then,

$$f((1-\gamma)x + \gamma y) \leq f(x) + \gamma g^T(y - x), \tag{6}$$

$$f((1-\gamma)x + \gamma y) \leq f(y) + (1-\gamma)g^T(x - y). \tag{7}$$

Then, adding $(1-\gamma)*(6)$ with $\gamma*(7)$ clearly shows that $f$ is a convex function.

Now let us suppose that $f$ is convex. Let $x \in \mathcal{X}$. Clearly, $(x, f(x)) \in \partial \text{epi}(f)$, and $\text{epi}(f)$ is a convex set. Thus, by the Supporting Hyperplane Theorem, there exists $(a, b) \in \mathbb{R}^n \times \mathbb{R}$, $(a, b) \neq 0$, such that for all $(y, t) \in \text{epi}(f)$,

$$a^T x + bf(x) \geq a^T y + bt. \tag{8}$$

By allowing $t \to \infty$, we can see that $b \leq 0$. Also, if we assume $x \in \text{int}(\mathcal{X})$, then $y = x + \varepsilon a \in \mathcal{X}$ for small $\varepsilon > 0$. Plugging this into (8), we see that $b = 0$ implies $a = 0$, which is a contradiction; therefore, $b < 0$. We can now rewrite (8) with $t = f(y)$ as

$$f(x) - f(y) \leq \frac{1}{|b|} a^T(x - y),$$

i.e. $a/|b|$ is a subgradient of $f$.

Finally, if $f$ is convex and differentiable, it is not hard to show that $\nabla f(x) \in \partial f(x)$ by definition. □

### 2.1.2 first order optimality condition

We also have a nice first order optimality condition when dealing with convex functions.

**Proposition 11** (First order optimality condition)**.** *Let $f$ be convex, and $\mathcal{X}$ be a closed set on which $f$ is differentiable. Then,*

$$x^* \in \arg\min_{x \in \mathcal{X}} f(x)$$

*if and only if for all $y \in \mathcal{X}$,*

$$\nabla f(x^*)^T(x^* - y) \leq 0.$$

*Proof.* If we first assume that $\nabla f(x^*)^T(x^* - y) \leq 0$ for all $y \in \mathcal{X}$, then since the gradient is a subgradient,

$$f(x^*) - f(y) \leq \nabla f(x^*)^T(x^* - y) \leq 0,$$

i.e. $f(x^*) \leq f(y)$ for all $y \in \mathcal{X}$.

Now, if we assume $x^* \in \arg\min_{x \in X} f(x)$, we know that $f$ is locally non-decreasing around $x^*$. Define $h(t) = f(x^* + t(y - x^*))$ for all $y \in \mathcal{X}$ – the rate of change of $f$ along the line from $x$ to $y$. We require $h'(0) \geq 0$ since $x^*$ is a minimizer. Thus,

$$h'(0) = \nabla f(x^*)^T(y - x^*) \geq 0,$$

i.e. $\nabla f(x^*)^T(x^* - y) \leq 0$ for all $y \in \mathcal{X}$. □

We will see that the above proposition proves to be very useful when proving the convergence rates of projected gradient descent methods. Of course the above proposition holds with equality when $x^*$ is an interior point, as $\nabla f(x^*) = 0$ in this case. However, it may also be the case that $x^*$ is on the boundary of $\mathcal{X}$ and not necessarily a minimum in the space in which $\mathcal{X}$ is embedded, and this is the less-intuitive case that Proposition 11 handles well.

## 2.2 Black-box model

In the black-box model of computation, we assume that we know the constraint set $\mathcal{X}$ but not the objective function $f : \mathcal{X} \to \mathbb{R}$. Nonetheless, we assume that we can make queries to an *oracle* and receive some information about $f$ as output. Of particular interest to us is a **first order oracle**, which takes a point $x \in \mathcal{X}$ as input and outputs a subgradient of $f$ at $x$. Of course, if $f$ is convex, we will always be able to find a subgradient. We are interested in understanding the *oracle complexity* of convex optimization – the number of necessary and sufficient queries to the oracle to find an $\varepsilon$-approximate minima.

## 2.3 Projected gradient descent methods

We will now move into developing projected gradient descent algorithms for constrained optimization problems of the form

$$\min_{x \in \mathcal{X}} f(x),$$

where $f : \mathcal{X} \to \mathbb{R}$, and $\mathcal{X} \subset \mathbb{R}^n$ is the constraint set. If $f$ is differentiable and $\mathcal{X} = \mathbb{R}^n$, we can write out the basic gradient descent method as

$$x_{t+1} = x_t - \eta \nabla f(x_t),$$

for some initial point $x_1 \in \mathbb{R}^n$ and step size $\eta > 0$. Methods of this type can obtain an *oracle complexity* that is independent of the dimension and are thus attractive in the high-dimensional setting.

However, we are interested in cases where the constraint set is a strict subset of $\mathbb{R}^n$ and $f$ may not be differentiable. We thus turn to a *projected subgradient descent* method instead, where each iteration performs the following: take a step in the direction of a subgradient of $f$, and then project this new point back onto the constraint set $\mathcal{X}$. We therefore begin by defining the projection operator and describe one of its properties.

Note that throughout this section, we will assume that the constraint set $\mathcal{X}$ is compact and convex, and is contained in a Euclidean ball of radius $B$ centred at $x_1 \in \mathcal{X}$. Furthermore, $\|\cdot\|$ denotes the Euclidean norm.

**Definition 5** (Projection operator)**.** *For all $y \in \mathbb{R}^n$, the* projection operator *on $\mathcal{X}$, $\Pi_{\mathcal{X}}$, is defined by*

$$\Pi_{\mathcal{X}}(y) = \arg\min_{x \in \mathcal{X}} \|x - y\|.$$

**Lemma 2.** *Let $x \in \mathcal{X}$ and $y \in \mathbb{R}^n$. Then,*

$$\left(\Pi_{\mathcal{X}}(y) - x\right)^T \left(\Pi_{\mathcal{X}}(y) - y\right) \leq 0,$$

*which also implies $\|\Pi_{\mathcal{X}}(y) - x\|^2 + \|y - \Pi_{\mathcal{X}}(y)\|^2 \leq \|y - x\|^2$.*

*Proof.* This is a direct consequence of Proposition 11 since $\Pi_{\mathcal{X}}(y)$ is a minimizer of $h_y(z) = \|y - z\|$, and $\nabla h_y(z) = (z - y)/\|z - y\|$. □

### 2.3.1 $L$-Lipschitz functions

We now introduce projected subgradient descent for the case where the subgradients of $f(x)$ satisfy $\|g\| \leq L$, for all $x \in \mathcal{X}$. Note that this immediately implies $f$ is $L$-Lipschitz continuous. The algorithm consists of two update steps at every iteration $t \geq 1$:

$$\begin{aligned} y_{t+1} &= x_t - \eta_t g_t, \text{where } g_t \in \partial f(x_t), \\ x_{t+1} &= \Pi_{\mathcal{X}}(y_{t+1}). \end{aligned}$$

We state and prove the convergence of this method in the $L$-Lipschitz case below.

**Theorem 2** ($L$-Lipschitz Projected Subgradient Descent)**.** *Suppose $\mathcal{X}$ is contained in a Euclidean ball of radius $B$ centred at $x_1$. Suppose also that for every $x \in \mathcal{X}$, the subgradients $g \in \partial f(x)$ satisfy $\|g\| \leq L$ and that $f$ is convex. Then, the projected subgradient descent method with $\eta_s \equiv \eta = \frac{B}{L\sqrt{t}}$ satisfies*

$$f\left(\frac{1}{t}\sum_{s=1}^{t} x_s\right) - f(x^*) \leq \frac{LB}{\sqrt{t}} \quad \textit{and} \quad f(x^o) - f(x^*) \leq \frac{LB}{\sqrt{t}}, \tag{9}$$

*where $x^o \in \arg\min_{x \in \{x_1, \ldots, x_t\}} f(x)$. Moreover, with $\eta_s = \frac{B}{L\sqrt{s}}$, then $\exists c > 0$ (for instance, $c = 2(1 + \log 2)$) such that*

$$f\left(\left(\sum_{s=\lceil t/2 \rceil + 1}^{t} \eta_s\right)^{-1} \sum_{s=\lceil t/2 \rceil + 1}^{t} \eta_s x_s\right) - f(x^*) \leq c\frac{LB}{\sqrt{t}} \quad \textit{and} \quad f(x^o) - f(x^*) \leq c\frac{LB}{\sqrt{t}}. \tag{10}$$

*Proof.* Recall that $2a^T b = \|a\|^2 + \|b\|^2 - \|a - b\|^2$. Then, for any $1 \leq s \leq t$,

$$\begin{aligned} f(x_s) - f(x^*) &\leq g_s^T (x_s - x^*) \\ &= \frac{1}{\eta}(x_s - y_{s+1})^T (x_s - x^*) \\ &= \frac{1}{2\eta}\left(\|x_s - x^*\|^2 + \|x_s - y_{s+1}\|^2 - \|y_{s+1} - x^*\|^2\right) \\ &= \frac{1}{2\eta}\left(\|x_s - x^*\|^2 - \|y_{s+1} - x^*\|^2\right) + \frac{\eta}{2}\|g_s\|^2. \end{aligned}$$

Now, from Lemma 2, we know that $\|y_{s+1} - x^*\| \geq \|x_{s+1} - x^*\|$. Therefore, summing from $s = 1$ to $t$, we get

$$\frac{1}{t}\sum_{s=1}^{t}(f(x_s) - f(x^*)) \leq \frac{1}{2\eta t}\left(\|x_1 - x^*\|^2 - \|x_{t+1} - x^*\|^2\right) + \frac{\eta}{2}L^2 \leq \frac{B^2}{2\eta t} + \frac{\eta L^2}{2}.$$

Selecting $\eta = \frac{B}{L\sqrt{t}}$ to minimize the right-hand side of the above inequality gives the first result in (9) (since $f\left(\frac{1}{t}\sum_{s=1}^{t} x_s\right) \leq \frac{1}{t}\sum_{s=1}^{t} f(x_s)$ by Jensen's inequality). Clearly, $f(x^o) \leq \frac{1}{t}\sum_{s=1}^{t} f(x_s)$, proving the second result of (9).

