17
The Gradient Descent Framework

Consider the problem of finding the minimum-energy s-t electrical unit flow: we wanted to minimize the total energy burn

$$\mathcal{E}(f) = \sum_e f_e^2 r_e$$

for flow values f that represent a unit flow from s to t (these form a polytope). We alluded to algorithms that solve this problem, but one can also observe that $\mathcal{E}(f)$ is a convex function, and we want to find a minimizer within some polytope K. Equivalently, we wanted to solve the linear system

$$L\phi = (e_s - e_t),$$

which can be cast as finding a minimizer of the convex function

$$\|L\phi - (e_s - e_t)\|^2.$$

How can we minimize these functions efficiently? In this lecture, we will study the gradient descent framework for the general problem of minimizing functions, and give concrete performance guarantees for the case of convex optimization.

17.1 Convex Sets and Functions

First, recall the following definitions:

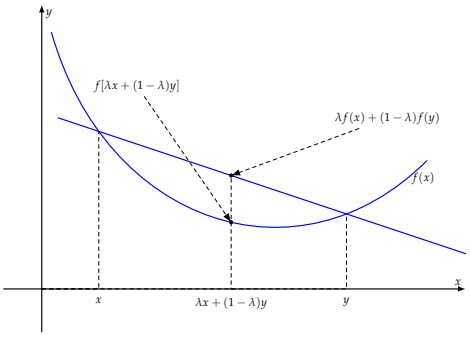
Definition 17.1 (Convex Set)**.** A set $K \subseteq \mathbb{R}^n$ is called *convex* if for all $x, y \in K$,

$$\lambda x + (1 - \lambda)y \in K, \tag{17.1}$$

for all values of $\lambda \in [0, 1]$. Geometrically, this means that for any two points in K, the line connecting them is contained in K.

Definition 17.2 (Convex Function)**.** A function $f : K \to \mathbb{R}$ defined on a convex set K is called *convex* if for all $x, y \in K$,

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y), \tag{17.2}$$



for all values of $\lambda \in [0, 1]$.

There are two kinds of problems that we will study. The most basic question is that of *unconstrained convex minimization* (UCM): given a convex function f, we want to find

$$\min_{x \in \mathbb{R}^n} f(x).$$

In some cases we will be concerned with the constrained convex minimization (CCM) problem: given a convex function f and a convex set K, we want to find

$$\min_{x \in K} f(x).$$

Note that setting $K = \mathbb{R}^n$ gives us the unconstrained case.

17.1.1 Gradient

For most of the following discussion, we assume that the function f is differentiable. In that case, we can give an equivalent characterization, based on the notion of the *gradient* $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$.

The *directional derivative* of f at x (in the direction y) is defined as

$$f'(x; y) := \lim_{\varepsilon \to 0} \frac{f(x + \varepsilon y) - f(x)}{\varepsilon}.$$

If there exists a vector g such that $\langle g, y \rangle = f'(x; y)$ for all y, then f is called *differentiable* at x, and g is called the *gradient*. It follows that the gradient must be of the form

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \cdots, \frac{\partial f}{\partial x_n}(x) \right).$$

Fact 17.3 (First-order condition). A function $f : K \to \mathbb{R}$ is convex if and only if

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle, \tag{17.3}$$

for all $x, y \in K$.

Geometrically, Fact 17.3 states that the function always lies above its tangent plane, for all points in K. If the function f is twice-differentiable, and if $H_f(x)$ is its *Hessian matrix*, i.e. its matrix of second derivatives at $x \in K$:

$$(H_f)_{i,j}(x) := \frac{\partial^2 f}{\partial x_i \, \partial x_j}(x), \tag{17.4}$$

then we get yet another characterization of convex functions.

Fact 17.4 (Second-order condition). A twice-differentiable function f is convex if and only if $H_f(x)$ is positive semidefinite for all $x \in K$.

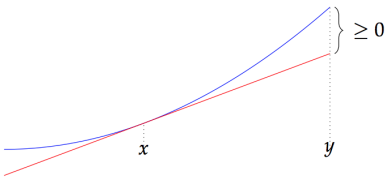


Figure 17.1: The blue line denotes the function and the red line is the tangent line at x. (Figure from Nisheeth Vishnoi.)

