MS&E 213 / CS 269O : Chapter 8 - Online Linear Optimization, Subgradient Descent, and Mirror Descent *

By Aaron Sidford (sidford@stanford.edu)

June 7, 2017

1 Motivation

In the last section we saw how to use separation oracles for convex function and sets to compute ϵ-suboptimal points for a variety of points. These algorithms had a very good dependence on the desired error level $\epsilon > 0$, but left a bit to be desired in terms of their dependence on the dimension of the problem d. They all had a polylogarithmic dependence on ϵ but a dependence on d that was at least $\Omega(d)$.

This was rather inherent in the abstraction we used for using separation oracles and the assumptions we made about the problem. In other words, we saw that the query complexity of $O(d \log \epsilon^{-1})$ we achieved was optimal for the feasibility problem we considered and that to do better different assumptions need to be made.

Here we show how to achieve a different dimension versus error trade-off for non-smooth convex optimization while making a slightly different set of assumptions. Our emphasis in this chapter is on performing convex optimization with a *subgradient oracle*. Our abstraction for using a subgradient oracle is the *online linear optimization* problem, a fundamental problem in learning and online algorithms. We show how to solve this problem efficiently and achieve efficient algorithms not just for subgradient descent, but also the learning from experts problem and more.

2 Subgradients

We start by formally defining *subgradients*. As we have discussed these are close analogs gradients of convex functions and when we minimize convex functions that are possibly non-differentiable, this is what we will assume we can compute in this chapter.

Definition 1 (Subgradient)**.** For a function $f : \mathbb{R}^n \to \mathbb{R}$ and $x \in \mathbb{R}^n$ we call g a *subgradient* of f at x if for all $y \in \mathbb{R}^n$ we have the property that

$$f(y) \geq f(x) + g^\top (y - x)$$

and we let $\partial f(x)$ denote the set of *subgradients* of f at x.

In the remainder of this section we prove various properties of subgradients and their connections to convex functions more broadly.

*These notes are a work in progress. They are not necessarily a subset or superset of the in-class material and there may also be occasional *TODO* comments which demarcate material I am thinking of adding in the future. These notes will converge to a superset of the class material that is TODO-free. Your feedback is welcome and highly encouraged. If anything is unclear, you find a bug or typo, or if you would find it particularly helpful for anything to be expanded upon, please do not hesitate to post a question on the discussion board or contact me directly at sidford@stanford.edu.

First we show that if a function has a subgradient at every point, then the function is convex.

Lemma 2. *Suppose for convex $S \subseteq \mathbb{R}^n$ and $f : S \to \mathbb{R}$ it is the case that $\partial f(x) \neq \emptyset$ for all $x \in S$ then f is convex.*

Proof. Let $x, y \in S$ and $t \in [0,1]$ be arbitrary. Let $x_t \stackrel{\text{def}}{=} t \cdot y + (1-t) \cdot x$ and $g \in \partial f(x)$, as $x_t \in S$ by convexity. By definition of $\partial f(x)$ we have that

$$f(y) \geq f(x_t) + g^\top (y - x_t) = f(x_t) + (1-t) \cdot g^\top (y - x)$$

and

$$f(x) \geq f(x_t) + g^\top (x - x_t) = f(x_t) + t \cdot g^\top (x - y)$$

adding a t multiple of the first inequality to a $(1-t)$ multiple of the second inequality yields that

$$t \cdot f(y) + (1-t) \cdot f(x) \geq f(x_t)\,.$$

□

Next we prove that a convex function has a subgradient at every point in the interior of the domain.

Definition 3 (Interior)**.** For $S \subseteq \mathbb{R}^n$ we denote the interior of S by $\text{int}(S) = S \setminus \partial S$ that is, the set of points in S that are not reachable by a convergent sequence of points not in S.

Note that equivalently we can view the interior of a convex set as a point in the set such that the set contains a ball centered around the point.

Lemma 4. *Let $S \subseteq \mathbb{R}^n$ be a convex set and $f : S \to \mathbb{R}$ be convex function. Then $\partial f(x) \neq \emptyset$ for all $x \in \text{int}(S)$.*

Proof. Let $x \in \text{int}(S)$ be arbitrary. Clearly $(x, f(x)) \in \partial \text{epi}(f)$ since $(x, f(x)+t)$ is in $\text{epi}(f)$ whenever $t \leq 0$ and is not in $\text{epi}(f)$ whenever $t \geq 0$. Consequently, from the supporting hyperplane theorem we know that there is a vector $g \in \mathbb{R}^n$ and $v_g \in \mathbb{R}$ with either $g \neq 0$ or $v \neq 0$ such that for all $(y, v_y) \in \text{epi}(f)$ we have that

$$g^\top y + v_g \cdot v_y \geq g^\top x + v_g \cdot f(x)\,.$$

However, since $(y, v_y + \alpha) \in \text{epi}(f)$ for all α we see that $v_g \geq 0$ since otherwise we could make the above inequality false for large enough α.
Now, since $(y, f(y)) \in \text{epi}(f)$ this implies

$$v_g \cdot (f(y) - f(x)) \geq (-g)^\top (y - x)\,.$$

However, since $x \in \text{int}(S)$ we know that $x - \epsilon g \in \text{int}(S)$ for small enough $\epsilon > 0$ and therefore

$$v_g \cdot (f(y) - f(x)) \geq \epsilon \|g\|_2^2$$

Now either $g = \vec{0}$ in which case $v_g \neq 0$ or $g \neq 0$ in which case $\epsilon \|g\|_2^2 > 0$ in which case $v_g \neq 0$. Consequently, $v_g > 0$ and $f(y) \geq f(x) + (-g/v_g)^\top (y - x)$ for all $y \in S$ and we have that $(-g/v_g)$ is a subgradient of f. □

Finally, we conclude this section by showing that subgradients of differentiable functions must be their subgradients.

Lemma 5. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable at $x \in \mathbb{R}^n$ and $g \in \partial f(x)$, then $g = \bigtriangledown f(x)$..*

Proof. Homework.

□

3 Online Linear Optimization

Now that we know what subgradients are, we provide our abstraction for how we will use having a subgradient oracle to perform minimization of smooth funcitons. Our abstraction is a problem known as *online linear optimization problem*. It is more general then the problem of subgradient descent and we define it formally as follows.

Definition 6 (Online Linear Optimization Problem)**.** The *online linear optimization problem* is as follows: we are given a closed convex set $S \subseteq \mathbb{R}^n$ and in each round $t \geq 1$ we pick some $x_t \in S$. We are then revealed a penalty $p_t \in K$. We wish to minimize *regret* defined as follows

$$\text{regret}(T) \stackrel{\text{def}}{=} \sum_{t \in [T]} p_t^\top x_t - \min_{x \in S} \sum_{t \in [T]} p_t^\top x \,.$$

The idea behind this game is that S should be though of as some sort of space of actions we are allowed to take and $p_t^\top x_t$ is to be though of as the penalty we incur for picking that action. Now, we wish to minimize our total penalty, $p_t^\top x_t$, as compared to the minimal penalty we could have incurred if we had just fixed an action $x \in S$ and stuck with it, throughout the game. We call this *regret*, as it captures how much we might regret simply picking a fixed strategy from S.

Note, that there are other notions of regret we might pick or different things we wish to optimize in such an online linear optimization problem. This notion of regret was chosen as it directly maps to subgradient descent and a variety of problems. It is one of the simplest problems in online optimization and learning, yet still highly nontrivial to solve.

Now, as we will see our goal will ultimately be to provide algorithms so that $\lim_{R \to \infty} \frac{\text{regret}(T)}{T} = 0$, i.e. the average regret over the rounds of the game converges to 0. This seems highly difficult to achieve though as in general the penalty functions p_t are allowed to depend on anything (including the x_t themselves).

Also, note that regret need not always be positive. Since x_t may change between rounds in principle it is possibly that $\sum_{t \in [T]} p_t^\top x_t$ is less than $\min_{x \in S} \sum_{t \in [T]} p_t^\top x$. While this is suggestive of picking a regret measure like $\sum_{t \in [T]} p_t^\top x_t - \min_{x_1^*, \ldots, x_T^* \in S} \sum_{t \in [T]} p_t^\top (x_t^*)$, as we will see in the homework this in general impossible to have low average regret for.

4 Reduction to Convex Optimization

Before we discuss algorithm for solving the linear optimization problem, let's quickly see why this problem encompasses convex optimization with a subgradient oracle. The idea is quite simple, if we pick the penalty functions to be subgradients, then we can show that a low regret algorithm for the online linear optimization problem produces an ϵ-suboptimal point. First, we formally define a subgradient oracle and then we show this claim.

Definition 7 (Subgradient Oracle)**.** For $f : \mathbb{R}^n \to \mathbb{R}$ as *subgradient oracle* is an oracle that given query point $x \in \mathbb{R}^n$ returns $g_x \in \partial f(x)$.

In the next lemma we show that a solution to the online linear optimization problem plus a subgradient oracle can be used to comptue ϵ-suboptimal points for constrained convex function mnimization.