Now consider the case of an adaptive step size $\eta_s$. The above derivation yields

$$f(x_s) - f(x^*) \leq \frac{1}{2\eta_s}\left(\|x_s - x^*\|^2 - \|y_{s+1} - x^*\|^2\right) + \frac{\eta_s}{2}\|g_s\|^2.$$

If we want to apply the same argument as above and get a telescoping sum with cancellations, we need to take a weighted sum weighted by $\eta_s$. Namely, replacing $t^{-1}\sum_{s=1}^t$ with $(\sum_{s=1}^t \eta_s)^{-1}\sum_{s=1}^t \eta_s$ we get

$$\left(\sum_{s=1}^t \eta_s\right)^{-1} \sum_{s=1}^t \eta_s \left(f(x_s) - f(x^*)\right) \leq \left(\sum_{s=1}^t \eta_s\right)^{-1} \left(\frac{B^2}{2} + \left(\sum_{s=1}^t \eta_s^2\right)\frac{L^2}{2}\right),$$

which reduces to the previous result when $\eta_s \equiv \eta$. We want the right-hand-side to go to zero as $t$ increases, so we need both $\sum \eta_s \to \infty$ and $\frac{\sum \eta_s^2}{\sum \eta_s} \to 0$. This is the case if we take $\eta_s = \frac{K}{\sqrt{s}}$, as $\sum_{s=1}^t \eta_s \geq c_1 K\sqrt{t}$ (e.g., $c_1 = 1$) and $\sum_{s=1}^t \eta_s^2 \leq c_2 K^2 \log t$ (e.g., $c_2 = 1 + 1/\log 2$ if $t \geq 2$, using that $\sum_{s=1}^t 1/s \leq 1 + \int_{s=0}^t \frac{1}{s} ds = 1 + \log t \leq c_2 \log t$). We can then choose $K = \frac{B}{L}$ such that

$$\left(\sum_{s=1}^t \eta_s\right)^{-1} \sum_{s=1}^t \eta_s \left(f(x_s) - f(x^*)\right) \leq \frac{B^2}{2c_1 K\sqrt{t}} + \frac{c_2 K L^2 \log t}{2c_1\sqrt{t}} \leq \left(\frac{1+c_2}{2c_1}\right) LB\sqrt{\frac{\log t}{t}}.$$

To get rid of the log term, we note that if we only sum from $\lceil t/2\rceil + 1$ to $t$, for $t \geq 3$, we have $\sum_{s=\lceil t/2\rceil+1}^t \eta_s \geq c_1' K\sqrt{t}$ (e.g., $c_1' = 1/4$, as $\sum_{s=\lceil t/2\rceil+1}^t 1/\sqrt{s} \geq \frac{t-1}{2\sqrt{t}} = \frac{\sqrt{t}}{2}(1 - 1/t) \geq \frac{\sqrt{t}}{4}$) and $\sum_{s=\lceil t/2\rceil+1}^t \eta_s^2 \leq c_2' K^2$ (e.g., $c_2' = \log 2$ using that $\sum_{s=\lceil t/2\rceil+1}^t \frac{1}{s} \leq \int_{t/2}^t \frac{1}{s} ds = \log 2$). Therefore, we finally have that

$$\min_{1\leq s\leq t} f(x_s) - f(x^*) \leq \min_{\lceil t/2\rceil+1 \leq s \leq t} f(x_s) - f(x^*) \leq \left(\sum_{s=\lceil t/2\rceil+1}^t \eta_s\right)^{-1} \sum_{s=\lceil t/2\rceil+1}^t \eta_s \left(f(x_s) - f(x^*)\right) \leq c\frac{LB}{\sqrt{t}},$$

with $c = \left(\frac{1+c_2'}{2c_1'}\right)$, where we used that the minimum element of a set is always less or equal to a convex combination of the elements. The proof of both results in (10) follows again by Jensen's inequality. □

The result of (10) is promising since we would rather use an adaptive step size than one that depends on the total number of iterations $t$, to be set and fixed in advance, i.e., prior to running the algorithm. Unfortunately, both step sizes still depend on $B$ and $L$ – quantities which may not be known. The results in Theorem 2 demonstrate that the projected subgradient method for convex and $L$-Lipschitz functions exhibits a convergence rate of $O\left(\frac{BL}{\sqrt{t}}\right)$. We can also phrase this in terms of *oracle complexity*: as at each iteration we make a constant (in fact, one) number of calls to the oracle, then to achieve $\varepsilon$-convergence, we require $\frac{BL}{\sqrt{t}} \leq \varepsilon$, or $t \geq \frac{B^2L^2}{\varepsilon^2}$.

### 2.3.2 $\beta$-smooth functions

Now, we move to the case of a $\beta$-smooth convex function. Recall a function $f : \mathcal{X} \to \mathbb{R}$ is $\beta$-smooth if, for all $x, y \in \mathcal{X}$, we have $\|\nabla f(x) - \nabla f(y)\| \leq \beta\|x - y\|$. We show some auxiliary results first for convex and $\beta$-smooth functions – deferring to [3] for the proofs – before deriving the convergence rate of projected subgradient descent with $\eta = \frac{1}{\beta}$.

**Lemma 3.** *Let $f : \mathcal{X} \to \mathbb{R}$ be a convex and $\beta$-smooth function. Then, for all $x, y \in \mathcal{X}$, we have*

$$f(x) - f(y) - \nabla f(y)^T(x - y) \leq \frac{\beta}{2}\|x - y\|^2.$$

*Proof.* Refer to the proof in [3, Lemma 3.4]. □

**Lemma 4.** *Let $x, y \in \mathcal{X}$, $x^+ = \Pi_{\mathcal{X}}\left(x - \frac{1}{\beta}\nabla f(x)\right)$, and $g_{\mathcal{X}}(x) = \beta(x - x^+)$. Then, we have the following:*

$$f(x^+) - f(y) \leq g_{\mathcal{X}}(x)^T(x - y) - \frac{1}{2\beta}\|g_{\mathcal{X}}(x)\|^2.$$

*Proof.* Refer to the proof in [3, Lemma 3.6]. □

**Theorem 3.** *[$\beta$-Smooth Projected Subgradient Descent] Let $f : \mathcal{X} \to \mathbb{R}$ be convex and $\beta$-smooth on $\mathcal{X}$. Also, suppose $\mathcal{X}$ is contained in a Euclidean ball of radius $B$ centred at $x_1$. Then, the projected subgradient descent method with $\eta = \frac{1}{\beta}$ satisfies*

$$f(x_t) - f(x^*) \leq \frac{3\beta B^2 + f(x_1) - f(x^*)}{t}. \tag{11}$$

*Proof.* From Lemma 4 and since $x_{s+1} = \Pi_{\mathcal{X}}(x_s - \frac{1}{\beta}\nabla f(x_s))$, we have the following:

$$\begin{aligned}
f(x_{s+1}) - f(x_s) &\leq g_{\mathcal{X}}(x_s)^T(x_s - x_s) - \frac{1}{2\beta}\|g_{\mathcal{X}}(x_s)\|^2 = -\frac{1}{2\beta}\|g_{\mathcal{X}}(x_s)\|^2, \text{ and} \\
f(x_{s+1}) - f(x^*) &\leq g_{\mathcal{X}}(x_s)^T(x - x^*) \leq \|g_{\mathcal{X}}(x_s)\| \cdot \|x_s - x^*\|,
\end{aligned}$$

since $f(x_{s+1}) \geq f(x^*)$. Also, we can show that $\|x_s - x^*\|$ is decreasing with $s$: from Lemma 4, we have $g_{\mathcal{X}}(x_s)^T(x_s - x^*) \geq \frac{1}{2\beta}\|g_{\mathcal{X}}(x_s)\|$, and thus

$$\begin{aligned}
\|x_{s+1} - x^*\|^2 &= \|x_s - \frac{1}{\beta}g_{\mathcal{X}}(x_s) - x^*\|^2 \\
&= \|x_s - x^*\|^2 - \frac{2}{\beta}g_{\mathcal{X}}(x_s)^T(x_s - x^*) + \frac{1}{\beta^2}\|g_{\mathcal{X}}(x_s)\|^2 \leq \|x_s - x^*\|^2.
\end{aligned}$$

We can therefore bound the difference $f(x_{s+1}) - f(x^*)$ in terms of the difference at the previous iterate:

$$\begin{aligned}
f(x_{s+1}) - f(x^*) &= (f(x_{s+1}) - f(x_s)) + (f(x_s) - f(x^*)) \\
&\leq f(x_s) - f(x^*) - \frac{1}{2\beta}\|g_{\mathcal{X}}(x_s)\|^2 \\
&\leq f(x_s) - f(x^*) - \frac{\|g_{\mathcal{X}}(x_s)\|^2\|x_s - x^*\|^2}{2\beta\|x_1 - x^*\|^2} \\
&\leq f(x_s) - f(x^*) - \frac{(f(x_{s+1}) - f(x^*))^2}{2\beta\|x_1 - x^*\|^2}.
\end{aligned}$$

Thus, we can prove (11) using induction. It is trivial to show the base case $t = 1$. Then, assuming it is true for $t = s$,

$$\begin{aligned}
f(x_{s+1}) - f(x^*) &\leq f(x_s) - f(x^*) - \frac{(f(x_{s+1}) - f(x^*))^2}{2\beta\|x_1 - x^*\|^2} \\
&\leq \frac{3\beta B^2 + f(x_1) - f(x^*)}{s}, \text{ by the inductive hypothesis} \\
&\leq \frac{3\beta B^2 + f(x_1) - f(x^*)}{s+1}.
\end{aligned}$$

Therefore, (11) is true for all $t \in \mathbb{N}$. □

The above theorem demonstrates that for $\beta$-smooth convex functions, the error in estimating the optimal function value $O\left(\frac{\beta B^2}{t}\right)$, corresponding to an oracle complexity of $O\left(\frac{\beta B^2}{\varepsilon}\right)$. We notice that the error rate decreasing faster in $t$ in this case than in the Lipschitz case, but the dependence on $B^2$ instead of $B$ is more loose, and we will see in subsection 2.4 that we cannot compare $\beta$ and $L$ directly. We also note here that the step size $\eta$ depends on $\beta$, which can be difficult to find in practice. As in the Lipschitz case, we again have no explicit reference to the underlying dimension $n$, although both $B$ and $\beta$ implicitly depend on $n$.

## 2.4 Remark on convexity, strong convexity, smoothness, and Lipschitz continuity

Finally, having shown convergence results for convex $\beta$-smooth and $L$-Lipschitz functions, it is important to note the geometric connection between these types of functions and convex functions. Recall the Taylor series expansion of $f(y)$ around $x$:

$$f(y) \approx f(x) + \nabla f(x)^T(y-x) + \frac{1}{2}(y-x)^T\nabla^2 f(x)(y-x) + \cdots$$

Convex functions are uniformly bounded *below* by an hyperplane that can be constructed at any point $x$:

$$f(y) \geq f(x) + \nabla f(x)^T(y-x),$$

and $\alpha$-strongly convex functions are uniformly bounded *below* by this hyperplane and a quadratic:

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\alpha}{2}\|y-x\|^2.$$

On the other hand, $\beta$-smooth functions are uniformly bounded *above* by this hyperplane and a quadratic:

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{\beta}{2}\|y-x\|^2.$$

Finally, $L$-Lipschitz functions are uniformly bounded *above and below* by a double cone:

$$f(x) - L\|y-x\| \leq f(y) \leq f(x) + L\|y-x\|.$$

Another important point to note is the relative independence of $\beta$-smoothness and Lipschitz continuity. Consider the case where $f(x) = x^2$ – clearly, this is $\beta$-smooth for $\beta = 2$ (and $\alpha$-strongly convex with $\alpha = 2$). However, it is not Lipschitz continuous, as we cannot hope to uniformly upper-bound a quadratic function by a line. On the other hand, consider the hinge function $f(x) = \max(0, 1+x)$. It is easy to see that this is Lipschitz continuous with $L = 1$, but not $\beta$-smooth for any $\beta$. Thus, neither condition implies the other.