17.1.2 Lipschitz Functions

We will need a notion of "niceness" for functions:

Definition 17.5 (Lipschitz continuity)**.** For a convex set $K \subseteq \mathbb{R}^n$, a function $f : K \to \mathbb{R}$ is called *G-Lipschitz* (or *G-Lipschitz continuous*) with respect to the norm $\|\cdot\|$ if

$$|f(x) - f(y)| \leq G \|x - y\|,$$

for all $x, y \in K$.

In this chapter we focus on the Euclidean or ℓ_2-norm, denoted by $\|\cdot\|_2$. General norms arise in the next chapter, when we talk about mirror descent. Again, assuming that the function is differentiable allows us to give an alternative characterization of Lipschitzness.

Fact 17.6. A differentiable function $f : K \to \mathbb{R}^n$ is G-Lipschitz with respect to $\|\cdot\|_2$ if and only if

$$\|\nabla f(x)\|_2 \leq G, \tag{17.5}$$

for all $x \in K$.

17.2 Unconstrained Convex Minimization

If the function f is convex, any *stationary point* (i.e., a point x^* where $\nabla f(x^*) = 0$) is also a *global minimum*: just use Fact 17.3 to infer that $f(y) \geq f(x^*)$ for all y. Now given a convex function, we can just solve the equation

$$\nabla f(x) = 0$$

to compute the global minima exactly. This is often easier said than done: for instance, if the function f we want to minimize may not be given explicitly. Instead we may only have a gradient oracle that given x, returns $\nabla f(x)$.

Even when f is explicit, it may be expensive to solve the equation $\nabla f(x) = 0$, and gradient descent may be a faster way. One example arises when solving linear systems: given a quadratic function $f(x) = \frac{1}{2}x^\intercal Ax - bx$ for a symmetric matrix A (say having full rank), a simple calculation shows that

$$\nabla f(x) = 0 \iff Ax = b \iff x = A^{-1}b.$$

This can be solved in $O(n^\omega)$ (i.e., matrix-multiplication) time using Gaussian elimination—but for "nice" matrices A we are often able to approximate a solution much faster using the gradient-based methods we will soon see.

17.2.1 The Basic Gradient Descent Method

Gradient descent is an iterative algorithm to approximate the optimal solution x^*. The main idea is simple: since the gradient tells us the direction of steepest increase, we'd like to move opposite to the direction of the gradient to decrease the fastest. So by selecting an initial position x_0 and a step size η_t at each time t, we can repeatedly perform the update:

$$x_{t+1} \leftarrow x_t - \eta_t \cdot \nabla f(x_t). \tag{17.6}$$

There are many choices to be made: where should we start? What are the step sizes? When do we stop? While each of these decisions depend on the properties of the particular instance at hand, we can show fairly general results for general convex functions.

17.2.2 *An Algorithm for General Convex Functions*

The algorithm fixes a step size for all times t, performs the update (17.6) for some number of steps T, and then returns the average of all the points seen during the process.

Algorithm 14: Gradient Descent

14.1 $x_1 \leftarrow$ starting point
14.2 **for** $t \leftarrow 1$ **to** T **do**
14.3 $\quad x_{t+1} \leftarrow x_t - \eta \cdot \nabla f(x_t)$
14.4 **return** $\widehat{x} := \frac{1}{T}\sum_{t=1}^{T} x_i$.

This is easy to visualize in two dimensions: draw the level sets of the function f, and the gradient at a point is a scaled version of normal to the tangent line at that point. Now the algorithm's path is often a zig-zagging walk towards the optimum (see Fig 17.2).

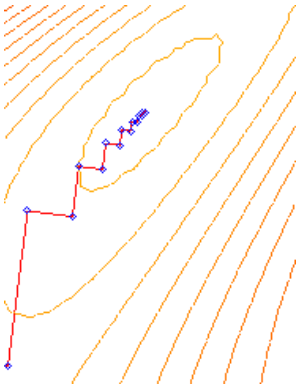


Figure 17.2: The yellow lines denote the level sets of the function f and the red walk denotes the steps of gradient descent. (Figure from Wikipedia.)