Lemma 8. *For closed convex set S and convex $f : \mathbb{R}^n \to \mathbb{R}$ with minimizer $x_* \in S$ if we have a solution to the online linear optimization problem that achieves some regret,* $\text{regret}(T)$*, when for the sequence of points $x_1, \ldots, x_T$ we have that $p_t \in \partial f(x_t)$ then the point $\frac{1}{T} \sum_{t \in [T]} x_t$ is* $\text{regret}(T)/T$*-suboptimal.*

Proof. Note that by definition of $\partial f(x_t)$ it is the case that $f(x_*) \geq f(x_t) + p_t^\top (x_* - x_t)$. Consequently, by convexity

$$f\left(\frac{1}{T}\sum_{t\in[T]} x_t\right) - f_* \leq \frac{1}{T}\sum_{t\in[T]} f(x_t) - f_* \leq \frac{1}{T}\sum_{t\in[T]} p_t^\top (x_t - x_*)$$
$$\leq \frac{1}{T}\left[\sum_{t\in T} p_t^\top x_t - \min_{x\in S} p_t^\top x\right] = \frac{\text{regret}(T)}{T}$$

□

Consequently, this shows that if we have a solution to the online linear optimization problem where average regret converges to 0 then this yields an algorithm for computing ϵ-suboptoimal points with a subgradient oracle. Moreover, this shows that the exact query complexity of the problem depends on the rate at which average regret goes to 0.

5 Learning From Experts

Before we give our algorithm for solving the online linear optimization problem, as a warm-up to gain some intuition about the problem, let us consider a particular special case of the online linear optimization problem.

Definition 9 (Learning From Experts Problem)**.** The *learning from experts problem* is the online learning problem where $S = \Delta^n = \{x \in \mathbb{R}^n_{\geq 0} | \sum_{i\in[n]} x_i = 1\}$ and for all $t \geq 0$ and $i \in [n]$ we have $p_t(i) \in [0, 1]$.

The set $S = \Delta^n$ is known as the simplex. It is the set of non-negative vectors whose coordinates sum to 1. The set can be though of the set of probability vectors on n items. There is a natural bijection between Δ^n and the marginals of a distribution on n items. For every distribution on n items if we let p_i be the probability of picking item i then $p \in \Delta^n$. Conversely, for $x \in \Delta^n$ we let $\mathcal{D}_x$ be the simple distrubition on $[n]$ where if $\Pr_{i\sim\mathcal{D}_x}[i = j] = x_j$. With this interpretation in mind, if we think of the penalty in a round of the learning from experts problem we have that

$$p_t^\top x_t = \sum_{i\in[n]} p_t(i) \cdot x_t(i) = \mathbb{E}_{i\sim\mathcal{D}_{x_t}} p_t(i)$$

in other words, $p_t^\top x_t$ is the expected value of the i-th coordinate of p_t as sampled by the distribution induced by x_t.

Consequently, the learning from experts problem has the following nice interpretation. We can think there are n experts corresponding to each of the coordinates. We think that each day the experts are making some sort of prediction or giving some sort of advice and we would like to decide who to listen to. Maybe they are all predicting whether or not it will rain or whether the stock market will go up or down and $p_t(i) = 1$ means they were incorrect and $p_t(i) = 0$ means they were incorrect. Now, in each day we pick a distribution over the experts corresponding to the distribution over which expert we will listen to, i.e. $x_t \in \Delta^n$. Then at the end of the day, it is revealed how well each of the experts do $p_t \in \mathbb{R}^n$ with $p_t(i)$ large means did badly (large penalty) and $p_t(i)$ small means they did well (small penalty).

Now, $\sum_{t\in[T]} x_t^\top p_t$ is how well we do by following our strategy. It is the expected penalty we incur from picking experts as we did. However, $\min_{x\in\Delta^n} \sum_{i\in[n]} p_t^\top x = \min_{i\in[n]} \sum_{i\in[n]} p_t(i)$, i.e. the penalty incurred by the expert with minimum penalty. Consequently, regret is the difference between our total penalty and the penalty incurred by the best expert. Thus, achieving low average regret, means that we can do as well as the best expert does on average.

However, this is a highly non-trivial claim. Note that we are allowing the performance of the experts to depend on the distribution we pick. If we overcommit to listening to an expert, he might cease to perform well. However, if we pick a uniform distribution, then we might miss out on an expert performing particularly well. This learning from experts problem is about how to trade off these two issues while dealing with the fact that future performance of the experts might have nothing to do with their past performance. In the next section we take a closer look at some of these simple algorithms to help derive the actual algorithm that we will use.

6 Simple Learning from Expert Algorithms

Here we consider some simple algorithms to solve the learning from experts problem. We consider these algorithms to better understand the problem and ultimately motivate the efficient algorithms we consider for this problem.

6.1 Arbitrary

To start, to get an upper bound on performance. Let us consider the the penalty and possible regret of an arbitrary algorithm. Note that since $p_t(i) \in [0,1]$ and $x_t(i) \geq 0$ we have that

$$p_t^\top x_t = \sum_{i \in [n]} p_t(i) \cdot x_t(i) \leq \sum_{i \in [n]} x_t(i) = 1$$

Cnsequently, $\sum_{t \in [T]} p_t^\top x_t \leq T$. Furthermore, for any $x \in \Delta^n$ since $x \geq \vec{0}$ coordinate-wise and $p \geq \vec{0}$ coordinate wise we have

$$\min_{x \in \Delta} \sum_{t \in [T]} p_t^\top x \geq \sum_{t \in [T]} 0 = 0$$

and therefore, for any algorithm for solving the learning from experts problem we have $\text{regret}_T(T) \leq T$. Thus we can ensure average regret 1 easily, however getting an algorithm so average regret converges to 1 seems to require more work.

Note that it can actually be the case that $\text{regret}_T(T) = 1$. If we just have two experts and $p_t(1) = 1$ and $p_t(2) = 0$ for all $t \in [T]$ and we keep picking $x_t = \vec{1}_1$, i.e. we keep listening to the first expert, then we will get regret that is exactly t.

6.2 Random

Another simple algorithm to consider is a random strategy. We could simply pick $x_t = \frac{1}{n}\vec{1}$ for all $t \in [T]$. Note that for this strategy we have that if $j = \text{argmin}_{i \in [n]} \sum_{t \in [R]} p_t(j)$ then

$$\begin{aligned}
\text{regret}(T) &= \sum_{t \in [T]} p_t^\top x_t - \min_{x \in \Delta^n} \sum_{t \in [T]} p_t^\top x = \sum_{t \in [T]} \left[\sum_{i \in [n]} \frac{1}{n} p_t(i) - p_t(j) \right] \\
&\leq \sum_{t \in [T]} \sum_{j \neq i} \frac{1}{n} = T \cdot \frac{n-1}{n} = T \cdot \left(1 - \frac{1}{n}\right).
\end{aligned}$$

Furethermore, we can show that this is tight in the worst case. If $x_t(1) = 0$ for all t and $x_t(i) = 1$ for all $i \neq 1$ then this is precisely the regret that will be achieved.

Thus, we can show that random does a little better than arbitrary, but how much better degrades as n increases and the average regret still stays at a constant when $n \geq 2$.

6.3 Follow the Leader

Each of the strategies we have seen so far have been oblivious to the actual penalties, p_t encountered. If we wish to do better the above analysis suggest we need to some how be focus in on experts doing better than the average. Another natural strategy follow to achieve this is known as *follow the leader (FTL)*. In each round t we can pick the strategy for which would have done the best on what we have observed in the past. In other words, we could let $x_{t+1} = \text{argmin}_{x \in \Delta^n} \sum_{k \in [t]} p_k^\top x$.

However, we can show that this strategy gets regret that converges to $T/2$ in the worst case. To see this, consider a simple scenario where we have two experts. In the first round, expert 1 gets penalty ϵ and expert 2 gets penalty 0. In the next round we pick expert 2 to listen to as they have done better in the past. However, expert 2 gets penalty 1 and expert 1 gets penalty 0. Now expert 1 has had total penalty ϵ and expert 2 has had total penalty 1 so we pick expert 1. However, we then give expert 1 penalty 1 and expert 2 penalty 0. We continue in this way, penalizing only the expert we pick. Note that our total penalty after T rounds is $\epsilon + (T-1)$. However, each of the experts individually have had total penalty at most $\lceil \frac{T}{2} \rceil$. Consequently, our regret converges to $T/2$.

In short, this strategy perform poorly as even though both experts have done about the same over time, by us switching experts every round, we realize a penalty that is much worse. Note that we can make the factor of 2 even worse by considering a similar counterexample with more experts.

7 Follow The Regularized Leader

In the previous section we saw several algorithms for the learning from experts problem. A random strategy failed as it didn't identify an expert performing much better than the others. However, the greedy follow the leader strategy also failed as it kept changing strategy missing out on the true leader. At first glance, this seems like an insurmountable problem, we want to learn what experts to listen to, however past performance says nothing about future performance and thus if we pick an expert, it might do worse down the road. How should we hope to surmount this?

One idea is that if we start to focus on an expert based on past performance, then if the expert ceases to do well, then there was likely not any strategy that does well. In our example where follow the leader does poorly, uniform does great, and in our example where uniform did poorly, greedy did great. Perhaps, the right algorithm is one that carefully interpolates between these two algorithms.