# 3 Lower bounds for oracle complexity

*Speaker: Dominic Richards, 26/10/2017.*

## 3.1 Week 2 Recap

Recall the objective is to minimise some convex function $f : \mathcal{X} \to \mathbb{R}$, being phrased as the following optimisation problem

$$\min f(x) \tag{12}$$
$$\text{s.t. } x \in \mathcal{X} \tag{13}$$

where $\mathcal{X}$ has the assumption $\sup_{x\in\mathcal{X}} ||x||_2 \leq R$, that is we can enclose the optimisation space inside a Euclidean ball of size $R$. The optimal will be denoted $x^* = \arg\min_{x\mathcal{X}} f(x)$.

An assortment of assumptions can be placed upon $f$, the following 3 are of most interest

1. L-Lipschitz: $|f(x) - f(y)| \leq L||x - y||_2$ for any $x, y \in \mathbb{R}^n$ ($||\nabla f(x)|_2 \leq L$).
2. $\beta-$smooth: $||\nabla f(x) - \nabla f(x)||_2 \leq \beta||x - y||_x$ for any $x, y \in \mathbb{R}^n$ ($\nabla^2 f(x) \preccurlyeq \beta I$ for any $x, y \in \mathbb{R}^n$)
3. $\alpha-$Strong convexity: $f(x) - f(y) \leq \nabla f(x)^T(x - y) - \frac{\alpha}{2}||x - y||_2^2$ for any $x, y \in \mathbb{R}^n$ ($\nabla^2 f(x) \succcurlyeq \alpha I$) for any $x \in \mathbb{R}^n$).

Recall that $\alpha-$Strong convexity may not be reasonable in applied problems where the sample size is small, due to is aligning implying the empirical covariance matrix is invertible. For more information on this look to proposition 2 from the notes accompanying this reading group.

Now we recall some results from the previous chapter regarding projected gradient descent. The routine, denoting the set of sub gradients for $f(x)$ as $\partial f(x)$, is the following

$$y_{t+1} = x_t - \eta g_t \quad \text{where } g_t \in \partial f(x_t) \tag{14}$$
$$x_{t+1} = \Pi_{\mathcal{X}}(y_{t+1}) \tag{15}$$

where the function $\Pi_{\mathcal{X}}(y_{t+1})$ projects the point $y_{t+1}$ back into the constraint set $\mathcal{X}$. The rate at the above routine finds $x^*$ depends upon which of the assumptions above $f$ satisfies. We now list down the bounds between the the sequence of points produced from the above routine and the optimal $x^*$ under different conditions on $f$.

**L-Lipschitz and Convex** from Theorem 3.2 from [3] with $\eta = \frac{R}{L\sqrt{t}}$

$$f\left(\frac{1}{t}\sum_{s=1}^{t} x_s\right) - f(x^*) \leq \frac{RL}{\sqrt{t}} \tag{16}$$

giving a complexity of $O(RL/\sqrt{t})$

**$\beta-$Smooth and Convex** from Theorem 3.7 in [3] with $\eta = \frac{1}{\beta}$

$$f(x_t) - f(x^*) \leq \frac{3\beta||x_1 - x^*||^2 + f(x_1) - f(x^*)}{t} \tag{17}$$

giving a complexity of $O\left(\frac{R^2\beta}{t}\right)$.

**L-Lipschitz and $\alpha-$strongly Convex** from Theorem 3.9 in [3] with $\eta_s = \frac{2}{\alpha(s+1)}$, that is the gradient step now depends upon time, we get

$$f\left(\sum_{s=1}^{t} \frac{2s}{t(t+1)} x_s\right) - f(x^*) \leq \frac{2L^2}{\alpha(t+1)} \tag{18}$$

giving a complexity of $O\left(\frac{L^2}{\alpha(t+1)}\right)$.

**$\beta-$Smooth and $\alpha-$strongly Convex** from Theorem 3.10 in [3] with $\eta = \frac{1}{\beta}$ we have

$$||x_{t+1} - x^*||^2 \leq \exp\left(-\frac{\alpha t}{\beta}\right) ||x_1 - x^*||^2 \tag{19}$$

giving a complexity of $O\left(R^2 \exp\left(-\frac{\alpha t}{\beta}\right)\right)$.

## 3.2 Week 3 - Lower Bounds

To prove lower complexity bounds a "sufficiently hard" $f$, satisfying the various conditions, must be found, that, when giving a black-box procedure a fixed number of first order oracle calls, will remain a fixed distance away from its optimum. An example of a black-box procedure being the projected gradient descent of previously - taking a position and sub gradient $(x_t, g_t)$ $g_t \in \partial f(x_t)$ and returning the next position $x_{t+1} = \Pi_{\mathcal{X}}(x_t - \eta g_t)$. Recalling that a first order oracle call at $f(x)$ returns to us a sub-gradient $g \in \partial f(x)$.

Generalising this, given a history $(x_1, g_1, \ldots, x_t, g_t)$ such that $g_s \in \partial f(x_s)$, a black-box procedure will be a mapping $(x_1, g_1, \ldots, x_t, g_t) \to x_{t+1}$. Letting $x_1 = 0$, we will assume that the black-box procedure, for any $t \geq 0$, returns a position in the span of the gradients

$$x_{t+1} \in \text{Span}(g_1, \ldots, g_t). \tag{20}$$

Additionally we denote the standard basis of $\mathbb{R}^n$ as $e_1, \ldots, e_n$, and the Euclidean ball of radius $R$ as $B_2(R) = \{x \in \mathbb{R}^n : ||x|| < R\}$.

Now for any $t \leq n$ we will show there exists an $f$ that will remain bounded away from its optimal within the first $t$ calls to a black-box procedure satisfying (20). Noting that if the number of oracle calls grows beyond the dimension ($t > n$) Vaidya method from Chapter 2 becomes viable, achieving $\varepsilon$ accuracy in $n \log(\frac{Rn}{r\varepsilon})$ first order oracle calls.

The remainder of this section is as follows: first lower oracle complexity bounds for various $f$ are listed, after which the proof for the $\beta-$Smooth and Convex case is provided. The proof for the $L-$Lipschitz $\alpha-$Strong Convex case being omitted due to it being similar in spirit to the $\beta-$Smooth case. After which we will find out there a gap between the rates achieved by projected gradient descent and the proven complexity lower bounds, suggesting new methods can be found in order to improve rates. Before moving on we highlight an additional source [2, 75-76] which included some interesting relevant points.

**Theorem 4.** *Lower Bound for L-Lipschitz and Convex / $\alpha-$Convex*

*Let $t \leq n$, $L, R > 0$. There exists an L-Lipschitz and Convex $f$ such that for any black-box procedure satisfying (20)*

$$\min_{1 \leq s \leq t} f(x_s) - \min_{x \in B_2(R)} f(x) \geq \frac{RL}{2(1+\sqrt{t})} \tag{21}$$

*thus giving oracle complexity of $\Omega\left(\frac{1}{\varepsilon^2}\right)$. Moreover if $f$ is $\alpha-$strongly Convex*

$$\min_{1\leq s\leq t} f(x_s) - \min_{x\in B_2(R)} f(x) \geq \frac{L^2}{8\alpha t} \tag{22}$$

*giving oracle complexity of* $\Omega\left(\frac{1}{\varepsilon}\right)$.
*Remark: These rates are optimal in the case of projected gradient descent.*
*Proof: See Theorem* 3.13 *[3].*

**Theorem 5.** *Lower Bound for* $\beta-$*Smooth and Convex*
*Let* $t \leq (n-1)/2, \beta > 0$. *There exists a* $\beta-$*smooth convex* $f$ *such that for any black-box procedure satisfying (20)*

$$\min_{1\leq s\leq t} f(x_s) - \min_{x\in B_2(R)} f(x) \geq \frac{3\beta}{32}\frac{||x_1 - x^*||^2}{(t+1)^2} \tag{23}$$

*giving a lower oracle complexity bound of* $\Omega\left(\frac{1}{\sqrt{\varepsilon}}\right)$.

*Proof Adapted from [3] Theorem 3.14*:
Let $A_k \in \mathbb{R}^{n\times n}$ be a matrix defined by

$$(A_k)_{i,j} = \begin{cases} 2, & i = j, i \leq k \\ -1, & j \in \{i-1, i+1\}, i \leq k, j \neq k+1 \\ 0, & \text{Otherwise} \end{cases}$$

which can alternatively be represented in the standard basis as

$$A_k = 2\sum_{i=1}^{k} e_i e_i^T - \sum_{i=1}^{k-1} e_i e_{i+1}^T - \sum_{i=2}^{k-1} e_{i+1} e_i^T \tag{24}$$

noting that $0 \preccurlyeq A_k \preccurlyeq 4I_n$ since

$$x^T A_k x = 2\sum_{i=1}^{k} x(i)^2 - 2\sum_{i=1}^{k-1} x(i)x(i+1) = x(1)^2 + x(k)^2 + \sum_{i=1}^{k-1}(x(i) - x(i+1))^2.$$

Consider now the $\beta-$smooth convex function

$$f(x) = \frac{\beta}{8}x^T A_{2t+1} x - \frac{\beta}{4}x^T e_1$$

which has the sub gradient $\partial f(x) = \frac{\beta}{4}A_{2t+1}x - \frac{\beta}{4}e_1$. As $x_1 = 0$ we have $\partial f(x_1) = -\frac{\beta}{4}e_1$, and due to (20), $x_2 = \eta_{21}e_1$ for $\eta_{21} \in \mathbb{R}$. Next iteration we have $\partial f(x_2) = \frac{\beta}{4}\left(A_{2t+1}x_2 - e_1\right)$ which, using the representation (24), is now in the direction of $e_2$. To see this consider the following

$$\begin{aligned} A_{2t+1}x_2 = \eta_{21}A_k e_1 &= \eta_{21}\left(2\sum_{i=1}^{2t+1} e_i e_i^T - \sum_{i=1}^{2t} e_i e_{i+1}^T - \sum_{i=2}^{2t} e_{i+1}e_i^T\right)e_1 \\ &= \eta_{21}\left(2e_1 - e_2\right) \end{aligned}$$

as $e_i^T e_1 = 0$ if $i \neq 1$. Therefore we can deduce that $\text{Span}(e_1, \partial f(x_2)) = \text{Span}(e_1, e_2)$.
Proceeding inductively for iterations $2 \leq s \leq t$, we have $x_s \in \text{Span}(e_1, \ldots, e_{s-1})$, and therefore $x_s = \sum_{i=1}^{s-1}\eta_{s-1,i}e_i$. Once again $\partial f(x_s)$ will contain

$$
\begin{aligned}
A_{2t+1}x_s &= \left(2\sum_{i=1}^{2t+1} e_i e_i^T - \sum_{i=1}^{2t} e_i e_{i+1}^T - \sum_{i=2}^{2t} e_{i+1}e_i^T\right)\left(\sum_{j=1}^{s-1}\eta_{s-1,j}e_j\right) \\
&= \sum_{j=1}^{s-1}\eta_{s-1,j}\left(2\sum_{i=1}^{2t+1} e_i e_i^T - \sum_{i=1}^{2t} e_i e_{i+1}^T - \sum_{i=2}^{2t} e_{i+1}e_i^T\right)e_j \\
&= \sum_{j=1}^{s-1}\eta_{s-1,j}\left(2e_j - e_{j-1} - e_{j+1}\right)
\end{aligned}
$$

which most importantly is now in the direction of $e_s$ due to element $e_{j+1}$ under the sum, meaning that $\text{Span}(e_1, \ldots, e_{s-1}, \partial f(x_s)) = \text{Span}(e_1, \ldots, e_s)$.