Interestingly, we can give rigorous bounds on the convergence of this algorithm to the optimum, based on the distance of the starting point from the optimum, and bounds on the Lipschitzness of the function. If both these are assumed to be constant, then our error is smaller than ε in only $O(1/\varepsilon^2)$ steps.

Proposition 17.7. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex, differentiable and G-Lipschitz. Let x^* be any point in $\mathbb{R}^d$. If we define $T := \frac{G^2\|x_0 - x^*\|^2}{\varepsilon^2}$ and $\eta := \frac{\|x_0 - x^*\|}{G\sqrt{T}}$, then the solution $\widehat{x}$ returned by gradient descent satisfies*

$$f(\widehat{x}) \leq f(x^*) + \varepsilon. \tag{17.7}$$

In particular, this holds when x^* is a minimizer of f.

The core of this proposition lies in the following theorem

Theorem 17.8. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex, differentiable and G-Lipschitz. Then the gradient descent algorithm ensures that*

$$\sum_{t=1}^{T} f(x_t) \leq \sum_{t=1}^{T} f(x^*) + \frac{1}{2}\eta T G^2 + \frac{1}{2\eta}\|x_0 - x^*\|^2. \tag{17.8}$$

We will prove Theorem 17.8 in the next section, but let's first use it to prove Proposition 17.7, our guarantee on the offline convergence of vanilla gradient descent.

Proof of Proposition 17.7. By definition of $\widehat{x}$ and the convexity of f,

$$f(\widehat{x}) = f\Big(\frac{1}{T}\sum_{t=1}^{T} x_t\Big) \le \frac{1}{T}\sum_{t=1}^{T} f(x_t).$$

By Theorem 17.8,

$$\frac{1}{T}\sum_{t=1}^{T} f(x_t) \le f(x^*) + \underbrace{\frac{1}{2}\eta G^2 + \frac{1}{2\eta T}\|x_0 - x^*\|^2}_{\text{error}}.$$

The error terms balance when $\eta = \frac{\|x_0 - x^*\|}{G\sqrt{T}}$, giving

$$f(\widehat{x}) \le f(x^*) + \frac{\|x_0 - x^*\| G}{\sqrt{T}}.$$

Finally, we set $T = \frac{1}{\varepsilon^2} G^2 \|x_0 - x^*\|^2$ to obtain

$$f(\widehat{x}) \le f(x^*) + \varepsilon. \qquad \square$$

Observe: we do not (and cannot) show that the point $\widehat{x}$ is close in distance to x^*; we just show that the function value $f(\widehat{x}) \approx f(x^*)$. Indeed, if the function is very flat close to x^* and we start off at some remote point, we make tiny steps as we get close to x^*, and we cannot hope to get close to it.

The $1/\varepsilon^2$ dependence of the number of oracle calls was shown to be tight for gradient-based methods by Yurii Nesterov, if we allow f to be any G-Lipschitz function. However, if we assume that the function is "well-behaved", we can indeed improve on the $1/\varepsilon^2$ dependence. Moreover, if the function is strongly convex, we can show that x^* and $\widehat{x}$ are close to each other as well: see §17.5 for such results.

The convergence guarantee in Proposition 17.7 is for the time-averaged point $\widehat{x}$. Indeed, using a fixed step size means that our iterates may get stuck in a situation where $x_{t+2} = x_t$ after some point and hence we never improve, even though $\widehat{x}$ is at the minimizer. One can also show that $f(x_T) \le f(x^*) + \varepsilon$ if we use a time-varying step size $\eta_t = O(1/\sqrt{t})$, and increase the time horizon slightly to $O(1/\varepsilon^2 \log 1/\varepsilon)$. We refer to the work of Shamir and Zhang.