This turns out to be the case and is precisely the algorithm we will show in the case. A random strategy can be though of picking $x_t = \text{argmin}_{x \in S} r(x)$ for some function r that depends only on S (not on the p_t) and forced the algorithm to pick $x_t = \frac{1}{n}\vec{1}$ every time. Here we consider an algorithm that interpolates between this algorithm and greedy. It is called *follow the regularized leader* and the term $r(x)$ is called a *regularizer*.

Definition 10 (Follow The Regularized Leader (FTRL))**.** To solve the *online linear optimization problem* the *follow the regularized leader* (FTRL) algorithm is as follows, pick some function $r : \mathbb{R}^n \to \mathbb{R}$, called a *regularizer,* that is differentiable and μ-strongly convex on S with respect to some norm $\|\cdot\|$. For all $T \geq 0$ let

$$\Phi_T(x) \stackrel{\text{def}}{=} \eta \sum_{t \in [T]} p_t^\top x + r(x)$$

and (with $[0] \stackrel{\text{def}}{=} \emptyset$ so $\Phi_0(x) \stackrel{\text{def}}{=} r(x)$) and for all $T \geq 0$ pick [1]

$$x_{T+1} \stackrel{\text{def}}{=} \text{argmin}_{x \in S} \Phi_T(x)\,.$$

This algorithm is quite simple, simply introduce a parameter η (sometimes called a learning rate) that trades off how much we care about optimizing past performance, i.e. minimizing $\sum_{t \in [T]} p_t^\top x$ and being

[1] Where exactly do we use the convexity of S? (It is in the minimality.)

stable, i.e. minimizing $r(x)$. The algorithm simply takes the next action to be the minimizer over S of the η weighted combination of these two contributions. The idea is that by picking a well behaved strongly convex regularizer, we might ensure that we do not over commit to well performing experts while still be able to take advantage of their improved performance.

In the remainder of this section we show that this algorithm achieves average regret that converges to 0 in many scenarios. We show this by analyzing how the minimizer of Φ_T changes in each iteration.

First, we recall the following basic lemma we proved earlier for this analysis.

Lemma 11 ((Recap) Characterization of Optimizer)**.** *If $f : \mathbb{R}^n \to \mathbb{R}$ is a differentiable convex function and $S \subseteq \mathbb{R}^n$ is a closed convex set, then x_* is a minimizer of f over S, i.e. $x_* \in S$, i.e. $f(x_*) \leq f(x)$ for all $x \in S$, if and only if $\bigtriangledown f(x_*)^\top (x - x_*) \geq 0$ for all $x \in S$.*

Using this we show that we bound how much Φ_T changes as we move away from its minimizer.

Lemma 12. *In an instance of FTRL for all $z \in S$ and $t \geq 0$ we have*

$$\Phi_t(z) \geq \Phi_t(x_{t+1}) + \frac{\mu}{2}\|z - x_{t+1}\|^2 .$$

Proof. Since $r(z)$ is μ-strongly convex and linear functions, e.g. $\eta \sum_{k\in[t]} p_k^\top z$ is 0-strongly convex, and the sum of a -strongly convex function and a 0-strongly convex function is μ-strongly convex we have that Φ_t is μ-strongly convex. Consequently, as r is differentiable (and therefore Φ is differentiable), we have

$$\Phi_t(z) \geq \Phi_t(x_{t+1}) + \bigtriangledown \Phi_t(x_{t+1})^\top (z - x_{t+1}) + \frac{\mu}{2}\|z - x_{Tt1}\|^2 .$$

However, since $x_{t+1} = \text{argmin}_{x\in S}\Phi_t(x)$ we have that $\bigtriangledown \Phi_t(x_{t+1})^\top (z - x_{t+1}) \geq 0$ by Lemma 11 and the result follows. □

Using this we show bound how much Φ_T increases in each iteration. Often the difficult part of analysis in online algorithms is showing that the optimal, e.g. $\min_{x\in S} \sum_{t\in[T]} p_t^\top x$ is lower bounded by something involving our performance. Since, the minimizer of Φ_T is approximately this quantity (up to scaling by η and addition of the regularizer), lower bounding Φ_T serves this purpose.

Lemma 13. *In FTRL for all $t \geq 1$ we have that*

$$\begin{aligned}\Phi_t(x_{t+1}) - \Phi_{t-1}(x_t) &\geq \eta \cdot p_t^\top x_t + \eta \cdot p_t^\top (x_{t+1} - x_t) + \frac{\mu}{2}\|x_t - x_{t+1}\|^2 \\ &\geq \eta \cdot p_t^\top x_t - \frac{\eta^2}{2\mu}\|p_t\|_*^2\end{aligned}$$

Proof. First, note that by the definition of Φ_t and Lemma 12 we have

$$\Phi_t(x_{t+1}) = p_t^\top x_{t+1} + \Phi_{t-1}(x_{t+1}) \geq p_t^\top x_{t+1} + \frac{\mu}{2}\|x_t - x_{t+1}\|^2 + \Phi_{t-1}(x_t) .$$

Rearranging terms yields the first inequality and using that

$$\eta \cdot p_t^\top z + \frac{\mu}{2}\|z\|^2 \geq -\frac{\eta^2}{2\mu}\|p_t\|_*^2$$

yields the second inequality. □

This lemma shows that in every iteration the minimizer of Φ_t increases by an η multiple of our realized penalty, $p_t^\top x_t$, minus η^2times $\frac{1}{2\mu}\|p_t\|_*^2$. Summing this up over each round lets us immediately bound the total regret.

Theorem 14 (FTRL Guarantee)**.** *If we implement FTRL for the online linear optimization problem and and $G \geq \|p_t\|_*$ for all $t \in [T]$ then for all $z \in S$*

$$\sum_{t\in[T]} p_t^\top (x_t - z) \leq \frac{\eta}{2} \cdot T \cdot G^2 + \frac{1}{\eta} \cdot \left[r(z) - \min_{x\in S} r(x) \right]$$

and consequently for $D \geq \max_{x\in S} r(x) - \min_{x\in S} r(x)$ and $\eta = \sqrt{(2D)/(TG^2)}$ we can achieve

$$\text{regret}(T) \leq \sqrt{2DTG^2}\,.$$

Proof. Summing Lemma 13 yields

$$\Phi_T(x_{T+1}) - \Phi_0(x_1) = \sum_{t\in[T]} \left[\Phi_t(x_{t+1}) - \Phi_{t-1}(x_t)\right] \geq \sum_{t\in[T]} \left[\eta \cdot p_t^\top x_t - \frac{\eta^2}{2\mu} \|p_t\|_*^2 \right].$$

Now clearly $\Phi_T(z) \geq \Phi_T(x_{T+1})$ and since $\|p_t\|_* \leq G$ and $\Phi_0(x_1) = \min_{x\in S} r(x)$ we have

$$\eta \sum_{t\in[T]} p_t^\top z + \left[r(z) - \min_{x\in S} f(x) \right] \geq \eta \sum_{t\in[T]} p_t^\top x_t - \frac{\eta^2 T \cdot G^2}{2\mu}\,.$$

Rearranging terms yields the first claim and the choice of η yields the second. □

Note that the analysis essentially says that each iteration we have that our regret grows by at most $\frac{\eta}{2\mu}\|p_t\|_*^2$ and there is a $\frac{1}{\eta}$ contribution from the regularizer. Trading off the two gives the final bound. Note that the rate gets worse as either D (how much the regularizer varies) or G (how large the subgradients are) increases. Also note, that the steps size or learning rate, η gets smaller as T increases, D decreases, or G increases. This says that as we run for longer, i.e. T increases, each step moves away from the regularizer less, i.e. we require great confidence in an experts performance to move towards them. It also says that the large the penalties the more cautious we should be (i.e. we should move away from the regularizer more slowly) and the large the regularizer the more we should move away from it (i.e. as it is less helpful and there is more space to explore).

Note also that this bound says we can achieve

$$\frac{\text{regret}(T)}{T} = \frac{\sqrt{2DTG^2}}{T} = \sqrt{\frac{2DG^2}{T}}\,.$$

Consequently as $T \to \infty$ we have $\text{regret}(T)/T \to 0$. Consequently, we can use this algorithm for our applications. We discuss using FTRL for the problems we considered more extensively in the next few sections.

8 Solving The Experts Problem

Here we show how to use FTRL to solve the learning from experts problem. Note that this is not immediate as FTRL does not specify what regularizer to use or how to compute $x_{t+1} = \text{argmin}_{x\in S}\Phi_t(x)$. Here we discuss these issues for the learning from experts problem.

Our main question is what regularizer r do we want to use. However, even more fundamental perhaps, we could ask what norm do we want to do our analysis in? Note that FTRL depends on the bounds on $\|p_t\|_*$ and we have that $\|p_t\|_\infty \leq 1$. Since $\|\cdot\|_\infty$ is the dual of $\|\cdot\|_1$ (see homework) this would suggest that we may want to do our analysis in $\|\cdot\|_1$. Furthermore, note that since r is strongly convex in $\|\cdot\|$ Lemma 12

shows that if there is some $x \in S$ such that $\|x - x_1\|$ is large then D will be large. However, since we know that $\|x - y\|_1 \leq 2$ for all $x, y \in \Delta$ (by triangle inequality) this would again suggest that we may want to do our analysis in ℓ_1 to keep D down as well.