Intuitively, due to the right most sum in (24), one extra co-ordinate is explored per iteration. This means, for some $s \leq t$, $x_s$ will be zero in the co-ordinates not yet to explore, that is $x_s(i) = 0$ for $i = s, \ldots, n$. Therefore $x_s^T A_{2t+1} x_s = x_s^T A_s x_s$, which means the best we can do in the first $s$ iterations is find the optimal of a restricted version of $f$, namely $f_s(x)$, defined for some $k$ as

$$
f_k(x) = \frac{\beta}{8} x A_k x - \frac{\beta}{4} x^T e_1. \tag{25}
$$

The objective is to bound the difference between the optimal values of the restricted $f_s$ and global $f$, thus proving the black box method can only do so well in $t$ iterations. Denoting $f^* = \inf_{x \in \mathcal{X}} f(x)$, consider the following set of inequalities

$$
f(x_s) - f^* = f_s(x_s) - f_{2t+1}^* \geq f_s^* - f_{2t+1}^* \geq f_t^* - f_{2t+1}^* \tag{26}
$$

To see $f_s^* \geq f_t^*$ observe that

$$
\begin{aligned}
f_t(x) &= \sum_{i=1}^{s} x(i) - 2\sum_{i=1}^{s-1} x(i)x(i+1) + 2\sum_{i=s+1}^{t} x(i)^2 - 2\sum_{i=s}^{t-1} x(i)x(i+1) \\
&= f_s(x) + x(s+1)^2 + x(t)^2 + \sum_{i=s+1}^{t-1} \left(x(i) - x(i+1)\right)^2 - 2x(s)x(s+1)
\end{aligned}
$$

which when plugging in the optimal point for $f_s$ can attain $f_t(x_s^*) = f_s(x_s^*) = f_s^*$ as $x_s^*(i) = 0$ for $i = s+1, \ldots, n$.

We must now explicitly find the minimiser $x_k^*$ for $f_k$, its norm, and the corresponding objective $f_k^*$. To find $x_k^*$ we have $\partial f_k(x) = 0 \implies A_k x = e_1$, and since $x_k^* \in \text{Span}(e_1, \ldots, e_k)$, the solution becomes $x_k^*(i) = 1 - \frac{i}{k+1}$ for $i = 1, \ldots, k$. We then immediately get

$$
f_k^* = -\frac{\beta}{8}\left(1 - \frac{1}{k+1}\right). \tag{27}
$$

With the norm of $x_s^*$ then becoming

$$
||x_k^*|| = \sum_{i=1}^{k}\left(1 - \frac{i}{k+1}\right)^2 = \sum_{i=1}^{k}\left(\frac{i}{k=1}\right)^2 \leq \frac{k+1}{3}. \tag{28}
$$

Finally bringing together (26), (27), (28) we get

$$f(x_s) - f^* \geq f_t^* - f_{2t+1}^* = \frac{\beta}{8}\left(\frac{1}{t+1} - \frac{1}{2t+2}\right) \geq \frac{3\beta}{32}\frac{||x_{2t+1}^*||^2}{(t+1)^2}$$

noting that as the right hand side does not depending upon $x_s$, a minimum can be taken. □

To make comparison between the lower bound rates as well as those obtained by projected gradient descent, we look to oracle complexities contained in Figure 1.

| | | Upper Bound (PGD) | Lower Bound |
|---|---|---|---|
| Convex | $L-$Lipschitz | $O\left(\frac{1}{\varepsilon^2}\right)$ | $\Omega\left(\frac{1}{\varepsilon^2}\right)$ |
| | $\beta-$Smooth | $O\left(\frac{1}{\varepsilon}\right)$ | $\Omega\left(\frac{1}{\sqrt{\varepsilon}}\right)$ |
| $\alpha-$Strong Convex | $L-$Lipschitz | $O\left(\frac{1}{\varepsilon}\right)$ | $\Omega\left(\frac{1}{\varepsilon}\right)$ |
| | $\beta-$Smooth | $O\left(\frac{\beta}{\alpha}\log\left(\varepsilon^{-1}\right)\right)$ | $\Omega\left(\sqrt{\frac{\beta}{\alpha}}\log\left(\varepsilon^{-1}\right)\right)$ |

Figure 1: Oracle complexity- order of iterations $t$ required for $f(x_t) - f^* \leq \varepsilon$. Source for lower bound in the $\beta-$Smooth $\alpha-$Strong Convex case from Theorem 3.15 [3]. Upper Bound is oracle complexity of (PGD) Projected Gradient Descent.

Observe that there is a gap in the upper and lower bounds in the case of $\beta-$smooth functions, specifically an order $\frac{1}{\sqrt{\varepsilon}}$ and $\sqrt{\frac{\beta}{\alpha}}$ for Convex and $\alpha-$Strong Convex functions respectively. This suggests there are more optimal black box procedure over projected gradient descent that could close the complexity gap.

# 4 Application: Boosting

*Speaker: Patrick Rebeschini, 02/11/2017.*

In this section we go back to machine learning. We consider the same setting introduced in Section 1, and apply the projected subgradient descent algorithm to minimize the empirical risk in an important example: Boosting. The setting is that of binary classification, where given $m$ i.i.d. labeled data points $(W_1, Y_1), \dots, (W_m, Y_m) \in \mathcal{W} \times \mathcal{Y}$, $\mathcal{Y} = \{-1, 1\}$, coming from an unknown distribution, one wants to construct a classifier $h_{\text{hard}} : \mathcal{W} \to \{-1, 1\}$ that minimizes the probability of mistakes over the unseen data, i.e., that minimize the expected risk with respect to the "true" loss $\mathbf{E}\varphi_{\text{true}}(-Yh_{\text{hard}}(W)) = \mathbf{P}(h_{\text{hard}}(W) \neq Y)$, where $\varphi_{\text{true}}(u) = \mathbf{1}_{u \geq 0}$, and where $(W, Y)$ is a random variable (independent of everything else) coming from the same unknown distribution as the $m$ data points we are given.

As stated the problem is discrete. To get a continuous problem where we can apply gradient descent methods, we relax the setting above in two ways. First, instead of the true loss $\varphi_{\text{true}}$ we consider a convex "surrogate" $\varphi$, i.e., a convex function $\varphi$ such that $\varphi_{\text{true}}(u) \leq \varphi(u)$ for any $u \in \mathbb{R}$. Possible choices are:

- Exponential loss: $\varphi(u) = \exp(u)$.
- Hinge loss: $\varphi(u) = \max\{0, 1 + u\}$.
- Logistic loss: $\varphi(u) = \log_2(1 + \exp(u))$.

Second, instead of *hard* classifiers $h_{\text{hard}} : \mathcal{W} \to \{-1, 1\}$, we consider *soft* classifiers $h : \mathcal{W} \to [-1, 1]$. The problem we want to solve is then

$$\begin{aligned} \text{minimize} \quad & \mathbf{E}\varphi(-Yh(W)) \\ \text{subject to} \quad & h \in \mathcal{H}, \end{aligned}$$

where $\mathcal{H}$ is a given class of soft classifiers. In the setting of Boosting, one assumes to have access to a set of $n$ *base* (soft or hard, it does not matter here) classifiers encoded in a vector map $\Phi : \mathcal{W} \to [-1, 1]^n$, where $\Phi(W)_k \equiv \Phi_k(W)$ represents the outcome of the $k$-th base classifier on the data point $W$, and one wants to find the best convex combinations of these base classifiers. If we let $\Delta_n := \{x \in [0, 1]^n : \sum_{k=1}^n x_k = 1\}$ be the $n$-dimensional probability simplex, we want to find $x \in \Delta_n$ so that $x^T\Phi = \sum_{k=1}^n x_k\Phi_k$ minimizes the expected risk. In other words, we consider the family $\mathcal{H} = \{h : h = x^T\Phi \text{ for some } x \in \Delta_n\}$, and the problem we want to solve reads:

$$\begin{aligned} \text{minimize} \quad & r(x) = \mathbf{E}\varphi(-Yx^T\Phi(W)) \\ \text{subject to} \quad & x \in \Delta_n. \end{aligned}$$

This problem is a particular instance of the general formulation given in (1), with the choice $\ell(z, y) = \varphi(-zy)$ for the loss function and $\mathcal{X} = \Delta_n$ for the constraint set. We can then work within the framework of empirical risk minimization as introduced in Section 1, and use the error decomposition given in Proposition 2 to bound the *STATISTICS* term $\sup_{x \in \mathcal{X}}(r(x) - R(x)) + \sup_{x \in \mathcal{X}}(R(x) - r(x))$ and the *OPTIMIZATION* term $R(\hat{X}_t) - R(X^\star)$, respectively.

## 4.1 Statistics term

In particular, note that in the case of Boosting, as $x \in \Delta_n$ and $\Phi \in [-1, 1]^n$ and $\mathcal{Y} = \{-1, 1\}$, the loss function $\varphi$ is only evaluated in the interval $[-1, 1]$, as $-1 \leq yx^T\Phi(w) \leq 1$ for any $x \in \Delta_n, w \in \mathcal{W}$, and $y \in \mathcal{Y}$. So, we can restrict to this interval as far as the Lipschitz property of $\varphi$ goes. On $[-1, 1]$ we have the following Lipschitz constants:

- Exponential loss: $\varphi(u) = \exp(u)$, $L_\varphi = e$.