17.2.3 *Proof of Theorem 17.8*

Like in the proof of the multiplicative weights algorithm, we will use a potential function. Define

$$\Phi_t := \frac{\|x_t - x^*\|^2}{2\eta}. \tag{17.9}$$

We start the proof of Theorem 17.8 by understanding the one-step change in the potential:

Lemma 17.9 (Change in Potential)**.**

$$\Phi_{t+1} - \Phi_t \leq \langle \nabla f(x_t), x^* - x_t \rangle + \frac{\eta}{2} \|\nabla f(x_t)\|^2.$$

Proof. Using the identity

$$\|a + b\|^2 = \|a\|^2 + 2\langle a, b \rangle + \|b\|^2,$$

with $a + b = x_{t+1} - x^*$ and $a = x_t - x^*$, we get

$$\begin{aligned} \Phi_{t+1} - \Phi_t &= \frac{1}{2\eta}\left(\|x_{t+1} - x^*\|^2 - \|x_t - x^*\|^2\right) \qquad (17.10) \\ &= \frac{1}{2\eta}\Big(2\underbrace{\langle x_{t+1} - x_t, x_t - x^* \rangle}_{\langle b, a \rangle} + \underbrace{\|x_{t+1} - x_t\|^2}_{\|b\|^2}\Big); \end{aligned}$$

now using $x_{t+1} - x_t = -\eta \nabla f(x_t)$ from gradient descent,

$$= \frac{1}{2\eta}\left(2\langle -\eta \nabla f(x_t), x_t - x^* \rangle + \|\eta \nabla f(x_t)\|^2\right).$$

Now rearranging terms proves the lemma. □

Now that we understand how our potential changes over time, proving the theorem is straightforward.

Proof of Theorem 17.8. We start with the inequality we proved above, and use that since f is G-Lipschitz, $\|\nabla f(x)\| \leq G$ for all x. Thus,

$$f(x_t) + (\Phi_{t+1} - \Phi_t) = f(x_t) + \langle \nabla f(x_t), x^* - x_t \rangle + \frac{\eta}{2} G^2.$$

Since f is convex, we know that $f(x_t) + \langle \nabla f(x_t), x^* - x_t \rangle \leq f(x^*)$. Thus, we conclude that

$$f(x_t) + (\Phi_{t+1} - \Phi_t) \leq f(x^*) + \frac{\eta}{2} G^2.$$

Summing over $t = 1, \ldots, T$,

$$\sum_{t=1}^{T} f(x_t) + \sum_{t=1}^{T} (\Phi_{t+1} - \Phi_t) \leq \sum_{t=1}^{T} f(x^*) + \frac{\eta}{2} G^2 T$$

The sum of potentials on the left telescopes to give:

$$\sum_{t=1}^{T} f(x_t) + \Phi_{T+1} - \Phi_1 \leq \sum_{t=1}^{T} f(x^*) + \frac{\eta}{2} G^2 T$$

Since the potentials are nonnegative, we can drop the Φ_T term:

$$\sum_{t=1}^{T} f(x_t) - \Phi_1 \leq \sum_{t=1}^{T} f(x^*) + \frac{\eta}{2} G^2 T$$

Substituting in the definition of Φ_1 and moving it over to the right hand side completes the proof. □

17.2.4 *Some Remarks on the Algorithm*

We assume a gradient oracle for the function: given a point x, it returns the gradient $\nabla f(x)$ at that point. If the function f is not given explicitly, we may have to estimate the gradient using, e.g., random sampling. One particularly sample-efficient solution is to pick a uniformly random point $u \sim \mathbb{S}^{n-1}$ from the sphere in $\mathbb{R}^n$, and return

$$d\left[\frac{f(x+\delta u)}{\delta} u\right]$$

As $\delta \to 0$, the expectation of this expression tends to $\nabla f(x)$, using Stokes' theorem.

for some tiny $\delta > 0$. It is slightly mysterious, so perhaps it is useful to consider its expectation in the case of a univariate function:

$$\mathbb{E}_{u\sim\{-1,+1\}}\left[\frac{f(x+\delta u)}{\delta}\, u\right] = \frac{f(x+\delta) - f(x-\delta)}{2\delta} \approx f'(x).$$

In general, randomized strategies form the basis of *stochastic gradient descent*, where we use an unbiased estimator of the gradient, instead of computing the gradient itself (because it is slow to compute, or because enough information is not available). The challenge is now to control the variance of this estimator.

