In this lecture, we will study the gradient descent algorithm and analyze it in the context of convex optimization.

1 Preliminaries

First, recall the following definitions:

Definition 16.1 (Convex Set)**.** A set $K \subseteq \mathbb{R}^n$ is *convex* iff

$$\big(\lambda x + (1-\lambda)y\big) \in K \qquad \forall x, y \in K, \forall \lambda \in [0,1].$$

Definition 16.2 (Convex Function)**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* iff

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \qquad \forall x, y \in \mathbb{R}^n, \forall \lambda \in [0,1].$$

In the context of this lecture, we will always assume that the function f is differentiable.

Fact 16.3 (First-Order condition)**.** *A function f is convex iff $\forall x, y$ $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$*

Geometrically, Fact 16.3 states that the function always lies above its tangent (see Fig 16.1).

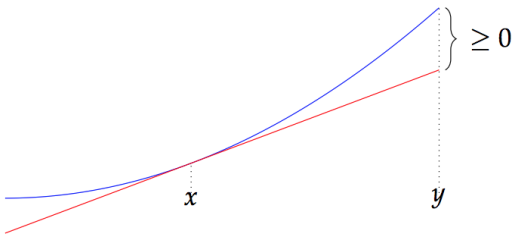

Figure 16.1: The blue line denotes the function and the red line is the tangent at x. [VT]

If the function f is also twice differentiable, then we denote by $\nabla^2 f$ its matrix of second derivatives (a.k.a. its Hessian), where $(\nabla^2 f)_{i,j} := \frac{\partial^2 f}{\partial x_i \partial x_j}$.

Fact 16.4 (Second-order condition)**.** *A twice-differentiable function f is convex iff $\nabla^2 f \succeq 0$.* [1]

Definition 16.5 (Lipschitz)**.** Function $f : \mathbb{R}^n \to \mathbb{R}$ is *G-Lipschitz* with respect to the norm $\|\cdot\|$ if

$$|f(x) - f(y)| \leq G \|x - y\| \qquad \forall x, y \in \mathbb{R}^n.$$

For today, we will only work with the ℓ_2-norm $\|\cdot\|_2$.

Fact 16.6. *A differentiable function f is G-Lipschitz w.r.t. $\|\cdot\|_2$ iff $\|\nabla f(x)\|_2 \leq G$ for all $x \in \mathbb{R}^n$.*

[1] The notation $A \succeq B$ signifies $A - B$ is positive semidefinite; the ordering given by $\succeq$ is called the Löwner ordering.

2 Convex Minimization and Gradient Descent

There are two kinds of problems that we will study.

1. Unconstrained Convex Minimization (UCM): Given a convex function f, find
$$\min_{x \in \mathbb{R}^n} f(x).$$
2. Constrained Convex Minimization (CCM): Given a convex function f and convex set K, find
$$\min_{x \in K} f(x).$$
This is a more general problem, since setting $K = \mathbb{R}^n$ gives us the unconstrained case.

2.1 Unconstrained Convex Minimization

One useful property of convex functions is that f is convex implies that all local minima are also global minima. Hence, solving
$$\nabla f(x) = 0$$
would enable us to compute the global minima exactly. Quite often, it is not possible to solve $\nabla f = 0$. For instance, the function f may not be given explicitly, but we may be given an oracle to compute gradients at any point. Or even when we can write down and solve $\nabla f = 0$, it may be too expensive, and gradient descent may be a faster way to get better. One example is in *solving linear systems*: when $f(x) = \frac{1}{2}x^\intercal Ax - bx$, we have that $\nabla f(x) = 0 \iff Ax = b \iff x = A^{-1}b$, which can be solved in $O(n^\omega)$ (i.e., matrix-multiplication) time—but for "nice" matrices A we may want to approximate a solution much faster.