- Hinge loss: $\varphi(u) = \max\{0, 1+u\}$, $L_\varphi = 1$.
- Logistic loss: $\varphi(u) = \log_2(1+\exp(u))$, $L_\varphi = \log_2(e)\frac{e}{1+e} \approx 1.05$.

A direct application of Proposition 3 immediately yields the following result.

**Corollary 1.** *Let $\varphi$ be $L_\ell$-Lipschitz on $[-1,1]$. In the case of Boosting, we have*

$$\mathbf{E}[STATISTICS] \leq 4L_\varphi \frac{n}{\sqrt{m}}.$$

*Proof.* The proof follows from Proposition 3 noticing that in the setting of Boosting we have $\|\Phi(W)\|_2 \leq \sqrt{n}$ and $\sup_{x\in\Delta_n} \|x\|_2 \leq \sqrt{n}$. □

We could also directly apply Proposition 4 to obtain a bound in high probability, and we would similarly get a bound that depends *linearly* on $n$. Upon a closer look, we note that the assumptions of Proposition 3 and Proposition 4 are with respect to the Euclidean norm $\|\cdot\|_2$. However, while $\|\Phi(W)\|_2 \leq \sqrt{n}$ and $\sup_{x\in\Delta_n} \|x\|_2 \leq \sqrt{n}$, one has $-1 \leq x^T\Phi(W) \leq 1$ for each $x \in \Delta_n$, so perhaps one can avoid the linear dependence on $n$ by developing a new version of Proposition 3 and Proposition 4 that can avoid the use of Cauchy-Swartz $x^T\Phi(W) \leq \|x\|_2\|\Phi(W)\|_2$ and directly use that $-1 \leq x^T\Phi(W) \leq 1$. This is our goal. We prove the following results.

**Proposition 12** (Mean)**.** *Let $\varphi$ be $L_\ell$-Lipschitz on $[-1,1]$. In the case of Boosting we have*

$$\mathbf{E}[STATISTICS] \leq 4\frac{L_\varphi}{\sqrt{m}}\sqrt{2\log n}.$$

**Proposition 13** (High probability)**.** *Let $\varphi$ be $L_\ell$-Lipschitz on $[-1,1]$. In the case of Boosting we have*

$$STATISTICS \leq 2\frac{L_\varphi}{\sqrt{m}}(2\sqrt{2\log n} + \sqrt{2\log(1/\delta)}).$$

### 4.1.1 Bound on the mean

We prove Proposition 12. To bound the mean of the $STATISTICS$ term in the case of Boosting, we follow the same plan described in Section 1.1.1 and use the notion of Rademacher complexity. This time, however, we avoid the use of Cauchy-Swartz to bound the Rademacher complexity. First, following the proof of Proposition 5 it is immediate to derive a new version of this result for a loss function of the type $\ell(z,y) = \varphi(-yz)$.

**Proposition 14.** *Let $yx^T\Phi(w) \in [d_-, d_+]$ for each $x \in \mathcal{X}, w \in \mathcal{W}$, and $y \in \mathcal{Y}$. Let $\varphi$ be $L_\varphi$-Lipschitz on $[d_-, d_+]$. Then,*

$$\mathbf{E}\sup_{x\in\mathcal{X}}(r(x) - R(x)) \leq 2L_\varphi \mathbf{E}\sup_{x\in\mathcal{X}}\frac{1}{m}\sum_{i=1}^m \epsilon_i x^T\Phi(W_i)Y_i,$$

*where $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. random variables uniform in $\{-1,1\}$, independent of $(W_1,Y_1), \ldots, (W_m,Y_m)$.*

*Proof.* The proof is exactly the same as the proof of Proposition 5 up to equation (4), which now reads

$$\mathbf{E}\left[\sup_{x\in\mathcal{X}}\frac{1}{m}\sum_{i=1}^m \epsilon_i\varphi(-x^T\Phi(W_i)Y_i)\middle| Z_1,\ldots,Z_m\right] \leq L_\ell\mathbf{E}\left[\sup_{x\in\mathcal{X}}\frac{1}{m}\sum_{i=1}^m \epsilon_i x^T\Phi(W_i)Y_i\middle| Z_1,\ldots,Z_m\right].$$

□

We now derive a new version of Proposition 6 to bound $\mathbf{E}\sup_{x\in\mathcal{X}}\frac{1}{m}\sum_{i=1}^m \epsilon_i x^T\Phi(W_i)Y_i$, where instead of using the Cauchy-Swartz inequality and get a bound that depends on the Euclidean norms of $x$ and $\Phi$, we use that in the case of Boosting where $\mathcal{X}=\Delta_n$ the supremum inside the expectation is achieved in a vertex of the simplex, and hence, conditioning on the data, we are left with the (empirical) Rademacher complexity of a *finite* set, which only grows *logarithmically* with the size of the set. We first state the result about the behaviour of the Rademacher complexity $\mathcal{R}(T)$ when the set $T$ has finite cardinality.

**Lemma 5.** *Let $T\subseteq\mathbb{R}^m$ with $|T|<\infty$. We have*

$$\mathcal{R}(T)\leq\max_{t\in T}\|t\|_2\frac{\sqrt{2\log|T|}}{m}.$$

*Proof.* Recall from Definition 1 that $\mathcal{R}(T):=\mathbf{E}\sup_{t\in T}\frac{1}{m}\sum_{i=1}^m\epsilon_i t_i$. If we use Cauchy-Swartz, we would get

$$\mathcal{R}(T)\leq\sup_{t\in T}\|t\|_2\frac{\mathbf{E}\|\epsilon\|_2}{m}=\sup_{t\in T}\|t\|_2\frac{1}{\sqrt{m}},$$

which is too general for our case. In particular, this result does not use the fact that $T$ is a finite set. To get the $1/m$ rate in the bound in the case when $|T|<\infty$, we adopt two usual tricks in probability: first, we take exponentials and use Jensen's inequality; second, we bound a maximum over a set of positive numbers by its sum. For the first step, note that for any real-valued random variable $X$ and any $s>0$, Jensen's inequality yields

$$\mathbf{E}X=\frac{1}{s}\log e^{s\mathbf{E}X}\leq\frac{1}{s}\log\mathbf{E}e^{sX}.$$

For the second step, note that if $X=\max_{t\in T}X_t$, then

$$\mathbf{E}e^{sX}=\mathbf{E}\max_{t\in T}e^{sX_t}\leq\sum_{t\in T}\mathbf{E}e^{sX_t}.$$

If we choose $X_t=\sum_{i=1}^m\epsilon_i t_i$, then

$$\mathbf{E}e^{sX_t}=\prod_{i=1}^m\mathbf{E}e^{s\epsilon_i t_i}\leq\prod_{i=1}^m e^{s^2t_i^2/2}=e^{s^2\|t\|_2^2/2},$$

where we used Hoeffding's Lemma that says that for a random variable $Z$ that is bounded in the interval $[a,b]$, i.e., $a\leq Z\leq b$ a.s., one can bound its moment generating function as follows: $\mathbf{E}e^{sZ}\leq e^{s^2(b-a)^2/8}$. Putting everything together, we get

$$\mathbf{E}X\leq\frac{1}{s}\log\mathbf{E}e^{sX}\leq\frac{1}{s}\log\sum_{t\in T}\mathbf{E}e^{sX_t}\leq\frac{1}{s}\log\sum_{t\in T}e^{s^2\|t\|_2^2/2}\leq\frac{1}{s}\log|T|+\frac{s}{2}\max_{t\in T}\|t\|_2^2.$$

Optimizing this bounds over $s>0$, one finds $\mathbf{E}X\leq\max_{t\in T}\|t\|_2\sqrt{2\log|T|}$, and the proof follows. □

Armed with the previous bound on the Rademacher complexity of a finite set, we can bound the object $\mathbf{E}\sup_{x\in\mathcal{X}}\frac{1}{m}\sum_{i=1}^m\epsilon_i x^T\Phi(W_i)Y_i$ as follows.

**Proposition 15.** *In the case of Boosting, we have*

$$\mathbf{E}\sup_{x\in\Delta_n}\frac{1}{m}\sum_{i=1}^m\epsilon_i x^T\Phi(W_i)Y_i\leq\sqrt{\frac{2\log n}{m}}.$$

*Proof.* Let us define the (random) function

$$x \in \Delta_n \to g(x) := \frac{1}{m}\sum_{i=1}^{m} \epsilon_i x^T \Phi(W_i) Y_i = \sum_{j=1}^{n} x_j A_j,$$

where $A_j := \frac{1}{m}\sum_{i=1}^{m} \epsilon_i \Phi(W_i)_j Y_i$. As $g$ is a linear function, its supremum over the simplex $\Delta_n$ is achieved in at least one of the $n$ vertices of the simplex. Note that

$$\sup_{x\in\Delta_n} g(x) = \sup_{x\in\Delta_n} \sum_{j=1}^{n} x_j A_j \leq \max\{A_1, \ldots, A_n\}.$$

If we let $e_1, \ldots, e_n \in \mathbb{R}^n$ be the base vectors, then it is immediate to see that, for instance, the bound above is achieved with equality at the vertex $e_{j^\star}$, where $j^\star := \min\{k \in \{1, \ldots, n\} : A_k = \max\{A_1, \ldots, A_n\}\}$. Hence,

$$\sup_{x\in\Delta_n} g(x) = \max_{x\in\{e_1,\ldots,e_n\}} g(x),$$

which yields

$$\mathbf{E} \sup_{x\in\Delta_n} \frac{1}{m}\sum_{i=1}^{m} \epsilon_i x^T \Phi(W_i) Y_i = \mathbf{E} \max_{x\in\{e_1,\ldots,e_n\}} \frac{1}{m}\sum_{i=1}^{m} \epsilon_i x^T \Phi(W_i) Y_i.$$

Conditioning on the data we get that the empirical Rademacher complexity is the Rademacher complexity over a finite set. Using the notation $Z_i = (W_i, Y_i)$, in the spirit of Remark 1, we can use Lemma 5 to get

$$\begin{aligned}\mathbf{E}\left[\max_{x\in\{e_1,\ldots,e_n\}} \frac{1}{m}\sum_{i=1}^{m} \epsilon_i x^T \Phi(W_i) Y_i \middle| Z_1, \ldots, Z_m\right] &= \mathbf{E}\left[\max_{t\in T_{Z_1,\ldots,Z_m}} \frac{1}{m}\sum_{i=1}^{m} \epsilon_i t_i \middle| Z_1, \ldots, Z_m\right] \\ &\leq \max_{t\in T_{Z_1,\ldots,Z_m}} \|t\|_2 \frac{\sqrt{2\log|T_{Z_1,\ldots,Z_m}|}}{m} \leq \sqrt{\frac{2\log n}{m}},\end{aligned}$$

where $T_{Z_1,\ldots,Z_m} := \{(x^T\Phi(W_1)Y_1, \ldots, x^T\Phi(W_m)Y_m)^T : x \in \{e_1, \ldots, e_n\}\}$ contains $n$ vectors, i.e., $|T_{Z_1,\ldots,Z_m}| = n$, and clearly $\max_{t\in T_{Z_1,\ldots,Z_m}} \|t\|_2 \leq \sqrt{m}$. □

Proposition 14 (with $d_- = -1$, $d_+ = 1$) and Proposition 15 immediately yields the following result, from which Proposition 12 follows immediately.