Consequently, we will perform our analysis for the experts problem in the $\|\cdot\|_1$ norm. The next question is what regularizer to use or equivalently what is a good strongly convex function with respect to $\|\cdot\|_1$ over Δ_n. Another, essentially equivalent question, is what is a good way to measure distances in the simplex? A natural choice here is entropy.

Definition 15. For $x \in \mathbb{R}^n_{>0}$ the *entropy regularizer* is define as $e(x) = \sum_{i\in[n]} x_i \cdot \log x_i$ where we let $0 \log 0 \stackrel{\text{def}}{=} 0$.

We note that this choice of treatment of $0 \log 0$ makes sense due to the following.

Lemma 16. *Note that* $\lim_{x\to 0} x \cdot \log x = 0$.

Proof. By L'Hopital we have

$$\lim_{x\to 0} x \cdot \log x = \lim_{x\to 0} \frac{\log x}{(1/x)} = \lim_{x\to 0} \frac{1/x}{-1/x^2} = \lim_{x\to 0} -x = 0\,.$$

□

Next, we analyze the strong convexity of $e(x)$ with respect to $\|\cdot\|_1$.

Lemma 17. *The function* $e(x) \stackrel{\text{def}}{=} \sum_{i\in[n]} x_i \cdot \log x_i$ *is 1-strongly convex with respect to* $\|\cdot\|_1$ *on* Δ^n.

Proof. Note that $\bigtriangledown r(x)_i = 1 + \log x_i$ and $\bigtriangledown^2 r(x) = \mathbf{X}^{-1}$ where $\mathbf{X} = \mathbf{diag}(x)$. Now for $x \in \Delta^n$ we have that

$$\|y\|_1^2 = \left(\sum_{i\in[n]} \sqrt{x_i} \cdot \frac{|y_i|}{\sqrt{x_i}}\right)^2 \leq \sum_{i\in[n]} x_i \cdot \sum_{i\in[n]} \frac{y_i^2}{x_i} = y^\top \mathbf{X}^{-1} y\,.$$

Consequently we have that $r(x)$ is 1-strongly convex on Δ^n. □

Next, we characterize the minimizers of $r(x)$ plus a linear function over the simplex.

Lemma 18. *For* $e(x) = \sum_{i\in[n]} x_i \cdot \log x_i$, $p \in \mathbb{R}^n$ *we have*

$$z = \text{argmin}_{x\in\Delta_n} p^\top x + r(x)$$

is given by

$$z_i = \frac{\exp(-p_i)}{\sum_{i\in[n]} \exp(-p_i)}\,.$$

Proof. Let $f(x) = p^\top x + r(x)$ we have that $\bigtriangledown f(x)_i = p_i + 1 + \log(x_i)$ where action is coordinate wise. Now we have that for all $x \in \Delta^n$ it is the case that

$$\begin{aligned}\bigtriangledown f(z)^\top (x - z) &= \sum_{i\in[n]} \left[p_i + 1 + \log(z_i)\right]^\top (x - z_i) \\ &= \sum_{i\in[n]} \left[1 - \log\left[\sum_{j\in[n]} \exp(-p_i))\right]\right]^\top (x - z) = 0\end{aligned}$$

Since $z \in \Delta^n$ clearly we have that $\bigtriangledown f(z)^\top (x - z) \geq 0$ for all $x \in \Delta^n$ and thus it is optimal. (Note that this characterization is essentially the way we computed z_i). □

Using this characterization of the minimizer of $r(x)$ plus a linear over the simplex we bound D.

Lemma 19. *We have*

$$\max_{x \in \Delta^n} e(x) - \min_{x \in \Delta^n} e(x) \leq \log n \,.$$

Proof. Note that $x \log x$ is convex on $[0,1]$ with $0 \log 0 = 0$ and $1 \log 1 = 0$ and consequently

$$\max_{x \in \Delta^n} e(x) \leq 0$$

Furthermore, we have that $\arg\min_{x \in \Delta^n} e(x) = \frac{1}{n}\vec{1}$ and this

$$\min_{x \in \Delta^n} e(x) = e\left(\frac{1}{n}\vec{1}\right) = \sum_{i \in [n]} \left(\frac{1}{n}\right) \log\left(\frac{1}{n}\right) = -\log n$$

□

Putting these pieces together gives the following result on solving the learning from experts problem.

Theorem 20. *Suppose in the learning from experts problem we start with $w_1 = 1$ and in each round i let $x_i = w_i / \|w\|_1$, i.e. pick expert i with probability proportional to w_i and let $w_{i+1} = w_i \cdot \exp(-\eta p_i)$ for $\eta = \sqrt{2 \log n / T}$ then* $\text{regret}(T) \leq \sqrt{2T \log n}$.

Proof. We apply FTRL with $r(x) = e(x)$. We have shown that r is 1-strongly convex with respect to $\|\cdot\|_1$ over Δ, that $\|p_t\|_* \leq 1$ for all t and $\max_{x \in \Delta^n} r(x) - \min_{x \in \Delta^n} r(x) \leq \log n$. Consequently, FTRL achieves the desired regret bound as we can pick $G = 1$ and $D = \log n$. To see that the algorithm we have proposed is the same as FTRL note that

$$x_{t+1}(i) = \frac{\exp\left(\sum_{k \in [t]} -\eta \cdot p_t(i)\right)}{\sum_{j \in [n]} \exp\left(\sum_{k \in [t]} -\eta \cdot p_t(j)\right)}$$

and therefore $x_{t+1} = \text{argmin}_{x \in \Delta^n} \Phi_t(x)$ as required. □

Note that it can be shown that this bound is tight in general.

9 Solving Convex Optimization

Here we show how to use FTRL to solve L-Lipschitz continous optimization problems. First we provide the following lemma showing that a function is L-Lipschitz with respect to some norm if and only if its subgradients are bounded with respect to the dual norm.

Lemma 21. *A convex function $f : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz continuous with respect to a norm $\|\cdot\|$, i.e. $|f(x) - f(y)| \leq L \cdot \|x - y\|$ for $x, y \in \mathbb{R}^n$, if and only if for all x and $g \in \partial f(x)$ it is the case that $\|g\|_* \leq L$.*

Proof. First suppose that $\|g\|_* \leq L$ for all $g \in \partial f(x)$. Then for $x, y \in \mathbb{R}^n$ and $g_x \in \partial f(x)$ and $g_y \in \partial f(y)$ we have that

$$f(y) \geq f(x) + g_x^\top (y - x) \geq f(x) - \|g_x\|_* \|y - x\| \geq f(x) - L\|y - x\|$$

and similarly

$$f(x) \geq f(y) + g_y^\top (x - y) \geq f(y) - \|g_y\|_* \|y - x\| \geq f(y) - L\|y - x\|$$

and therefore we have that

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

On the other hand suppose that $\|g\|_* > L$ for some $g \in \partial f(x)$. Then, since for some $z \in \mathbb{R}^n$ with $\|z\| = 1$ we have $g_x^\top z = \|g_x\|_*$ it is the case that

$$f(x+z) \geq f(x) + g_x^\top (x+z-x) = f(x) + \|g_x\|_*$$

Consequently

$$|f(x+z) - f(x)| \geq \|g_x\|_* > L \cdot \|(x+z) - x\| .$$

□

Using this we show provide bounds for unconstrained Lipschitz continuous minimization in $\|\cdot\|_2$. The idea is quite simple, we simply use $\frac{1}{2}\|x_0 - x_*\|_2^2$ as are regularizer.

Lemma 22. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a L-Lipschitz function and S is a closed convex set known explicitly. Further suppose we have a subgradient oracle (and $\partial f(x) \neq \emptyset$ for all $x \in S$) and some point $x_0 \in S$ and $R \geq \|x_0 - x_*\|_2^2$then if x_* is a minimizer of f we can design a method that with k call to the subgradient oracle produces $x_k \in S$ such that*

$$f(x_k) - f_* \leq \sqrt{\frac{2RL^2}{k}}$$

and therefore we can produce an ϵ-suboptimal point with $O(2RL^2/\sqrt{\epsilon})$ queries to the subgradient oracel.