However, we can still iteratively approximate the optimal solution x^*. The main idea is simple: the gradient tells us the direction of steepest increase, so to decrease the fastest we'd like to move opposite to the direction of the gradient. Of course, we'd like to take an infinitesimal step in this negative gradient direction, recompute the gradient, etc. To make this a finite algorithm, we define a *step size* η_t, and hope that changing our current point x by $-\eta_t \nabla f(x)$ decreases the function value. This gives us the classical gradient descent algorithm.[2]

Algorithm 1: Gradient Descent

$x_0 \leftarrow$ starting point;
for $t \leftarrow 0$ **to** $T - 1$ **do**
  $x_{t+1} \leftarrow x_t - \eta_t \cdot \nabla f(x_t)$;
end
Result: $\hat{x} = \frac{1}{T}\sum_{t=0}^{T-1} x_i$

Some comments: in 2-dimensions, this is easy to visualize, since we can draw the level sets of the function f, and the gradient at a point is the normal to the tangent line at that point. The algorithm's path may be a zig-zagging walk towards the optimum goal (see Fig 16.2).

Theorem 16.7. *For any convex, G-Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}$, if $\|x^* - x_0\| \leq D$, then $f(\hat{x}) - f(x^*) \leq \varepsilon$ if we set $T = \left(\frac{GD}{\varepsilon}\right)^2$ and $\eta_t = \eta = \frac{\varepsilon}{G^2}$.*

We will use the following elementary fact in the proof
$$\langle a, b \rangle = \frac{1}{2}\Big(\|a\|^2 + \|b\|^2 - \|a - b\|^2 \Big) \tag{16.1}$$

[2] We have defined neither the start point x_0 nor the step sizes η_t, but we will fix that soon.

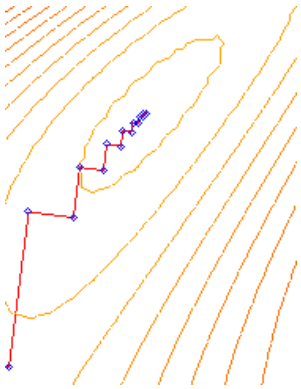

Figure 16.2: The yellow lines denote the level sets of the function f and the red walk denotes the steps of gradient descent. [Com06]

Proof. We will use a potential

$$\Phi_t = \frac{1}{2\eta} \|x_t - x^*\|^2, \tag{16.2}$$

which is positive for all t. We will soon show that, for some upper bound B,

$$f(x_t) - f(x^*) + \Phi_{t+1} - \Phi_t \leq B. \tag{16.3}$$

Before we prove this, let us see what this implies. Using convexity of f and averaging the above over all t, we get

$$\begin{aligned}
f(\hat{x}) - f(x^*) &= f\left(\frac{1}{T}\sum_{t=0}^{T-1} x_t\right) - f(x^*) && \text{by definition of } \hat{x} \\
&\leq \frac{1}{T}\sum_{t=0}^{T-1}(f(x_t) - f(x^*)) && \text{because } f \text{ is convex} \\
&\leq B + \frac{1}{T}(\Phi_0 - \Phi_T) && \text{average (16.3), the } \Phi_t \text{ terms telescope} \\
&\leq B + \frac{D^2}{2\eta T} = B + \frac{D^2}{2\eta T} && \text{drop } -\Phi_T \text{ and expand } \Phi_0.
\end{aligned} \tag{16.4}$$

Now we just need to compute B. The potentials use differences of the form $x_t - x^*$. The key is to express as much as possible in terms of $x_{t+1} - x_t$, because

$$x_{t+1} - x_t = -\eta \nabla f(x_t). \tag{16.5}$$

by the definition of the algorithm.