**Proposition 16.** *Let $\varphi$ be $L_\ell$-Lipschitz on $[-1, 1]$. In the case of Boosting we have*

$$\mathbf{E} \sup_{x\in\mathcal{X}} (r(x) - R(x)) \leq 2\frac{L_\varphi}{\sqrt{m}}\sqrt{2\log n}.$$

### 4.1.2 Bound with high probability

We prove Proposition 13. To derive a bound in high probability for the $STATISTICS$ term in the case of Boosting, we follow the same plan described in Section **??** and use the Bounded Difference concentration inequality. This time, however, we avoid the use of Cauchy-Swartz to find the constant $c$ that bounds the variation of a single component in the Bounded Difference inequality. Below is a new version of Proposition 9, where instead of using Cauchy-Swartz to get the generic bound $x^T\Phi(W) \leq \|x\|_2\|\Phi(W)\|_2$, we use the assumption that $x^T\Phi(W) \leq [d_-, d_+]$. This assumption is satisfied in the case of Boosting with $d_- = -1$ and $d_+ = 1$. This allows to save a factor $n$ as opposed to what one would have by directly using Proposition 9 (Proposition 9 yields $c = \frac{2}{m}(|\varphi(0)| + nL_\ell)$).

**Proposition 17.** *Let $h$ be the function defined in* (5)*. Let $\mathcal{X}$ be bounded, and let $yx^T\Phi(w) \in [d_-, d_+]$ for each $x \in \mathcal{X}, w \in \mathcal{W}$, and $y \in \mathcal{Y}$. Let $\varphi$ be $L_\varphi$-Lipschitz on $[d_-, d_+]$. Then, $c \in \mathbb{R}$ satisfying the requirement of the Bounded Difference inequality is given by*

$$c = (d_+ - d_-)\frac{L_\varphi}{m}.$$

*Proof.* Fix $k \in \{1, \ldots, m\}$ and let $z = (z_1, \ldots, z_{k-1}, z_k, z_{k+1}, \ldots, z_m)$ and $z' = (z_1, \ldots, z_{k-1}, z'_k, z_{k+1}, \ldots, z_m)$. Then,

$$|h(z) - h(z')| = \left| \sup_{x\in\mathcal{X}} \left( r(x) - \frac{1}{m}\sum_{i=1}^m R_i^z(x) \right) - \sup_{x\in\mathcal{X}} \left( r(x) - \frac{1}{m}\sum_{i=1}^m R_i^{z'}(x) \right) \right|,$$

where $R_i^z(x) := \ell(x^T\Phi(w_i), y_i) = \varphi(-y_i x^T\Phi(w_i))$. If $h(z) - h(z') \geq 0$ and we let $\tilde{x} \in \mathcal{X}$ be the maximizer of $\sup_{x\in\mathcal{X}}(r(x) - \frac{1}{m}\sum_{i=1}^m R_i^z(x))$ (note that the supremum is attained by the Extreme Value Theorem), we have

$$\begin{aligned}
h(z) - h(z') &= \left( r(\tilde{x}) - \frac{1}{m}\sum_{i=1}^m R_i^z(\tilde{x}) \right) - \sup_{x\in\mathcal{X}} \left( r(x) - \frac{1}{m}\sum_{i=1}^m R_i^{z'}(x) \right) \\
&\leq \left( r(\tilde{x}) - \frac{1}{m}\sum_{i=1}^m R_i^z(\tilde{x}) \right) - \left( r(\tilde{x}) - \frac{1}{m}\sum_{i=1}^m R_i^{z'}(\tilde{x}) \right) \\
&= \frac{1}{m}(R_k^z(\tilde{x}) - R_k^{z'}(\tilde{x})) = \frac{1}{m}(\varphi(-y_k\tilde{x}^T\Phi(w_k)) - \varphi(-y'_k\tilde{x}^T\Phi(w'_k))) \\
&\leq \frac{1}{m}(\sup_{d_-\leq u\leq d_+} \varphi(u) - \inf_{d_-\leq u\leq d_+} \varphi(u))) \leq (d_+ - d_-)\frac{L_\varphi}{m},
\end{aligned}$$

where the last but one inequality comes as $d_- \leq yx^T\Phi(w) \leq d_+$ for each $y \in \mathcal{Y}, x \in \mathcal{X}, w \in \mathcal{W}$. Proceeding analogously in the case $h(z) - h(z') \leq 0$, we obtain the result. □

The proof of Proposition 13 follows the same lines as the proof of Proposition 4 in Section **??**.

*Proof of Proposition 13.* Using, respectively, Proposition 8 and Proposition 16, we have that with probability at least $1 - \delta$ the following holds:

$$\sup_{x\in\mathcal{X}}(r(x) - R(x)) \leq \mathbf{E}\left[ \sup_{x\in\mathcal{X}}(r(x) - R(x)) \right] + c\sqrt{\frac{m}{2}\log\frac{1}{\delta}} \leq 2L_\varphi\sqrt{\frac{2\log n}{m}} + c\sqrt{\frac{m}{2}\log\frac{1}{\delta}},$$

with $c = 2L_\varphi/m$ by Proposition 17, which yields

$$\mathbf{P}\left( \sup_{x\in\mathcal{X}}(r(x) - R(x)) \leq c' \right) \geq 1 - \delta,$$

where $c' := L_\ell(2\sqrt{2\log n} + \sqrt{2\log(1/\delta)})/\sqrt{m}$. As the bounds we have derived holds also for $\sup_{x\in\mathcal{X}}(R(x) - r(x))$, we have

$$\mathbf{P}\left( \sup_{x\in\mathcal{X}}(r(x) - R(x)) + \sup_{x\in\mathcal{X}}(R(x) - r(x)) \leq 2c' \right) \geq \mathbf{P}\left( \left\{ \sup_{x\in\mathcal{X}}(r(x) - R(x)) \leq c' \right\} \bigcap \left\{ \sup_{x\in\mathcal{X}}(R(x) - r(x)) \leq c' \right\} \right) \geq 1 - \delta.$$

□

## 4.2 Optimization term

Following the error decomposition given in Proposition 2, Proposition 12 and Proposition 13 show that in the case of Boosting the *STATISTICS* term is $O(\log n/\sqrt{m})$, which only grows logarithmically with the number of base classifiers $n$. This is a nice feature in applications where the number $n$ of base classifiers is very large, possibly exponential. Hence, one would hope to be able to design an algorithmic procedure that only uses $\log n$ calls to the first order oracle in order to match the accuracy of the *STATISTICAL* term to find an approximate solution to the empirical risk minimization problem:

$$\begin{aligned} \text{minimize} \quad & R(x) = \frac{1}{m}\sum_{i=1}^{m} \varphi(-Y_i x^T \Phi(W_i)) \\ \text{subject to} \quad & x \in \Delta_m. \end{aligned}$$

Given Theorem 2, we can assess the guarantees given by the projected subgradient descent method to solve this optimization problem. Note that if $\varphi$ is $L_\varphi$-Lipschitz on $[-1,1]$, then the empirical risk $R$ is $L_\varphi\sqrt{n}$-Lipschitz on $[-1,1]$:

$$\begin{aligned} |R(x) - R(y)| &\le \frac{1}{m}\sum_{i=1}^{m} |\varphi(-Y_i x^T\Phi(W_i)) - \varphi(-Y_i y^T \Phi(W_i))| \le \frac{1}{m}\sum_{i=1}^{m} L_\varphi \|Y_i(x-y)^T\Phi(W_i)\|_2 \\ &\le L_\varphi \|\Phi(W_i)\|_2 \|x-y\|_2 \le \sqrt{n} L_\varphi \|x-y\|_2. \end{aligned}$$

As $B = \sup_{x\in\Delta_n} \|x\|_2 = 1$, projected subgradient descent yields a rate $O(L_\varphi\sqrt{n/t})$. Imposing the condition $L_\varphi\sqrt{n/t} \lesssim \log n/\sqrt{m}$, we find that $t \gtrsim L_\varphi^2 nm/\log n^2$, which does *not* scale logarithmically with $n$ as we hoped for!

Boosting is one example where one would like to apply subgradient descent methods on a non-Euclidean space, i.e., on the probability simplex $\Delta_n$, but these methods are not suited for this type of problems as they are really designed for the Euclidean geometry. Note, in fact, that all the definitions we gave for $L$-Lipschitz, $\beta$-smoothness, and $\alpha$-strong convexity are expressed in terms of the Euclidean norm $\|\cdot\|_2$. This motivates the design of a new class of algorithms that can adapt to the geometry of the problem at hand, as we will see next.

# 5 Non-Euclidean setting: mirror descent

# 6 Acceleration by coupling gradient descent and mirror descent

We have seen as a consequence of Theorem 3 that if $f$ is $\beta$-smooth then projected gradient descent needs $T = O\left(\frac{\beta R^2}{\varepsilon}\right)$ iterations to obtain an $\varepsilon$-minimizer. However, we derived in Theorem 5 a lower bound in which the dependence on $\beta$ and $\varepsilon$ was $O\left(\sqrt{\frac{\beta}{\varepsilon}}\right)$. Nesterov in [5] proposed a gradient descent method and proved that its complexity matches this lower bound. This method only works for $\|\cdot\|_2$. Later, in [4] he generalized his method to allow arbitrary norms. Algorithms whose complexity is optimal are said to be accelerated. Nesterov's accelerated gradient descent has always been regarded as an obscure and unintuitive method whose proof uses "magical" algebra tricks. Since he published his seminal work in 1983, there have been several works trying to understand acceleration by proposing other accelerated methods, trying to give more intuition and showing acceleration from other points of view. For example, in [3] we can find a geometric interpretation of acceleration. Linear Coupling [1] (2014) is one of these methods. We think that it is one of the best methods to understand acceleration. We start giving some intuition about how gradient and mirror descent can be combined to obtain these accelerated method. We present first a simplified of linear coupling that also achieves acceleration but under more restrictive assumptions and finally, we present linear coupling. Linear coupling has the virtue that its analysis is the same for any norm.