Another concern is that the step-size η and the number of steps T both require knowledge of the distance $\|x_1 - x^*\|$ as well as the bound on the gradient. More here. As an exercise, show that using the time-varying step-size $\eta_t := \frac{\|x_0 - x^*\|}{G\sqrt{t}}$ also gives a very similar convergence rate.

Finally, the guarantee is for $f(\widehat{x})$, where $\widehat{x}$ is the time-average of the iterates. What about returning the final iterate? It turns out this has comparable guarantees, but the proof is slightly more involved. Add references.

17.3 *Constrained Convex Minimization*

Unlike the unconstrained case, the gradient at the minimizer may not be zero in the constrained case—it may be at the boundary. In this case, the condition for a convex function $f : K \to \mathbb{R}$ to be minimized at $x^* \in K$ is now

This is the analog of the minimizer of a single variable function being achieved either at a point where the derivative is zero, or at the boundary.

$$\langle \nabla f(x^*), y - x^* \rangle \geq 0 \qquad \text{for all } y \in K. \tag{17.11}$$

In other words, all vectors $y - x^*$ pointing within K are "positively correlated" with the gradient.

When x^* is in the interior of K, the condition (17.11) is equivalent to $\nabla f(x^*) = 0$.

17.3.1 *Projected Gradient Descent*

While the gradient descent algorithm still makes sense: moving in the direction opposite to the gradient still moves us towards lower

function values. But we must change our algorithm to ensure that the new point x_{t+1} lies within K. To ensure this, we simply project the new iterate x_{t+1} back onto K. Let $\text{proj}_K : \mathbb{R}^n \to K$ be defined as

$$\text{proj}_K(y) = \arg\min_{x \in K} \|x - y\|_2 .$$

The modified algorithm is given below in Algorithm 15, with the changes highlighted in blue.

Algorithm 15: Projected Gradient Descent For CCM

15.1 $x_1 \leftarrow$ starting point
15.2 **for** $t \leftarrow 1$ **to** T **do**
15.3 $\quad x'_{t+1} \leftarrow x_t - \eta \cdot \nabla f(x_t)$
15.4 $\quad x_{t+1} \leftarrow \text{proj}_K(x'_{t+1})$
15.5 **return** $\widehat{x} := \frac{1}{T} \sum_{t=1}^{T} x_t$

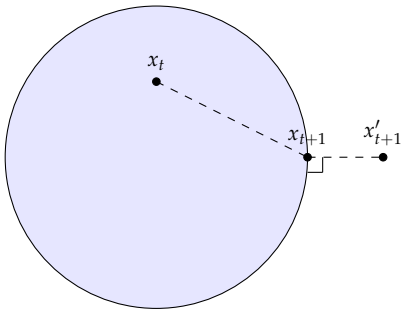


Figure 17.3: Projection onto a convex body

We will show below that a result almost identical to that of Theorem 17.8, and hence that of Proposition 17.7 holds.

Proposition 17.10. *Let K be a closed convex set, and $f : K \to \mathbb{R}$ be convex, differentiable and G-Lipschitz. Let $x^* \in K$, and define $T := \frac{G^2 \|x_0 - x^*\|^2}{\varepsilon^2}$ and $\eta := \frac{\|x_0 - x^*\|}{G\sqrt{T}}$. Then the solution $\widehat{x}$ returned by projected gradient descent satisfies*

$$f(\widehat{x}) \leq f(x^*) + \varepsilon. \tag{17.12}$$

In particular, this holds when x^* is a minimizer of f.

Proof. We can reduce to an analogous constrained version of Theorem 17.8. Let us start the proof as before:

$$\Phi_{t+1} - \Phi_t = \frac{1}{2\eta}\left(\|x_{t+1} - x^*\|^2 - \|x_t - x^*\|^2\right) \tag{17.13}$$

But x_{t+1} is the projection of x'_{t+1} onto K, which is difficult to reason about. Also, we know that $-\eta \nabla f(x_t) = x'_{t+1} - x^*$, not $x_{t+1} - x^*$, so we would like to move to the point x'_{t+1}. Indeed, we claim that $\|x'_{t+1} - x^*\| \geq \|x_{t+1} - x^*\|$, and hence we get

$$\Phi_{t+1} - \Phi_t = \frac{1}{2\eta}\left(\|x'_{t+1} - x^*\|^2 - \|x_t - x^*\|^2\right). \tag{17.14}$$

Now the rest of the proof of Theorem 17.8 goes through unchanged.