Proof. For all T let $g_T \in \partial f(x_T)$ and let consider the algorithm

$$x_{T+1} = \text{argmin}_{x \in S} \eta \sum_{t \in [T]} g_t^\top x + \frac{1}{2}\|x - x_0\|_2^2$$

we know that $\|g_T\|_2 \leq L$ for all T and thus for $r(x) = \frac{1}{2}\|x - x_0\|_2^2$ since r is 1 strongly convex we have

$$\sum_{t \in [T]} g_t^\top (x_t - x_*) \leq \frac{\eta}{2} \cdot T \cdot L^2 + \frac{1}{\eta} \cdot \left[r(x_*) - \min_{x \in S} r(x) \right]$$

However, since clearly $\min_{x \in S} r(x) = x_0$ we have that

$$\frac{1}{T} \sum_{t \in [T]} g_t^\top (x_t - x_*) \leq \frac{\eta}{2} \cdot L^2 + \frac{R}{\eta T}$$

Since $f(x_*) \geq f(x_t) + g_t^\top (x_* - x_t)$ we have that for $\eta = \sqrt{2R/(L^2 T)}$

$$f\left(\frac{1}{T} \sum_{t \in [T\}} x_t \right) - f_* \leq \frac{1}{T} \sum_{t \in [T]} g_t^\top (x_t - x_*) \leq \sqrt{\frac{2RL^2}{T}} .$$

□

What exactly is this algorithm in the unconstrained case? Note that $\vec{0} = \bigtriangledown \Phi_t(x_{t+1}) = \eta \sum_{t \in [T]} g_t + x_{t+1} - x_0$ and thus $x_{t+1} = x_0 - \eta \sum_{t \in [T]} g_t$ and thus this algorithm is $x_{t+1} = x_t - \eta g_t$, i.e. it is essentially gradient descent, aka *subgradient descent*. In this case we have shown that the average of the iterates converges for this algorithm for decaying choice of η.

10 Mirror Descent

Here we consider another algorithm for solving the online linear optimization. This algorithm is known as *mirror descent* and the analysis will give us another interesting perspective on the problem.

Now rather than using our strongly convex function directly in a single minimization problem, we will use it through Bregman divergences defined as follows.

Definition 23 (Bregman Divergence)**.** For differentiable strongly convex function $r : \mathbb{R}^n \to \mathbb{R}$ the Bregman Divergence, $D_r(x||y)$ for $x, y \in \mathbb{R}^n$ is given by $D_r(x||y) \stackrel{\text{def}}{=} r(x) - \left[r(y) + \bigtriangledown r(y)^\top (x - y)\right]$. Furthermore, for closed, convex, non-empty $S \subseteq \mathbb{R}^n$ we let $\pi_S^r(y) \stackrel{\text{def}}{=} \text{argmin}_{x \in S} D(x||y)$.

The Bregman divergence is a way of turning an arbitrary differentiable strongly convex function into a distance-like function (though triangle inequality need not hold of it). Indeed, so long as r is μ-strongly convex with respect to $\|\cdot\|$for $\mu > 0$ we can show that $D(x||y) \geq \frac{\mu}{2}\|x - y\|^2$ and thus $D(x||y) = 0$ if and only if $x = y$. (See homework.) Furthermore, this implies that it is reasonable to think of $\pi_S^r(y)$ as the projection of y onto S.

The other notion we need is that of a *mirror map*.

Definition 24 (Mirror Map)**.** We call differentiable strongly convex $r : \mathbb{R}^n \to \mathbb{R}$ a *mirror map if for all* $x \in \mathbb{R}^n$ it is the case that $x = \bigtriangledown r(y)$ for some y in the domain of r.

The idea of a mirror map is that it is a way to translate from the dual space, i.e. the space of gradients or subgradients, back to the domain. Note that we often measure differences between points in one norm but norms of gradients in the dual norm. While in finite dimensions all norms approximate each other up to multiplicative factors and therefore such a primal space can always be related to the dual space, this need not happen in infinite dimension. For example, if we have an infinite dimensional vector $g \in \ell_\infty$ and $x \in \ell_1$ the operation $y = x + \eta g$ doesn't necessarily even make sense. A mirror map provides a way to avoid such issues by providing the mapping from dual space back to primal.

Note that clearly $r(x) = \frac{1}{2}\|x - x_0\|_2^2$ is a mirror map as $\bigtriangledown r(x) = x - x_0$. Furthermore, clearly $e(x) = x \log x$ for $x \in \mathbb{R}^n_{\geq 0}$ is a mirror map as $\bigtriangledown e(x) = \vec{1} + \log(x)$ and $\bigtriangledown e(x) = y$ if and only if $x = \exp(y - 1)$ which we can always ensure.

These notions give another interesting perspective on FTRL provided that r is a mirror map. Note that the step

$$x_{t+1} = \text{argmin}_{x \in S} \eta \sum_{k \in [t]} p_k^\top x + r(x)$$

is trivially the same as

$$x_{t+1} = \text{argmin}_{x \in S} D_r(x||y_t)$$

provided where t is chosen such that

$$\bigtriangledown r(y_t) = -\eta \sum_{k \in [t]} p_k \,.$$

Consequently, we see that FTRL is equivalent to the following. Start with y_0 such that $\bigtriangledown r(y_0) = 0$ and then for all t

- Let $x_{t+1} = \pi_S^r(y_t) = \text{argmin}_{x \in S} D_r(x||y_t)$
- Let y_{t+1} be such that $\bigtriangledown r(y_{t+1}) = \bigtriangledown r(y_t) - \eta p_t$

In other words, this algorithm is essentially performing gradient descent in the dual (on $\bigtriangledown r(y_t)$) and then projecting back to x_t. This is known as *lazy mirror descent* or *dual averaging*, in the context of convex

minimization. Here, the updates are occurring primarily in the dual, $\bigtriangledown r$, space and we simply do projection to get points in S.

Another natural modification of this algorithm would be to do exactly the same procedure, but project every iteration and therefore better maintain points in S. Here we could start with some $y_0 = x_0 \in S$ and for all t repeat the following

- Let $x_{t+1} = \text{argmin}_{x \in S} D(x||y_t)$
- Let y_{t+1} be such that $\bigtriangledown r(y_{t+1}) = \bigtriangledown r(x_{t+1}) - \eta p_{t+1}$.

The algorithm is known as *mirror descent*. Here we perform the gradient descent update on the mapping of x_{t+1} back into the dual space. Note that in many cases, *mirror descent* and *lazy mirror descent,* can be shown to be equivalent. However, the analysis typically revolves around different potential functions. The analysis of mirror descent as the virtue that we can primarily measure progress by distances in the primal, S, space.

We analyze the performance of mirror descent in the remainder of this section.

Lemma 25. *In each iteration of mirror descent when r is a μ-strongly convex mirror map with respect to $\|\cdot\|$ and S is a closed set we have that for all $z \in \mathbb{R}^n$*

$$\begin{aligned} D_r(z||y_t) - D_r(z||x_t) &\leq -\frac{\mu}{2}\|y_t - x_t\|^2 - \eta p_t^\top (y_t - x_t + x_t - z) \\ &\leq \frac{\eta^2}{2\mu}\|p_t\|_*^2 - \eta \cdot p_t^\top (x_t - z) \end{aligned}$$

Proof. Direct calculation reveals that for all $t \geq 0$

$$\begin{aligned} D(z||y_t) - D(z||x_t) &= \left[r(z) - r(y_t) - \bigtriangledown r(y_t)^\top (z - y_t)\right] - \left[r(z) - r(x_t) - \bigtriangledown r(x_t)^\top (z - x_t)\right] \\ &= r(x_t) - r(y_t) - \bigtriangledown r(x_t)^\top (x_t - y_t) - \eta p_t^\top (y_t - z) \end{aligned}$$

Since $r(y_t) \geq r(x_t) + \bigtriangledown r(x_t)^\top (y_t - x_t) + \frac{\mu}{2}\|y_{t+1} - x_t\|^2$ by strong convexity this implies

$$\begin{aligned} D(z||y_t) - D(u||x_t) &\leq -\frac{\mu}{2}\|y_t - x_t\|^2 - \eta p_t^\top (y_t - x_t + x_t - z) \\ &\leq \frac{\eta^2}{2\mu}\|p_t\|_*^2 - \eta \cdot p_t^\top (x_t - z). \end{aligned}$$

□

This lemma shows that we can bound the changed in Bregman divergence between any vector and y_t and x_t. However, we would really like to relate this to changes only in x_t. For this we can show a type of generalized Pythagorean theorem that holds for Bregman divergences.

Lemma 26. *If r is strongly convex and S is a closed convex set then for $y \in S$ and $z \in \mathbb{R}^n$*

$$D_r(x||y) + D_r(y||z) \leq D_r(x||z) \text{ for all } x \in S$$

if and only if $y = \pi_S^r(z)$.

Proof. Homework. □

We call this a generalized Pythagorean theorem as if $r = \frac{1}{2}\|x\|_2^2$ then the inequality in this lemma is$\|x - y\|_2^2 + \|y - z\|_2^2 \leq \|x - z\|_2^2$.

Using this we see that in mirror descent that for all $z \in S$ clearly $D(z||x_{t+1}) \leq D(z||x_t)$.

Theorem 27 (Mirror Descent)**.** *Mirror Descent for μ-strongly convex r with respect to $\|\cdot\|$ where we let $y_0 = x_0 \in S$ ensures that for all $z \in S$*

$$\sum_{t\in[T]} p_t^\top (x_t - z) \leq \sum_{t\in[T]} \frac{\eta}{2\mu}\|p_t\|_*^2 + \frac{1}{\eta} D_r(z||x_0)\,.$$

Proof. By the previous lemmas we have that for all $t \in [T]$

$$D_r(z||x_{t+1}) - D(z||x_t) \leq \frac{\eta^2}{2\mu}\|p_t\|_*^2 - \eta \cdot p_t^\top (x_t - z)\,.$$

Summing over $t \in [T]$ and re-arranging terms yields that

$$\sum_{t\in[T]} p_t^\top (x_t - z) \leq \frac{\eta}{2\mu}\|p_t\|_*^2 + \frac{1}{\eta}\left[D_r(z||x_1) - D_r(z||x_{t+1})\right]$$

However, $D_r(z|||x_{t+1}) \geq 0$ and $x_1 = x_0$ yielding the claim □

Thus we see that we can also use this theorem to get a $\sqrt{T}$ regret growth rate with essentially the same bounds as those we got for FTRL.