$$\begin{aligned}
&f(x_t) - f(x^*) + \Phi_{t+1} - \Phi_t \\
&\quad = f(x_t) - f(x^*) + \frac{1}{2\eta}(\|x_{t+1} - x^*\|^2 - \|x_t - x^*\|^2) && (16.6) \\
&\quad = f(x_t) - f(x^*) + \frac{1}{2\eta}(2\langle x_{t+1} - x_t, x_t - x^*\rangle + \|x_{t+1} - x_t\|^2) && \text{use (16.1)} \\
&\quad = f(x_t) - f(x^*) - \langle \nabla f(x_t), x_t - x^*\rangle + \frac{\eta}{2}\|\nabla f(x_t)\|^2 && \text{use (16.5)} \\
&\quad \leq 0 + \frac{\eta}{2}\|\nabla f(x_t)\|^2 && \text{Convexity and Fact 16.3} \\
&\quad \leq \frac{\eta G^2}{2} && f \text{ is } G\text{-Lipschitz.}
\end{aligned}$$

Putting it together with (16.4), we have

$$f(\hat{x}) - f(x^*) \leq \frac{1}{2}\left(\eta G^2 + \frac{D^2}{\eta T}\right).$$

We want this to be at most ε, so we definitely need $\eta = O\left(\frac{\varepsilon}{G^2}\right)$. It is simple to verify that $\eta = \frac{\varepsilon}{G^2}$ and $T = \frac{D^2}{\varepsilon\eta} = \left(\frac{GD}{\varepsilon}\right)^2$ suffices. □

This analysis, and in particular the $1/\varepsilon^2$ dependence on ε is tight if we just assume f is G-Lipschitz. Moreover, we did not (and cannot) show that $\hat{x}$ is close in distance to x^*; we just show that $f(\hat{x}) \approx f(x^*)$. Indeed, if the function is very flat close to the origin, we cannot hope to be close in distance. (To improve on the $1/\varepsilon^2$ dependence, or to show physical closeness of x^* and $\hat{x}$, we need further assumptions; see Section 4.)

2.2 Constrained Convex Minimization

Unlike the unconstrained case, now the derivative may not be 0 at the optimum. Nonetheless, the main idea of gradient descent still yields a good algorithm. Here is some intuition why. When f is convex, its local optima are global optima, and

$$x^* \text{ is a local minimum} \quad \Leftrightarrow \quad \nabla f(x^*) = 0 \quad \Leftrightarrow \quad \forall y \in \mathbb{R}^n,\ \langle \nabla f(x^*), y - x^* \rangle \geq 0.$$

(The rightmost property is essentially that $\langle a, b\rangle = 0$ for all b if and only if $a = 0$.)

When we constrain our domain to convex set K, the minimum may not have gradient zero. However, if the minimum doesn't have gradient zero, it's on the boundary of K. Either way, we can show that x^* is a local minimum iff

$$\langle \nabla f(x^*), y - x^* \rangle \geq 0 \qquad \forall y \in K.$$

When x^* is in the interior of K, this is equivalent to $\nabla f(x^*) = 0$, but this is not so when x^* is on the boundary of K because we can't "test every direction's dot product". Here's another interpretation of the statement: starting from x^*, if we walk a little bit in some direction but stay in K, then f should increase. This means stepping in the reverse direction of the gradient is still a good idea!

2.2.1 Projected Gradient Descent

We need change our algorithm to ensure that the new point x_{t+1} lies within K. To ensure this, we simply project each step back onto K. Let $\Pi_K : \mathbb{R}^n \to K$ be defined as

$$\Pi_K(y) = \arg\min_{x \in K} \|x - y\|.$$

The modified algorithm is given below in Algorithm 2, with the changes highlighted in blue.

Algorithm 2: Gradient Descent For CCM

$x_0 \leftarrow$ starting point;
for $t \leftarrow 0$ **to** $T - 1$ **do**
  $x'_{t+1} \leftarrow x_t - \eta_t \cdot \nabla f(x_t)$;
  $x_{t+1} \leftarrow \Pi_K(x'_{t+1})$;
end
Result: $\hat{x} = \frac{1}{T}\sum_{t=0}^{T-1} x_i$

We will show below that a theorem (and analysis) similar to that of Theorem 16.7 holds.