Why is the claim $\|x'_{t+1} - x^*\| \geq \|x_{t+1} - x^*\|$ true? Since K is convex, projecting onto it *gets us closer to every point* in K, in particular to $x^* \in K$. To formally prove this fact about projections, consider the angle $x^* \to x_{t+1} \to x'_{t+1}$. This is a non-acute angle, since the orthogonal projection means K likes to one side of the hyperplane defined by the vector $x'_{t+1} - x_{t+1}$, as in the figure on the right. □

Note that restricting the play to K can be helpful in two ways: we can upper-bound the distance $\|x^* - x_1\|$ by the diameter of K, and moreover we need only consider the Lipschitzness of f for points within K.

17.4 Online Gradient Descent, and Relationship with MW

We considered gradient descent for the *offline* convex minimization problem, but one can use it even when the function changes over time. Indeed, consider the *online convex optimization (OCO)* problem: at each time step t, the algorithm proposes a point $x_t \in K$ and an adversary gives a function $f_t : K \to \mathbb{R}$ with $\|\nabla f_t\| \leq G$. The cost of each time step is $f_t(x_t)$ and your objective is to minimize

$$\text{regret} = \sum_t f_t(x_t) - \min_{x^* \in K} \sum_t f_t(x^*).$$

For instance if $K = \Delta_n$, and $f_t(x) := \langle \ell_t, x \rangle$ for some loss vector $\ell_t \in [-1, 1]^n$, then we are back in the experts setting of the previous chapters. Of course, the OCO problem is far more general, allowing arbitrary convex functions.

Surprisingly, we can use the almost same algorithm to solve the OCO problem, with one natural modification: the update rule is now taken with respect to gradient of the *current* function f_t:

This was first observed by Martin Zinkevich in 2002, when he was a Ph.D. student here at CMU.

$$x_{t+1} \leftarrow x_t - \eta \cdot \nabla f_t(x_t).$$

Looking back at the proof in §17.2, the proof of Lemma 17.9 immediately extends to give us

$$f_t(x_t) + \Phi_{t+1} - \Phi_t \leq f_t(x^*) + \frac{1}{2}\eta G^2.$$

Now summing this over all times t gives

$$\begin{aligned}\sum_{t=1}^{T} \left(f_t(x_t) - f_t(x^*)\right) &\leq \sum_{t=1}^{T} \left(\Phi_t - \Phi_{t+1}\right) + \frac{\eta}{2} T G^2 \\ &\leq \Phi_1 + \frac{1}{2}\eta T G^2,\end{aligned}$$

since $\Phi_{T+1} \geq 0$. The proof is now unchanged: setting $T \geq \frac{\|x_1 - x^*\|^2 G^2}{\varepsilon^2}$ and $\eta = \frac{\|x_1 - x^*\|}{G\sqrt{T}}$, and doing some elementary algebra as above,

$$\frac{1}{T} \sum_{t=0}^{T} \left(f_t(x_t) - f_t(x^*)\right) \leq \frac{\|x_1 - x^*\| G}{\sqrt{T}} \leq \varepsilon.$$

17.4.1 Comparison to the MW/Hedge Algorithms

One advantage of the gradient descent approach (and analysis) over the multiplicative weight-based ones is that the guarantees here hold

for all convex bodies K and all convex functions, as opposed to being just for the unit simplex Δ_n and linear losses $f_t(x) = \langle \ell_t, x \rangle$, say for $\ell_t \in [-1,1]^n$. However, in order to make a fair comparison, suppose we restrict ourselves to Δ_n and linear losses, and consider the number of rounds T before we get an average regret of ε.

- If we consider $\|x_1 - x^*\|$ (which, in the worst case, is the diameter of K), and G (which is an upper bound on $\|\nabla f_t(x)\|$ over points in K) as constants, then the $T = \Theta(\frac{1}{\varepsilon^2})$ dependence is the same.