11 Mirror Descent for Smooth Functions

Here we consider the performance of mirror descent on smooth functions. Suppose we ran the same algorithm as before for L-smooth $f : \mathbb{R}^n \to \mathbb{R}$ and let $p_t = \bigtriangledown f(x_t)$. Then, letting $z = x_*$, the guarantees of mirror descent yield that

$$\sum_{t\in[T]} [f(x_t) - f(x_*)] \leq \sum_{t\in[T]} p_t^\top (x_t - x_*) \leq \sum_{t\in[T]} \frac{\eta}{2\mu}\|p_t\|_*^2 + \frac{1}{\eta}\cdot D_r(x_*||x_0)$$

However, we also have that $\|p_t\|_*^2 = \|\bigtriangledown f(x_t)\|_2^2 \leq 2L \cdot [f(x_t) - f(x_*)]$ and thus

$$\sum_{t\in[T]} \left(1 - \frac{\eta L}{\mu}\right)[f(x_t) - f(x_*)] \leq \frac{1}{\eta} D_r(x_*||x_0)$$

Consequently picking $\eta = \frac{\mu}{2L}$ we have

$$f\left(\frac{1}{T}\sum_{t\in[T]} x_t\right) - f_* \leq \frac{1}{T}\sum_{t\in[T]} [f(x_t) - f_*] \leq \frac{2L}{\mu T}\cdot D_r(x_*||x_0)\,.$$

Thus we see that we now get a $1/T$ rate for smooth minimization. Note that if instead of running mirror descent we ran FTRL, then we could obtain exactly the same guarantee. The $1/T$ rate stems not from the analysis of mirror descent, but rather just an interpretation of its guarantees and picking η appropriately.

12 Stochastic Gradient Descent (SGD) and Finite Sums

Here we consider using the previous algorithms in the case where we pick p_t to be a random variable so that $\mathbb{E}p_t \in \partial f(x_t)$. Such algorithms are known as stochastic gradient descent (SGD). Note that if in each iteration we have that $\mathbb{E}\|p_t\|_*^2 \leq G^2$ then we get exactly the same bounds as before with the only difference

that our regret bounds become expected regret bounds and the function errors we obtain become expected function errors.

To demonstrate the power of this technique, here we show how to use SGD through a technique known as variance reduction to get faster algorithms for minimizing finite sums and more broadly, regression.

12.1 The Problem

The main problem we consider in this section is what is known as the *finite sum problem*.

Definition 28 (Finite Sum Problem)**.** We have a function $f : \mathbb{R}^d \to \mathbb{R}$ that is μ-strongly convex such that $f(x) = \sum_{i\in[n]} f_i(x)$ where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is L-smooth and convex. We wish to minimize f given some initial point $x_0 \in \mathbb{R}$ and an oracle that when queried at $x \in \mathbb{R}^d$ and $i \in [n]$ outputs $\bigtriangledown f_i(x)$.

This problem encompasses a common setting in machine learning, where we wish to opitmize over some set of variables x and have n pieces of data each of which should be explained by x. The terms $f_i(x)$ are then used to measure how well data point x_i is explained by f. Due to the popularity of this problem there are many variants considered, however we will restrict our analysis to this relatively simple form.

A particularly popular (and perhaps simplest) instance of the finite sum problem is the *regression problem*.

Definition 29 (Regression Problem)**.** Given $\mathbf{A} \in \mathbb{R}^{n\times d}$ with rows $a_1, ..., a_n \in \mathbb{R}^d$ we wish to solve $\min_{x\in\mathbb{R}^d} f(x)$ where

$$f(x) = \frac{1}{2}\|\mathbf{A}x - b\|_2^2 = \sum_{i\in[n]} \frac{1}{2}(a_i^\top x - b_i)^2 = \frac{1}{2}x^\top \mathbf{A}^\top \mathbf{A}x - b^\top \mathbf{A}x + \frac{1}{2}b^\top b\,.$$

such that $\lambda_{\min}(\mathbf{A}^\top\mathbf{A}) \geq \mu$ and $\|a_i\|_2^2 \leq L$ for all $i \in [n]$.

Note that $\bigtriangledown^2 f(x) = \mathbf{A}^\top\mathbf{A}$ and thus $\lambda_{\min}(\mathbf{A}^\top\mathbf{A}) \geq \mu$ if and only if f is μ-strongly convex. Also note that if we let $f_i(x) = \frac{1}{2}(a_i^\top x - b_i)^2$ then $\bigtriangledown^2 f_i(x) = a_i a_i^\top$ which has eigenvector a_i of eigenvalue $\|a_i\|_2^2$ and all other eigenvectors 0. Consequently f_i is convex and L-smooth if and only if $\|a_i\|_2^2 \leq L$ and thus this is an instance of the finite sum problem where the oracle can be implemented in $O(d)$ time.

Now note that in the finite sum problem clearly f is nL smooth and since a gradient can be computed with n oracle evaluations, an ϵ-suboptimal point can be computed by gradient descent with $O(n \cdot \frac{nL}{\mu} \cdot \log((f(x_0) - f_*)/\epsilon))$ calls to the oracle. Moreover, this can be improved by AGD to $O(n \cdot \sqrt{\frac{nL}{\mu}} \cdot \log((f(x_0) - f_*)/\epsilon))$ calls to the oracle. Note that in the case of regression these correspond to running times of $O(nd \cdot \frac{nL}{\mu} \cdot \log((f(x_0) - f_*)/\epsilon))$ and $O(nd \cdot \sqrt{\frac{nL}{\mu}} \cdot \log((f(x_0) - f_*)/\epsilon))$.

The main question of this section is can these be improved. We will show that the factor of n can be improved using SGD and sampling techniques at least over regression. Accelerated gains can be made as well, however we will simply provide references for those improvements. Again, this is an active area of research and there are more results then we can present here, the goal in this section is simply to show how sampling can provide general improvements over gradient descent.

12.2 Warm-up: Direct SGD

So how should we try to improve over gradient descent for the finite sum problem? Perhaps the most natural idea would simply be to perform SGD, i.e in iteration k pick $i_k \in [n]$ at random and let $p_k = n \cdot \bigtriangledown f_i(x_k)$ and use this for mirror descent. Note that

$$\mathbb{E}p_k = \frac{1}{n}\sum_{i\in[n]} n \cdot \bigtriangledown f_i(x_k) = \sum_{i\in[n]} \bigtriangledown f_i(x_k) = \bigtriangledown f(x_k)$$

While we could do this in the full generality of the situation we do mirror descent or FTRL, here for completeness we analyze this directly in the simple case where our regularizer is $r(x) = \frac{1}{2}\|x\|_2^2$ and we work with $\|\cdot\|_2^2$, i.e. our algorithm is $x_{k+1} = x_k - \eta \cdot p_k$. We provide this lemma with these settings but a slightly more general formulation for how p_k is chosen.

Lemma 30 (Specialized SGD)**.** *Let $\mathcal{D}$ be a distribution over convex L-smooth functions $g : \mathbb{R}^d \to \mathbb{R}$ and let $f(x) \stackrel{\text{def}}{=} \mathbb{E}_{g\sim\mathcal{D}} g(x)$. Suppose for some $x_0 \in \mathbb{R}^d$ and all $k \geq 0$ we let $x_{k+1} = x_k - \eta \bigtriangledown g_k(x_k)$ where $g_k \sim \mathcal{D}$ independently at random. Then, for all $T \geq 0$ we have*

$$2\eta \cdot [\mathbb{E}_{g_k\sim\mathcal{D}} f(x_k) - f_*] \leq \|x_k - x_*\|_2^2 - \mathbb{E}_{g_k\sim\mathcal{D}}\|x_{k+1} - x_*\|_2^2 + \eta^2 \cdot \mathbb{E}_{g_k\sim\mathcal{D}}\| \bigtriangledown g_k(x_k)\|_2^2$$

and therefore

$$\sum_{k\in[T]} \frac{1}{T}\mathbb{E} f(x_k) - f_* \leq \frac{\eta}{2T}\sum_{k\in[T]} \mathbb{E}\| \bigtriangledown g_k(x_k)\|_2^2 + \frac{1}{2\eta T}\|x_0 - x_*\|_2^2 .$$

If we further assume that f is μ-strongly convex this simplifies to

$$\sum_{k\in[T]} \frac{1}{T}\mathbb{E} f(x_k) - f_* \leq \frac{\eta}{2T}\sum_{k\in[T]} \mathbb{E}\| \bigtriangledown g_k(x_k)\|_2^2 + \frac{1}{\eta\mu T}\left[f(x_0) - f_*\right] .$$