Theorem 16.8. *Given a convex set K with diameter D and any convex function $f : \mathbb{R}^n \to \mathbb{R}$ with $\forall x \in \mathbb{R}^n \ \|\nabla f(x)\|_2 \leq G$ and suppose $x_0 \in K$ then $f(\hat{x}) - f(x^*) \leq \varepsilon$ if we set $T = \left(\frac{GD}{\varepsilon}\right)^2$ and $\eta_t = \frac{\varepsilon}{G^2}$.*

Proof. The argument is essentially the same as that for Theorem 16.7. The only hiccup is that now $-\eta\nabla f(x_t) = x'_{t+1} - x^*$, not $x_{t+1} - x^*$. But this is okay: if we could replace x_{t+1} with x'_{t+1} in (16.6), we would be all set. This boils down to showing

$$\left\|x'_{t+1} - x^*\right\| \geq \|x_{t+1} - x^*\|.$$

But this is easy, because $x_{t+1} = \Pi_K(x'_{t+1})$ and $x^* \in K$. Because K is convex, projecting to it gets us closer to *every point* in K, in particular to x^*. This is because the angle $x^* \to \Pi_K(x'_{t+1}) \to x'_{t+1}$ cannot be acute: if it were acute, we could show that K wasn't actually convex. □

3 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, you propose an $x_t \in K$ and an adversary exhibits 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

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

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

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

Looking back at the proof in Section 2, the inequality (16.3) immediately extends to give us

$$f_t(x_t) - f_t(x^*) + \Phi_{t+1} - \Phi_t \leq B.$$

Now summing this over all times t gives

$$\sum_{t=0}^{T-1} \left(f_t(x_t) - f_t(x^*)\right) \leq \Phi_0 + BT = \Phi_0 + \eta T G^2/2,$$

and using $T \geq \left(\frac{DG}{\varepsilon}\right)^2$ and $\eta = \frac{\varepsilon}{G^2}$ as above, this implies

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

One advantage of this algorithm (and analysis) is that it holds for all convex bodies K and all convex functions, as opposed to the MW algorithm which, as stated, works just for the unit simplex and linear losses. Of course it now depends on D (which, in the worst case is the diameter of K), and G (which is related to the class of functions). If we consider these quantities G, D as constants, the $(\frac{1}{\varepsilon^2})$ dependence is the same.

In many cases we do care about the fine-grained dependence on K and functions, so let's compare the two for the unit simplex and linear lossses (i.e., functions $f_t(x) = \langle \ell_t, x\rangle$ with $\|\ell\|_\infty = 1$). The regret bound above give us $T = \frac{2N}{\varepsilon^2}$ because $D \leq \text{diam}(K) = \sqrt{2}$ and $\|\nabla l_i\|_2 \leq \sqrt{N}$. This is much worse compared to $T = \frac{\log N}{\varepsilon^2}$, which is the guarantee that multiplicative weights provides.

The problem, at a high level, is that we are "choosing the wrong norm": we are working in ℓ_2 instead of ℓ_1. In the next lecture we will see what this means, and how this dependence on N be improved via the Mirror Descent framework.

3.1 Subgradients

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

Definition 16.9 (Subgradient)**.** A vector z_x is called a *subgradient* at point x if

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

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.

4 Stronger Assumptions

If the function f is better-behaved, then 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 parameters G, D. There are two standard assumptions one can 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 guarantees.

4.1 α-strongly convex functions

Definition 16.10 (Strong Convexity)**.** A function is *α-strongly convex* if

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2} \|x - y\|^2 \text{ for all } x, y \in K \tag{16.7}$$

Fact 16.11. *A twice-differentiable convex function f is α-strongly convex if and only if $\nabla^2 f \succeq \alpha I$.*

In this case, the gradient descent algorithm with step size $\eta_t = O\left(\frac{1}{\alpha t}\right)$ converges to a solution with error ε in $T = O(\frac{G^2}{\alpha \varepsilon})$ Put proof!