- For a more quantitative comparison, note that $\|x_1 - x^*\| \leq \sqrt{2}$ for $x_1, x^* \in \Delta_n$, and $\|\nabla f_t(x)\| = \|\ell_t\| \leq \sqrt{n}$ for $\ell_t \in [-1,1]^n$. Hence, Proposition 17.10 gives us $T = \Theta\big(\frac{\sqrt{n}}{\varepsilon^2}\big)$, as opposed to $T = \Theta\big(\frac{\log n}{\varepsilon^2}\big)$ for multiplicative weights.

The problem, at a high level, is that we are "choosing the wrong norm": when dealing with probabilities, the "right" norm is the ℓ_1 norm and not the Euclidean ℓ_2 norm. In the next lecture we will formalize what this means, and how this dependence on n be improved via the Mirror Descent framework.

17.5 Stronger Assumptions

If the function f is "well-behaved", we can improve the guarantees for gradient descent in two ways: we can reduce the dependence on ε, and we can weaken (or remove) the dependence on the parameters G and $\|x_1 - x^*\|$. There are two standard assumptions to make on the convex function: that it is "not too flat" (captured by the idea of *strong convexity*), and it is not "not too curved" (i.e., it is *smooth*). We now use these assumptions to improve the guarantees.

17.5.1 Strongly-Convex Functions

Definition 17.11 (Strong Convexity)**.** A function $f : K \to \mathbb{R}$ is α-*strongly convex* if for all $x, y \in K$, any of the following holds:

1. (Zeroth order) $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) - \frac{\alpha}{2}\lambda(1-\lambda)\|x-y\|^2$ for all $\lambda \in [0,1]$.

2. (First order) If f is differentiable, then

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2}\|x-y\|^2 . \tag{17.15}$$

3. (Second order) If f is twice-differentiable, then all eigenvalues of $H_f(x)$ are at least α at every point $x \in K$.

We will work with the first-order definition, and show that the gradient descent algorithm with (time-varying) step size $\eta_t = O(\frac{1}{\alpha t})$ converges to a value at most $f(x^*) + \varepsilon$ in time $T = \Theta(\frac{G^2}{\alpha\varepsilon})$. Note there is no more dependence on the diameter of the polytope. Before we give this proof, let us give the other relevant definitions.

17.5.2 Smooth Functions

Definition 17.12 (Lipschitz Smoothness)**.** A function $f : K \to \mathbb{R}$ is *β-(Lipschitz)-smooth* if for all $x, y \in K$, any of the following holds:

1. (Zeroth order) $f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda)f(y) - \frac{\beta}{2}\lambda(1-\lambda)\|x-y\|^2$ for all $\lambda \in [0,1]$.
2. (First order) If f is differentiable, then
$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{\beta}{2}\|x-y\|^2. \tag{17.16}$$
3. (Second order) If f is twice-differentiable, then all eigenvalues of $H_f(x)$ are at most β at every point $x \in K$.

In this case, the gradient descent algorithm with fixed step size $\eta_t = \eta = O(\frac{1}{\beta})$ yields an $\widehat{x}$ which satisfies $f(\widehat{x}) - f(x^*) \leq \varepsilon$ when $T = \Theta(\frac{\beta\|x_1 - x^*\|}{\varepsilon})$. In this case, note we have no dependence on the Lipschitzness G any more; we only depend on the diameter of the polytope. Again, we defer the proof for the moment.

17.5.3 Well-conditioned Functions

Functions that are both β-smooth and α-strongly convex are called *well-conditioned functions*. From the facts above, the eigenvalues of their Hessian H_f must lie in the interval $[\alpha, \beta]$ at all points $x \in K$. In this case, we get a much stronger convergence—we can achieve ε-closeness in time $T = \Theta(\log \frac{1}{\varepsilon})$, where the constant depends on the *condition number* $\kappa = \beta/\alpha$.