Proof. Direct algebraic manipulation yields that

$$\begin{aligned}\mathbb{E}\|x_{k+1} - x_*\|_2^2 &= \mathbb{E}\|x_k - x_* - \eta \bigtriangledown g_k(x_k\|_2^2 = \mathbb{E}\left[\|x_k - x_*\|_2^2 - 2\eta \bigtriangledown g_k(x_k)^\top (x_k - x_*) + \eta^2\|g_k\|_2^2\right] \\ &= \|x_k - x_*\|_2^2 - 2\eta \bigtriangledown f(x_k)^\top (x_k - x_*) + \eta^2\|g_k\|_2^2\end{aligned}$$

Since $f(x_*) \geq f(x_k) + \bigtriangledown f(x_k)^\top (x_* - x_k)$ by convexity we have that

$$2\eta \cdot [f(x_k) - f_*] \leq \|x_k - x_*\|_2^2 - \mathbb{E}_{g_k\sim\mathcal{D}}\|x_{k+1} - x_*\|_2^2 + \eta^2 \cdot \mathbb{E}_{g_k\sim\mathcal{D}}\|g_k\|_2^2$$

yielding the first inequality. Summing, taking expectation over all the g_k, and noting that the $\mathbb{E}\|x_k - x_*\|_2^2$ cancel yields

$$2\eta \sum_{k\in[T]} [\mathbb{E} f(x_k) - f_*] \leq \sum_{k\in[T]} \eta^2\mathbb{E}\|g_k\|_2^2 + \mathbb{E}\|x_0 - x_*\|_2^2 - \mathbb{E}\|x_{k+1} - x_*\|_2^2 .$$

Minor algebraic manipulations and the fact that $f(x_0) \geq f(x_*) + \frac{\mu}{2}\|x - x_*\|_2^2$ when f is strongly convex yields the result. □

This lemma reduces much of the analysis of SGD to bounding $\mathbb{E}\| \bigtriangledown g_k(x_k)\|_2^2$. As we have seen throughout our analysis of mirror descent the smaller we can show this quantity is, the faster we can converge. Moreover, as we have seen in the analysis of mirror descent for smooth funcitons, if we can somehow relate this quantity to $f(x_k) - f_*$ then we can simply have our error decay as the difference from x_0 to x_* over a linear term. Since, when our function is strongly convex, we can in turn relate this to our initial function error and hope to halve the error after some fixed number of iterations and achieve an algorithm a $O(\log \epsilon^{-1})$ dependence on error.

In short, to have any hope of competing with GD we need to ensure $\mathbb{E}\| \bigtriangledown g_k(x_k)\|_2^2$ decays as $f(x_k) - f_*$ goes to 0. However, note that it might be the case that $\mathbb{E}\| \bigtriangledown g_k(x_*)\|_2^2 \neq 0$ and consequently as we get closer to the minimizer we still might have a fixed lower bound on how big $\mathbb{E}\| \bigtriangledown g_k(x_k)\|_2^2$ thereby impinging on our ability to get a $O(\log \epsilon^{-1})$ rate. In the next section we show how to deal with this issue. For now we explore what rates we can get depending on what $\sigma^2 \stackrel{\text{def}}{=} \mathbb{E}\| \bigtriangledown g_k(x_*)\|_2^2$.

The first step in this analysis is to bound $\mathbb{E}\| \bigtriangledown g_k(x_k) - \bigtriangledown g_k(x_*)\|_2^2$. This will allow us to break the contribution of $\mathbb{E}\| \bigtriangledown g_k(x_k)\|_2^2$ into the contribution to this term and σ^2. Note that if there was no distribution, we would have that this is at most $2L \cdot [f(x_k) - f_*]$ by smoothness. Below we show the same holds.

Lemma 31 (Variance Bound)**.** *Let $\mathcal{D}$ be a distribution over convex L-smooth functions $g : \mathbb{R}^d \to \mathbb{R}$ and let $f(x) \stackrel{\text{def}}{=} \mathbb{E}_{g\sim\mathcal{D}} g(x)$ and x_* be a minimizer of f. Then for all $x \in \mathbb{R}^n$ we have $\mathbb{E}_{g\sim\mathcal{D}} \| \bigtriangledown g(x) - \bigtriangledown g(x_*)\|_2^2 \leq 2L \cdot [f(x) - f_*]$.*

Proof. For all $g \in \text{supp}(\mathcal{D})$ since g is convex and L-smooth then by Lemma 7 in Chapter 5 we have

$$\mathbb{E}_{g\sim\mathcal{D}} \| \bigtriangledown g(x) - \bigtriangledown g(x_*)\|_2^2 \leq \mathbb{E}_{g\sim\mathcal{D}} 2L \cdot \left[g(x) - \left[g(x_*) - \bigtriangledown g(x_*)^\top (x - x_*)\right]\right]$$

and the result follows from the definition of f and the fact that $\mathbb{E}_{g\sim\mathcal{D}} \bigtriangledown g(x_*) = \bigtriangledown f(x_*) = 0$ by the optimality of x_*.

Using this, we show that if $\sigma^2 = 0$ then we can achieve our desired rate. Note that $\sigma^2 = 0$ implies that for all $g \in \text{supp}(\mathcal{D})$ it isi the case that $\bigtriangledown g(x_*) = 0$. In the case of regression, where $g_k(x_k) = \frac{n}{2} \cdot (a_{i_k}^\top x_k - b_{i_k})^2$ for some $i_k \in [n]$ we have that $\bigtriangledown g_k(x_k) = n \cdot a_{i_k}(a_{i_k}^\top x_k - b_{i_k})$ and thus assuming that the $a_i \neq 0$ this implies that $a_i^\top x_* = b_i$ for all $i \in [n]$ or in other words, $\mathbf{A}x_* = b$. In other words, the following gives us bounds for solving linear systems $\mathbf{A}x = b$ when there is a x_* such that $\mathbf{A}x_* = b$. □

Lemma 32. *In the setup of Lemma 30 if $\sigma^2 = \mathbb{E}_{g\sim\mathcal{D}} \| \bigtriangledown g(x_*)\|_2^2 = 0$ then for all $k \geq 1$ if $\eta = \frac{1}{2L}$*

$$\mathbb{E}\|x_k - x_*\|_2^2 \leq \left(1 - \frac{\mu}{4L}\right) \mathbb{E}\|x_{k-1} - x_*\|_2^2 \leq \left(1 - \frac{\mu}{4L}\right)^k \|x_0 - x_*\|_2^2$$

and consequently we can compute an expected ϵ-suboptimal point with $O(\frac{L}{\mu} \cdot \log\left(\frac{L}{\mu} \cdot \frac{f(x_0) - f_*}{\epsilon}\right)$ oracle evaluations.

Proof. By Lemma 30 we have

$$\mathbb{E}_{g_k\sim\mathcal{D}} \|x_{k+1} - x_*\|_2^2 \leq \|x_k - x_*\|_2^2 - 2\eta \cdot [\mathbb{E}_{g_k\sim\mathcal{D}} f(x_k) - f_*] + \eta^2 \cdot \mathbb{E}_{g_k\sim\mathcal{D}} \| \bigtriangledown g_k(x_k)\|_2^2$$

However, since $\sigma^2 = 0$ by Lemma 31 we have

$$\mathbb{E}_{g_k\sim\mathcal{D}} \| \bigtriangledown g_k(x_k)\|_2^2 = \mathbb{E}_{g_k\sim\mathcal{D}} \| \bigtriangledown g_k(x_k) - \bigtriangledown g_k(x_*)\|_2^2 \leq 2 \cdot L \cdot [f(x_k) - f_*] \ .$$

Consequently we have

$$\mathbb{E}_{g_k\sim\mathcal{D}} \|x_{k+1} - x_*\|_2^2 \leq \|x_k - x_*\|_2^2 - 2\eta(1 - L\eta) \cdot \mathbb{E}_{g_k\sim\mathcal{D}} [f(x_k) - f_*]$$

Since for $\eta = 1/(2L)$ we have $2\eta(1 - L\eta) = 1/(2L)$ and since $f(x_k) - f_* \geq \frac{\mu}{2}\|x_k - x_*\|_2^2$ by strong convexity the first result holds. Since f is L-smooth we have that $f(x_k) - f_* \leq \frac{L}{2}\|x_k - x_*\|_2^2$ and therefore $\mathbb{E} f(x_k) - f_* \leq \frac{L}{\mu}\left(1 - \frac{\mu}{4L}\right)^k [f(x_0) - f_*]$ yielding the second claim. □

Note that this says we can achieve our desired running time for regression and minimizing finite sums when $\sigma^2 = 0$. What about when $\sigma^2 \neq 0$? We analyze this case in the remainder of this section.

In this case we have the following.