## 6.1 Intuition

To understand why combining gradient and mirror descent makes sense and why it is a good idea we will note some things about both methods. We have seen in 3 that projected gradient descent is defined by

$$\begin{aligned} y_{t+1} &= x_t - \frac{1}{\beta} g_t, \text{ where } g_t \in \partial f(x_t) \\ x_{t+1} &= \Pi_{\mathcal{X}}(y_{t+1}). \end{aligned}$$

One should not be surprised that in the case that $\mathcal{X} = \mathbb{R}^n$ regular gradient descent in unconstrained optimization is defined by the rule $x_{t+1} = x_t - \frac{1}{\beta} g_t$. But, why is gradient descent, constrained or not, defined in that way? Of course, it works and provides non trivial convergence rate, but proofs of this usually only define the method and prove convergence without explaining where the method comes from, specially regarding the choice of the learning rate. It can be useful to see gradient from the following point of view. Let's do it first for unconstrained gradient descent with $\|\cdot\|_2$ only. If we are at point $x_k \in \mathbb{R}^n$ and we compute the next point by moving against the gradient, the choice of $\frac{1}{\beta}$ is the best choice for the learning rate, in the sense that we can guarantee maximal local decrease for that choice of the learning rate. This is not difficult to see using the assumptions we have at hand. The smoothness assumption tells us that along the line defined by $x_k$ and $\nabla f(x_k)$, $f$ is lower bounded by a parabola with leading coefficient $\frac{\beta}{2}$ (blue graph in Figure 6.1) whose derivative in $x_k$ coincides with the one of $f$ (restricted to the line, i.e. it is $\|\nabla f(x_k)\|$). The derivative of $x^2\beta/2$ is $x\beta = \|\nabla f(x_k)\|$ so the distance to the minimum is $\mathtt{x} = \frac{1}{\beta}\|\nabla f(x_k)\|$ and it is clear now that maximal guaranteed progress is the evaluation of $x^2\beta/2$ at $\mathtt{x}$, i.e. $\frac{1}{2\beta} = \|\nabla f(x_k)\|$. And we have just proved that the guaranteed decrease is maximal for that choice of the learning rate and we have computed how much. All these arguments can be written using inequalities, but hopefully this can be considered cleaner by some people. Proving the rate of convergence of gradient descent given the guaranteed progress at each step is straight forward. With this picture in mind, it is also very easy to derive the rate of convergence of gradient descent in the case that $f$ is also $\mu$-strongly convex. Since this assumption lower bounds $f$ by another parabola with leading coefficient $\mu/2$, we can see that the guaranteed progress is proportional to $1/2\beta$ and $f(x_k) - f(x^*)$ is upper bounded by something proportional to $1/2\mu$ so after one step the value $f(x_k) - f(x^*)$ decreases to at least $(f(x_k) - f(x^*))\left(1 - \frac{\|\nabla f(x_k)\|/2\beta}{\|\nabla f(x_k)\|/2\mu}\right) = (f(x_k) - f(x^*))\left(1 - \frac{\mu}{\beta}\right)$. And therefore

$f(x_T) - f(x^*) \leq (f(x_0) - f(x^*)) \left(1 - \frac{\mu}{\beta}\right)^T$, so an $\varepsilon$-minimizer is found in $O\left((f(x_0) - f(x^*)) \log \frac{1}{\varepsilon}\right)$ iterations.

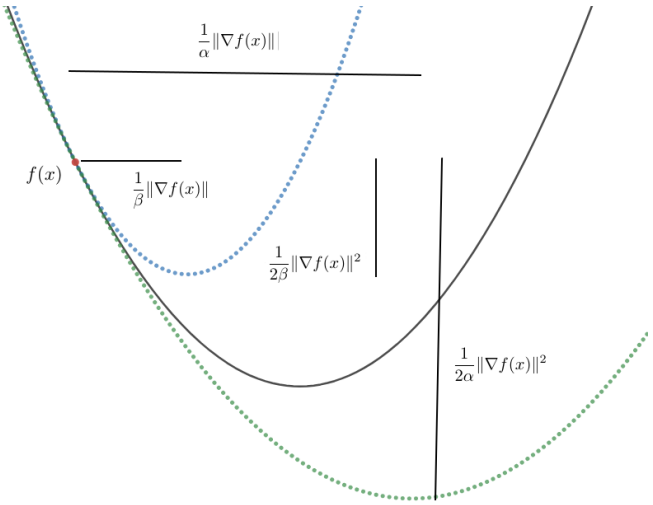


Figure 2: Visualization of the smoothness bound (blue) and the strong convexity bound (green) of a function $f$ (black).

We make two important remarks about the previous analysis. Firstly, gradient descent uses the assumption of $\beta$-smoothness to guarantee maximal decrease if we move in the direction of the gradient and secondly, the decrease is better if the norm of the gradient is large. We will note later that the regret of mirror descent is lower when the norm of the gradient is low, and this along the second remark is what we can leverage to combine gradient and mirror descent. But let's focus first in the first remark. The maximal decrease on the objective we can guarantee from $x_k$ occurs when we minimize, as we did before with our toy example, the bound that is given by the $\beta$-smoothness assumption, which is

$$f(y) \leq f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \frac{\beta}{2}\|y - x\|^2,$$

for every $y \in \mathcal{X}$. Note that for enough regular functions, smoothness condition can be derived by upper bounding a second order multivariate Taylor expansion using that the Hessian's eigenvalues are upper bounded by $\beta$. It is an easy way to remember the inequality. So we can define the next point as

$$\begin{aligned} x_{k+1} &:= \arg\min_{y \in \mathcal{X}} \left\{ f(x_k) + \langle \nabla f(x_k), y - x_k \rangle + \frac{\beta}{2}\|y - x_k\|^2 \right\} \\ &= \arg\min_{y \in \mathcal{X}} \left\{ \langle \nabla f(x_k), y - x_k \rangle + \frac{\beta}{2}\|y - x_k\|^2 \right\}. \end{aligned} \tag{29}$$

If we take $\|\cdot\|_2$ and $\mathcal{X} = \mathbb{R}^n$ we are searching for the minimizer in a quadratic function $ay^t y + b^t y + c$, $(a \in \mathbb{R}; b, c \in \mathbb{R}^n)$ which is $-\frac{b}{2a}$ or in our case

$$-\frac{\nabla f(x_k) - 2x_k\beta/2}{2\beta/2} = x_k - \frac{1}{\beta}\nabla f(x_k).$$

which matches our previous analysis. Maybe the following, for a general convex set $\mathcal{X}$, is more interesting (we

subtract constant terms inside the arg min's):

$$\arg\min_{y\in\mathcal{X}}\left\{\left\|\left(x_k-\frac{1}{\beta}\nabla f(x_k)\right)-y\right\|^2\right\}\stackrel{?}{=}\arg\min_{y\in\mathcal{X}}\left\{\langle\nabla f(x_k),y-x_k\rangle+\frac{\beta}{2}\|y-x_k\|^2\right\}$$

$$\Leftrightarrow\arg\min_{y\in\mathcal{X}}\left\{\langle y,y\rangle-2\left\langle y,\left(x_k-\frac{1}{\beta}\nabla f(x_k)\right)\right\rangle\right\}\stackrel{?}{=}\arg\min_{y\in\mathcal{X}}\left\{\langle\nabla f(x_k),y\rangle+\frac{\beta}{2}(\langle y,y\rangle-2\langle x_k,y\rangle)\right\}.$$

It is clear that the two arg min's of the last expression are the same, since we can obtain the left hand side by dividing by $\frac{\beta}{2}$ in the arg min of the right hand side. This means that our rule for projected gradient descent computes the point in $\mathcal{X}$ whose decrease guarantee given by the $\beta$-smoothness of $f$ is maximal among all the points in $\mathcal{X}$. Gradient descent for general norms is defined by rule (29). We denote

$$\texttt{Prog}(x):=-\min_{y\in\mathcal{X}}\left\{\langle\nabla f(x),y-x+\frac{\beta}{2}\|y-x\|^2\right\}\geq 0.$$

By the definition of $x_{k+1}$ it is clear that $f(x_{k+1})\leq f(x_k)-\texttt{Prog}(x)$ (and $\texttt{Prog}(x)=\frac{1}{2\beta}\|\nabla f(x)\|_*^2$ if $\mathcal{X}=\mathbb{R}^n$).

In short, we can say that **gradient descent at each iteration maximizes the guaranteed decrease**.

Our second remark was that with $\mathcal{X}=\mathbb{R}^n$ and $\|\cdot\|_2$ the decrease is better if $\|\nabla f(x)\|$ is larger. An intuition that we will formalize later is that mirror descent for $\mathcal{X}=\mathbb{R}^n$ and $\|\cdot\|_2$ suffers from a small loss . In general we will prove that a bound for the mirror descent loss is going to have a term easy to control and something proportional to $\texttt{Prog}(x)$. So when mirror descent suffers from a large loss, gradient descent decreases the objective a lot, and when gradient descent does not have a large guaranteed decrease, mirror descent's loss will be small. This is the key idea of linear coupling, it can be used to obtain a clean accelerated method.

We saw last week that mirror descent tackles the dual optimization problem by constructing lower bounds to the optimum. Recall that each queried gradient $\nabla f(x)$ can be viewed as a hyperplane lower bounding the objective $f$, that is, $f(y)\geq f(x)+\langle\nabla f(x),y-x\rangle$ for all $y$. Mirror-descent methods attempt to carefully construct a convex combination of these hyperplanes in order to yield even a stronger lower bound. From this point of view our claimed intuition about mirror descent having a small loss when $\|\nabla f(x)\|_2$ is small should be clear.

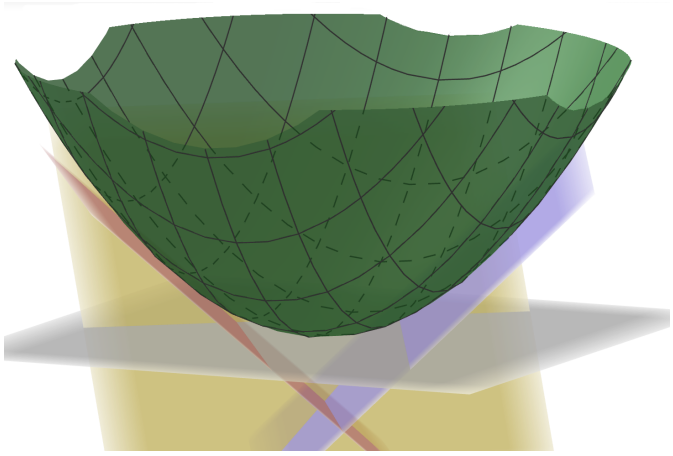

Figure 3: TODO Mirror descent dual loss

It is clear that if we have several hyperplanes lower bounding the epigraph of our function we could pick a point $x$ with a small loss, that is, the difference $f(x)-f(x^*)$ is small. That is because the difference between the optimum

and the maximum of the lower bound is small. Figure 6.1 there is a visualization of this process. In general, instead of picking the best point of the lower bound, which could need the solution of an expensive convex problem, we just take the mean of the points that we used to defined the hyperplanes and that point is good enough for the analysis. Of course this happens because mirror descent does not pick the hyperplanes naively.