Theorem 17.13. *For a function f which is β-smooth and α-strongly convex, let x^* be the solution to the unconstrained convex minimization problem* $\arg\min_{x\in\mathbb{R}^n} f(x)$*. Then running gradient descent with $\eta_t = 1/\beta$ gives*

$$f(x_t) - f(x^*) \leq \frac{\beta}{2}\exp\left(\frac{-t}{\kappa}\right)\|x_1 - x^*\|^2.$$

Proof. For β-smooth f, we can use Definition 17.12 to get

$$f(x_{t+1}) \leq f(x_t) - \eta\|\nabla f(x_t)\|^2 + \eta^2\frac{\beta}{2}\|\nabla f(x_t)\|^2.$$

The right hand side is minimized by setting $\eta = \frac{1}{\beta}$, when we get

$$f(x_{t+1}) - f(x_t) \leq -\frac{1}{2\beta}\|\nabla f(x_t)\|^2. \tag{17.17}$$

For α-strongly-convex f, we can use Definition 17.11 to get:

$$\begin{aligned} f(x_t) - f(x^*) &\leq \langle \nabla f(x_t), x_t - x^* \rangle - \frac{\alpha}{2}\|x_t - x^*\|^2, \\ &\leq \|\nabla f(x_t)\| \, \|x_t - x^*\| - \frac{\alpha}{2}\|x_t - x^*\|^2, \\ &\leq \frac{1}{2\alpha}\|\nabla f(x_t)\|^2, \end{aligned} \tag{17.18}$$

where we use that the right hand side is maximized when $\|x_t - x^*\| = \|\nabla f(x_t)\| / \alpha$. Now combining with (17.17) we have that

$$f(x_{t+1}) - f(x_t) \leq -\frac{\alpha}{\beta}\Big(f(x_t) - f(x^*)\Big), \tag{17.19}$$

or setting $\Delta_t = f(x_t) - f(x^*)$ and rearranging, we get

$$\Delta_{t+1} \leq \left(1 - \frac{\alpha}{\beta}\right)\Delta_t \leq \left(1 - \frac{1}{\kappa}\right)^t \Delta_1 \leq \exp\left(-\frac{t}{\kappa}\right) \cdot \Delta_1.$$

We can control the value of Δ_1 by using (17.16) in $x = x^*, y = x_1$; since $\nabla f(x^*) = 0$, get $\Delta_1 = f(x_1) - f(x^*) \leq \frac{\beta}{2}\|x_1 - x^*\|^2$. □

Strongly-convex (and hence well-conditioned) functions have the nice property that if $f(x)$ is close to $f(x^*)$ then x is close to x^*: intuitively, since the function is curving at least quadratically, the function values at points far from the minimizer must be significant. Formally, use (17.15) with $x = x^*, y = x_t$ and the fact that $\nabla f(x^*) = 0$ to get

$$\|x_t - x^*\|^2 \leq \frac{2}{\alpha}(f(x_t) - f(x^*)).$$

We leave it as an exercise to show the claimed convergence bounds using just strong convexity, or just smoothness. (Hint: use the statements proved in (17.17) and (17.18).

Before we end, a comment on the strong $O(\log 1/\varepsilon)$ convergence result for well-conditioned functions. Suppose the function values lies in $[0, 1]$. The $\Theta(\log 1/\varepsilon)$ error bound means that we are correct up to b bits of precision—i.e., have error smaller than $\varepsilon = 2^{-b}$—after $\Theta(b)$ steps. In other words, the number of bits of precision is linear in the number of iterations. The optimization literature refers to this as *linear convergence*, which can be confusing when you first see it.

17.6 Extensions and Loose Ends

17.6.1 Subgradients

What if the convex function f is not differentiable? Staring at the proofs above, all we need is the following:

Definition 17.14 (Subgradient)**.** A vector z_x is called a *subgradient* at point x if

$$f(y) \geq f(x) + \langle z_x, y - x \rangle \qquad \text{for all } y \in \mathbb{R}^n.$$

Now we can use subgradients at the point x wherever we used $\nabla f(x)$, and the entire proof goes through. In some cases, an approximate subgradient may also suffice.

17.6.2 Stochastic Gradients, and Coordinate Descent

17.6.3 Acceleration

17.6.4 Reducing to the Well-conditioned Case