Lemma 33. *In the setup of Lemma 30 if $\sigma^2 = \mathbb{E}_{g\sim\mathcal{D}} \frac{1}{2} \| \bigtriangledown g(x_*)\|_2^2 > 0$ and f is μ-strongly convex then for all $k \geq 1$ we have*

$$(1 - 2\eta \cdot L)\left[\frac{1}{T}\sum_{k\in[T]} \mathbb{E} f(x_k) - f_*\right] \leq \eta \cdot \sigma^2 + \frac{1}{\eta\mu T}[f(x_0) - f_*] \ .$$

and consequently we can compute an expected ϵ-suboptimal point with $O(\frac{L}{\mu} \cdot \log\left(\frac{L}{\mu} \cdot \frac{f(x_0) - f_*}{\epsilon}\right)$ oracle evaluations.

Proof. By Lemma 30 we have

$$\sum_{k\in[T]} \frac{1}{T}\mathbb{E}f(x_k) - f_* \leq \frac{\eta}{2T}\sum_{k\in[T]} \mathbb{E}\| \bigtriangledown g_k(x_k)\|_2^2 + \frac{1}{\eta\mu T}\left[f(x_0) - f_*\right].$$

However, since $\|a+b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2$ (as $\|a+b\|_2^2 + \|a-b\|_2^2 = 2\cdot\|a\|_2^2 + 2\|b\|_2^2$) we have that

$$\sum_{k\in[T]} \mathbb{E}\| \bigtriangledown g_k(x_k)\|_2^2 = \sum_{k\in[T]} \mathbb{E}\left[2\| \bigtriangledown g_k(x_k) - \bigtriangledown g_k(x_*)\|_2^2 + 2\| \bigtriangledown g_k(x_*)\|_2^2\right] \leq 4L\cdot\sum_{k\in[T]}\left[\mathbb{E}f(x_k) - f_*\right] + 2T\cdot\sigma^2$$

yielding the desired result. □

Thus we see that we can get fast rates provied we can related σ^2 to $f(x_0) - f_*$. This is precisely what we show how to do in the next section.

12.3 Variance Reduction

From the analysis in the previous section it seems that if we could find a way to relate σ^2 to $f(x_0) - f_*$ in the finite sum problem the we can achieve fast rates. Another way to say this, is that the problem with making SGD faster than gradient descent for regression is finding away to ensure that $\mathbb{E}_{g\sim\mathcal{D}}\| \bigtriangledown g(x_*)\|_2^2$ decays over time.

To do this we simply want to write $f(x) = \mathbb{E}_{g\sim\mathcal{D}}g(x)$ for better g. We have already seen the effects of picking $g = n\cdot f_i$ with probability $\frac{1}{n}$. The question is how to do better? What natural idea would be to pick $g(x) = n\cdot\left[f_i(x) - \bigtriangledown f_i(x_*)^\top x\right]$ with probability $\frac{1}{n}$. Note that $\mathbb{E}_{g\sim\mathcal{D}}g(x) = f(x) - \bigtriangledown f(x_*)^\top x = f(x)$ and thus this is a valid transformation. Furthermore, we have $\bigtriangledown g(x_*) = 0$ for all g in the distribution. This works quite well, but unfortunately has the problem of requiring x_* to be computed. Another, idea would be to perform the same transformation, but using x_0 as a proxy for x_*, this would yield $g(x) = n\cdot\left[f_i(x) - \bigtriangledown f_i(x_0)^\top x\right]$ with probability $\frac{1}{n}$. However, unfortunately, $\mathbb{E}g(x) = f(x) - \bigtriangledown f(x_0)^\top x$ and therefore is biased.

To fix all these issues what we actually do is let $g(x) = n\cdot\left[f_i(x) - \bigtriangledown f_i(x_0)^\top x\right] + \bigtriangledown f(x_0)^\top x$ with prbability $\frac{1}{n}$. This has that $\mathbb{E}g(x) = f(x)$ as desired. The corresponding algorithm is known as stochastic variance reduced gradient. To analyze it, all we have to do is bound σ^2 for this particular g, i.e. $\mathbb{E}_{g\sim\mathcal{D}}\frac{1}{2}\| \bigtriangledown g(x_*)\|_2^2 = \frac{1}{2}\sum_{i\in[n]}\frac{1}{n}\|n\left[\bigtriangledown f_i(x_*) - \bigtriangledown f_i(x_0)\right] + \bigtriangledown f(x_0)\|_2^2$.

Lemma 34. *Let $\mathcal{D}$ be a distribution over convex L-smooth functions $g:\mathbb{R}^d\to\mathbb{R}$ and let $f(x) \stackrel{\text{def}}{=} \mathbb{E}_{g\sim\mathcal{D}}g(x)$, $x_0\in\mathbb{R}^d$, and x_* be a minimizer of f then*

$$\mathbb{E}_{g\sim\mathcal{D}}\| \bigtriangledown g(x_*) - \bigtriangledown g(x_0) + \bigtriangledown f(x_0)\|_2^2 \leq 2L\cdot\left[f(x) - f_*\right].$$

Proof. Note that for any random $X\in\mathbb{R}^d$ we have

$$\mathbb{E}\|X - \mathbb{E}X\|_2^2 = \mathbb{E}\|X\|_2^2 - 2\mathbb{E}X^\top\mathbb{E}X + \|\mathbb{E}X\|_2^2 = \mathbb{E}\|X\|_2^2 - \|\mathbb{E}X\|_2^2 \leq \mathbb{E}\|X\|_2^2.$$

and consequently

$$\begin{aligned}\mathbb{E}_{g\sim\mathcal{D}}\| \bigtriangledown g(x_*) - \bigtriangledown g(x_0) + \bigtriangledown f(x_0)\|_2^2 &= \mathbb{E}_{g\sim\mathcal{D}}\| \bigtriangledown g(x_*) - \bigtriangledown g(x_0) - \left[\bigtriangledown f(x_*) - \bigtriangledown f(x_0)\right]\|_2^2 \\ &\leq \mathbb{E}_{g\sim\mathcal{D}}\| \bigtriangledown g(x_*) - \bigtriangledown g(x_0)\|_2^2.\end{aligned}$$

The result then follows from Lemma 31. □

This lemma then says we can bound σ^2 by $2L\cdot[f(x_0) - f_*]$.

Theorem 35 (SVRG). *Suppose in the finite sum problem for some $x_0 \in \mathbb{R}^d$ and all $k \geq 0$ we let*

$$x_{k+1} = x_k - \eta \left[n \cdot [\bigtriangledown f_i(x_i) - \bigtriangledown f_i(x_0)] + \bigtriangledown f(x_0) \right]$$

then for $\eta \in (0, \frac{1}{2L})$ we have

$$\frac{1}{T} \sum_{k \in [T]} \mathbb{E} f(x_k) - f_* \leq \frac{1}{1 - 2\eta \cdot L} \left[2L\eta + \frac{1}{\eta \mu T} \right] \cdot [f(x_0) - f_*] \, .$$

and consequently we can compute an expected ϵ-suboptimal point with $O((n + \frac{nL}{\mu}) \log(\frac{f(x_0) - f_*}{\epsilon}))$ queries to the oracle

Proof. Note that this is the same as running SGD where we draw $g_k \sim \mathcal{D}$ such that $g_k = n \cdot [f_{i_k}(x) - f_{i_k}(x_0)] + \bigtriangledown f(x_0)$ where $i_k \in [n]$ is uniformly random. Since each g_k is convex and nL smooth and since $\sigma^2 \leq 2L \cdot [f(x_0) - f_*]$ as we have argued we have

$$(1 - 2\eta \cdot L) \sum_{k \in [T]} \left[\frac{1}{T} \mathbb{E} f(x_k) - f_* \right] \leq 2L\eta \cdot [f(x_0) - f_*] + \frac{1}{\eta \mu T} [f(x_0) - f_*]$$

yielding the first claim. Now, if we pick $\eta = \frac{1}{8L}$ and $T = 64 \frac{L}{\mu}$ we have

$$\frac{1}{T} \sum_{k \in [T]} \mathbb{E} f(x_k) - f_* \leq 2 \cdot \left[\frac{1}{4} + \frac{1}{8} \right] \cdot [f(x_0) - f_*] \leq \frac{3}{4} \cdot [f(x_0) - f_*]$$

Consequently we can decrease the function error by a constant factor with $O(n \frac{L}{\mu})$ iterations. Repeating this (i.e. taking either the average point or a random point as the next x_0) and repeating then yields the claim. Each time it is repeated we need n oracle calls to get $\bigtriangledown f(x_0)$ and an additional $\frac{nL}{\mu}$, 1 for each of the iterations. □

Consequently, we can generically improve upon gradient descent. Furthermore, it can be shown that there are cases where $L/\mu < 1$ and thus the number of oracle calls needed scales just linearly with n in these cases.

12.4 Extensions

This is an incredibly well studied problem and a very active area of research. There are numerous extensions that can and are considered. Here, we briefly state a few. First, if f_i is L_i smooth (rather than L smooth) non-uniform smapling can be applied to achieve an $\tilde{O}(\sum_{i \in [n]} \frac{L_i}{\mu})$ query algortihm. Furthermore, this can be accelerated to yield a $\tilde{O}\left(\sum_{i \in [n]} \sqrt{\frac{L_i}{\mu}} \right)$ query algorithm. There are also composite analogs and it can be shown that there are interesting connections between these algorithms and coordinate descent (in some sense SVRG is an analog of running coordinate descent in the dual of the problem). If you would like further references to these, please post on Piazza.