**Thought experiment.** For sake of demonstrating the idea, suppose $\|\nabla f(x)\|_2$, the norm of the observed gradient, is **either** always $\geq K$, or always $\geq K$, where $K$ will be determined later. Under such "wishful assumption", we can propose the following algorithm: the norm of the gradient is always $\geq K$ perform $T$ gradient descent steps. Otherwise perform $T$ mirror descent steps. To analyze such an algorithm, suppose without loss of generality we start with some point $x_0$ whose objective distance $f(x_0) - f(x^*)$ is at most $2\varepsilon$, and we want to find some $x$ so that

If $T$ gradient descent steps are performed, the objective decreases by at least $\frac{\|\nabla f(\cdot)\|_2^2}{2L} \geq \frac{K^2}{2L}$ per step and we only need $T \geq \Omega\left(\frac{\varepsilon L}{K^2}\right)$ steps to achieve an $\varepsilon$ accuracy. If $T$ mirror descent steps are performed, we need $T \geq \Omega\left(\frac{K^2}{\varepsilon^2}\right)$ steps according to the mirror descent convergence. In sum, we need $T \geq \Omega\left(\max\left\{\frac{\varepsilon L}{K^2}, \frac{K^2}{\varepsilon^2}\right\}\right)$ steps to converge to an $\varepsilon$-minimizer. Setting K to be the magic number to balance the two terms, we only need $T \geq \Omega\left(\sqrt{\frac{L}{\varepsilon}}\right)$ iterations. This means that in the general case in which $f(x_0) - f(x^*) \leq d$ we only need $T \geq \Omega\left(\sqrt{\frac{L}{\varepsilon}}\left(\frac{\varepsilon}{d} + \frac{2\varepsilon}{d} + \cdots + \frac{1}{2} + 1\right)\right) = \Omega\left(\sqrt{\frac{L}{\varepsilon}}\right)$.

## 6.2 Warm-Up Method with Fixed Step Length

TODO

## 6.3 Final Method with Variable Step Lengths

TODO

# 7 Non-Euclidean setting: Frank-Wolfe

# 8 Stochastic oracle model

Assume that we want to solve the following problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{X}, \end{array} \tag{30}$$

where $f$ is a convex function, possibly not known. The first order *stochastic* oracle model defines the framework where given $x \in \mathcal{X}$ an oracle yields back a random variable $G$ that is an unbiased estimator of a subgradient of $f$ at $x$, namely, $\mathbf{E}[G] \in \partial f(x)$. If $X$ is a random variable, the oracle yields back a random variable $G$ that is an unbiased estimator of a subgradient of $f$ at $X$ conditionally on $X$, namely, $\mathbf{E}[G|X] \in \partial f(X)$.

### $L$-Lipschitz functions

We analyze the behavior of the projected gradient descent algorithm to solve problem (30) when we replace exact knowledge of subgradients of $f$ with unbiased estimates of them. For a given initial point $X_1 \in \mathcal{X}$, possibly random, and a given collection of step sizes $\eta_1, \eta_2, \ldots$, the stochastic projected gradient descent is defined by the sequence of random variables generated according to the following update:

$$\begin{aligned} Y_{s+1} &= X_s - \eta_s G_s, \\ X_{s+1} &= \Pi_{\mathcal{X}}(Y_{s+1}). \end{aligned}$$

Here, conditionally on $X_s$, the random variable $G_s$ is an unbiased estimator of the subgradient of the function $f$ at $X_s$, namely,

$$\mathbf{E}[G_s|X_s] \in \partial f(X_s).$$

We state and prove the convergence of this method in the $L$-Lipschitz case.

**Theorem 6** ($L$-Lipschitz Stochastic Projected Subgradient Descent)**.** *Let $f$ be convex. Suppose $\mathcal{X}$ is contained in a Euclidean ball of radius $B$ centred at $X_1$. Suppose that for every $X \in \mathcal{X}$ the stochastic oracle yields an unbiased estimator of the subgradient of $f$ at $X$ bounded in $L_2$, namely, $\mathbf{E}[G|X] \in \partial f(X)$ and $\mathbf{E}[\|G\|^2] \le L^2$. Then, the projected subgradient descent method with $\eta_s \equiv \eta = \frac{B}{L\sqrt{t}}$ satisfies*

$$\mathbf{E} f\left(\frac{1}{t}\sum_{s=1}^{t} X_s\right) - f(x^*) \le \frac{LB}{\sqrt{t}}. \tag{31}$$

*Proof.* For any $1 \le s \le t$ we have

$$f(X_s) - f(x^*) \le \mathbf{E}[G_s|X_s]^T (X_s - x^*) = \mathbf{E}[G_s^T (X_s - x^*)|X_s].$$

Proceeding as in the proof of Theorem 2, we find

$$G_s^T (X_s - x^*) \le \frac{1}{2\eta}\left(\|X_s - x^*\|^2 - \|X_{s+1} - x^*\|^2\right) + \frac{\eta}{2}\|G_s\|^2.$$

Taking the expectation, we get

$$\mathbf{E} f(X_s) - f(x^*) \le \mathbf{E}[G_s^T (X_s - x^*)] \le \frac{1}{2\eta}\left(\mathbf{E}\|X_s - x^*\|^2 - \mathbf{E}\|X_{s+1} - x^*\|^2\right) + \frac{\eta}{2}\mathbf{E}[\|G_s\|^2],$$

and using the assumption $\mathbf{E}[\|G_s\|^2] \le L^2$ we get

$$\frac{1}{t}\sum_{s=1}^{t}(\mathbf{E} f(X_s) - f(x^*)) \le \frac{1}{2\eta t}\left(\mathbf{E}\|X_1 - x^*\|^2 - \mathbf{E}\|X_{t+1} - x^*\|^2\right) + \frac{\eta}{2}L^2 \le \frac{B^2}{2\eta t} + \frac{\eta L^2}{2}.$$

Selecting $\eta = \frac{B}{L\sqrt{t}}$ to minimize the right-hand side of the above inequality gives the first result in (9) (since $f\left(\frac{1}{t}\sum_{s=1}^{t} X_s\right) \leq \frac{1}{t}\sum_{s=1}^{t} f(X_s)$ by Jensen's inequality). □

Theorem 6 shows that, in expectation, the stochastic projected subgradient descent method yields the same convergence guarantees as the deterministic counterpart analyzed in Theorem 2. In particular, the oracle complexity is the same[3]: to get an accuracy $\epsilon$, both methods requires $O(1/\epsilon^2)$ calls to their respective oracles. The main advantage of the stochastic version lies in the fact that in some applications the *computational* complexity involved in having access to a stochastic oracle is much cheaper than in the determinist case. We now show that the stochastic model yields substantial computational saving in machine learning.

## 8.1 Multiple passes over the data

TO DO.

## 8.2 Single pass over the data

We now show how the first order stochastic oracle model allows us to deal directly with expected risk minimization without invoking the empirical risk minimization paradigm. Let us recall the original problem that motivates us in Section 1, namely, the expected risk minimization:

$$\begin{array}{ll} \text{minimize} & r(x) = \mathbf{E}\ell(x^T\Phi(W), Y) \\ \text{subject to} & x \in \mathcal{X}. \end{array}$$

The assumption here is that we know the loss function $\ell$ (indeed, we can choose it!) and the constraint set $\mathcal{X}$, but we do not know the distribution of $(W, Y)$. We only have access to $m$ i.i.d. samples $(W_1, Y_1), \ldots, (W_m, Y_m)$ from this unknown distribution. So we do not know the function $r$ that we want to minimize, and in particular we can not operate in the deterministic first order oracle model discussed in the previous weeks as for a given $x \in \mathcal{X}$ we do not have access to a subgradient in $\partial r(x)$. On the other hand, given $x \in \mathcal{X}$ we can have access to an unbiased estimator of a subgradient of $r$ evaluated at $x$. In fact, given the function $x \to R_i(x) := \ell(x^T\Phi(W_i), Y_i)$, which is known to us, we can compute a subgradient of $R_i$ at $x$. This is a random variable $G_i \in \partial R_i(x)$ that satisfies $\mathbf{E}G_i \in \partial r(x)$. The same is true if we want to get an estimate of the subgradient of $r$ at $X \in \mathcal{X}$ when $X$ is a random variable *independent* of $(W_i, Y_i)$. In fact, in this case we can evaluate a subgradient of $R_i$ at $X$, and this random variable $G_i \in \partial R_i(X)$ satisfies $\mathbf{E}[G_i|X] \in r(X)$.[4] Hence, we satisfy the assumption of the first order stochastic oracle model. At the same time, as we have $m$ independent data points at out disposal, we can have access to at most $m$ independent unbiased estimators of subgradients evaluated at possibly different locations in $\mathcal{X}$. In other words, the requirement of independence restricts us to a *single pass* over the data, which is not as general as in the stochastic block model defined above where we can have how many queries to the oracle as we want.

It is easy to check that under the assumptions of Proposition 4, Theorem 6 yields the following convergence guarantees for the stochastic projected descent method:

$$\mathbf{E}r(\hat{X}_m) - r(x^*) \leq \frac{GL_{\text{loss}}B}{\sqrt{m}},$$

with $\hat{X}_m = \frac{1}{m}\sum_{i=1}^{m} X_i$. In other word, we can apply the stochastic gradient descent algorithm to *directly* minimize the expected risk $r$, without invoking the empirical risk minimization paradigm, i.e., without breaking the problem into statistics and optimization as outlined in Section 1. The computational savings are clear. If computing a

[3]Note that different notions of accuracy are used, however, as we consider the expected value of the stochastic method.

[4]Note that if $X$ is random, then there are two sources of randomness in $G_i$: one source is $X$ itself, the other is the data point $(W_i, Y_i)$. The statement $\mathbf{E}[G_i|X] \in r(X)$ holds if $X$ and $(W_i, Y_i)$ are independent.

subgradient for each functions $R_i$ costs $O(1)$, then the computational complexity of stochastic gradient descent to achieve precision $\epsilon$ (in expectation) is of order $O(1/\epsilon^2)$. On the other hand, if we apply the deterministic gradient descent method to minimize the empirical risk $R$ up to precision $1/\sqrt{m}$, as discussed in Section 1 and in Remark 2, then we see that the computational complexity is $O(m/\epsilon^2)$, as we need $O(1/\epsilon^2)$ calls to the deterministic oracle but each call costs $O(m)$ base iterations, as $R(x) := \frac{1}{m}\sum_{i=1}^{m} R_i(x)$.

# References


[1] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.

[2] Francis Bach. Statistical machine learning and convex optimization.

[3] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[4] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[5] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2).

[6] Philippe Rigollet. 18.657 mathematics of machine learning. fall 2015. 2015.