4.2 β-smooth function

Definition 16.12. A function f is a β-smooth convex function if

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2} \|x - y\|^2 \text{ for all } x, y \in K \tag{16.8}$$

Fact 16.13. *A twice-differentiable convex function f is β-smooth if and only if $\Leftrightarrow \nabla^2 f \preceq \beta I$.*

In this case, the gradient descent algorithm with step size $\eta_t = O\left(\frac{1}{\beta}\right)$ yields an $\hat{x}$ which satisfies $f(\hat{x}) - f(x^*) \leq \varepsilon$ when $T = O(\frac{D\beta}{\varepsilon})$.

In the smoothness definition (16.8), plug in $x = x_t$ and $y = x_{t+1} = x_t - \eta \nabla f(x_t)$ 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 RHS is minimized by setting $\eta = \frac{1}{\beta}$, when we get

$$f(x_{t+1}) - f(x_t) \leq -\frac{1}{2\beta} \|\nabla f(x_t)\|^2. \tag{16.9}$$

So the improvement is large when the gradients are large too. Finish proof!

4.3 Well-conditioned Functions

Functions that are both β-smooth and α-strongly convex are known as "well-conditioned" functions, with *condition number* $\kappa = \beta/\alpha$. From the facts above, the condition number is the ratio of the largest to the smallest eigenvalue of the Hessian $\nabla^2 f$. In this case, we get a much stronger convergence—ε-closeness in time $T = O(\log \frac{1}{\varepsilon})$.[3]

Theorem 16.14. *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*

$$f(x_t) - f(x^*) \leq \frac{\beta}{2} \exp\left(\frac{-t}{\kappa}\right) \|x_0 - x^*\|^2 .$$

Proof. For α-strongly-convex f, we can use its definition (16.7) to get:

$$\begin{aligned} f(x_t) - f(x^*) &\leq \langle \nabla f(x_t), x_t - x^* \rangle - \frac{\alpha}{2} \|x_t - x^*\|^2 \\ &\leq \|\nabla f(x_t)\| \, \|x_t - x^*\| - \frac{\alpha}{2} \|x_t - x^*\|^2 \\ &\leq \frac{1}{2\alpha} \|\nabla f(x_t)\|^2 \end{aligned} \tag{16.10}$$

where we use that the RHS is maximized when $\|x_t - x^*\| = \|\nabla f(x_t)\| / \alpha$. Now combining with (16.9) we have that

$$f(x_{t+1}) - f(x_t) \leq -\frac{\alpha}{\beta}\Big(f(x_t) - f(x^*)\Big), \tag{16.11}$$

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

$$\Delta_{t+1} \leq (1 - \frac{\alpha}{\beta})\Delta_t \leq (1 - 1/\kappa)^t \Delta_0 \leq \exp(-t/\kappa) \cdot \Delta_0.$$

We can control the value of Δ_0 by using (16.8) in $x = x^*, y = x_0$; since $\nabla f(x^*) = 0$, get $\Delta_0 = f(x_0) - f(x^*) \leq \frac{\beta}{2} \|x_0 - x^*\|^2 \leq \frac{\beta D^2}{2}$. □

Well-conditioned (and strongly-convex) functions have the good property that closeness in function value implies pointwise closeness: intuitively, since the function is curving at least quadratically, the function values at points far from the minimizer must be significant. Formally, use (16.7) with $x = x^*, y = x_t$ and the fact that $\nabla f(x^*) = 0$ to get

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

References

[Com06] Wikimedia Commons. Gradient ascent(countour), 2006. Available at http://en.wikipedia.org/wiki/Gradientdescent#/media/File:Gradientascent(contour).png. 16.2

[VT] Nisheeth Vishnoi and Jakub Tarnawski. Fundamentals of convex optimization. lecture 1 basics, gradient descent and its variants. Available at http://tcs.epfl.ch/files/content/sites/tcs/files/Lec1-Fall14-Ver1.pdf. 16.1

[3] In the numerical analysis literature this is called *linear convergence*, since the number of iterations increases linearly in the number of bits of accuracy—i.e., to decrease ε by a factor of 2 we need to run only for a "constant" number of extra iterations.