Lecture 12: Convexity and gradient descent

Jerry Li

November 11, 2026

In this class, we will start finally developing algorithms for some of the problems we discussed in previous classes, and we'll see how the core concepts underlying these algorithms will generalize to obtain very useful algorithms under very general assumptions.

As an example, recall the setting of linear regression we considered earlier in the class. Here, our data consist of feature vectors $x_1, \dots, x_n \in \mathbb{R}^d$ and responses $y_1, \dots, y_n \in \mathbb{R}$. Let

$$X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{pmatrix}, \qquad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} .$$

The least-squares problem is to find a parameter vector $\theta \in \mathbb{R}^d$ minimizing the ordinary least-squares (OLS) objective

$$L(\theta) = \|X\theta - y\|_2^2 .$$

In a previous lecture, we claimed that this loss could be optimized efficiently. The goal of this class (and of several of the next lectures) will be to understand what the underlying structure is that enables fast optimization of such problems. Indeed, recall we also considered 3 regularized versions of this task, with varying levels of complexity. For $\lambda > 0$, we considered:

$$\begin{aligned} \text{Ridge regression} \qquad & \min_\theta \ \|X\theta - y\|_2^2 + \lambda \|\theta\|_2^2 , \\ \text{LASSO} \qquad & \min_\theta \ \|X\theta - y\|_2^2 + \lambda \|\theta\|_1 , \\ \ell_0 \text{ penalization} \qquad & \min_\theta \ \|X\theta - y\|_2^2 + \lambda \|\theta\|_0 . \end{aligned}$$

Here $\|\theta\|_0$ counts the number of nonzero coordinates. It turns out that the complexity of optimizing these objectives is (roughly speaking)

$$\text{LASSO} \quad \leq \quad \text{OLS} \quad \leq \quad \text{LASSO} \quad \ll \quad \ell_0 \text{ penalization} ,$$

and indeed, while the first three can be solved in polynomial time, the last one likely cannot.

It turns out that there is a nice, generic reason explaining the complexity of these tasks, related to notions of *convexity*.

1 Convex sets and convex functions

There are related notions of convexity for sets and functions. First, let's define a notion of a convex set:

Definition 1.1. A set $S \subseteq \mathbb{R}^d$ is *convex* if for every $x, y \in S$ and every $\lambda \in [0, 1]$,

$$\lambda x + (1 - \lambda) y \in S .$$

In other words, S contains the whole line segment joining any two of its points, as in Figure 1.

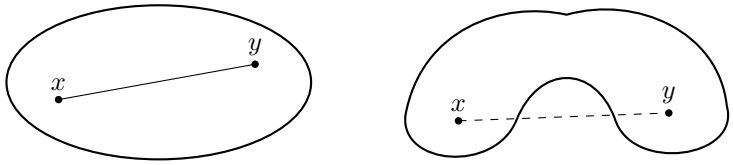


Figure 1: In the convex set, the entire segment between x and y stays inside. In the nonconvex set, the dashed segment passes through the indentation.

Example 1.1. The Euclidean unit ball $\{x \in \mathbb{R}^d : \|x\|_2 \leq 1\}$ is convex. More generally, the unit ball of any norm is convex, including the ℓ_p norms for $p \geq 1$. Indeed, if $\|x\|, \|y\| \leq 1$, then the triangle inequality gives

$$\|\lambda x + (1-\lambda)y\| \leq \lambda \|x\| + (1-\lambda)\|y\| \leq 1 .$$

On the other hand, $\{x \in \mathbb{R}^d : \|x\|_0 \leq k\}$ fails to be convex when $1 \leq k < d$. For example, when $k = 1$, two distinct coordinate vectors lie in the set, but their midpoint has two nonzero coordinates. For general k, take two vectors with $k-1$ shared coordinates equal to 1 and one additional coordinate equal to 1 in a different position in each vector. Their midpoint has $k+1$ nonzero coordinates.

Convex functions Here, and for the rest of the class, for simplicity, let's assume that the functions we care about are differentiable (or at least, differentiable almost everywhere). For such functions, we have the following definition:

Definition 1.2. A function $f : \mathbb{R}^d \to \mathbb{R}$ is *convex* if for all $x, h \in \mathbb{R}^d$, we have that

$$f(x+h) \geq f(x) + \langle \nabla f(x), h \rangle . \tag{1}$$

In one dimension, this just means that the function lies above every tangent line:

$$f(x+h) \geq f(x) + f'(x)h .$$

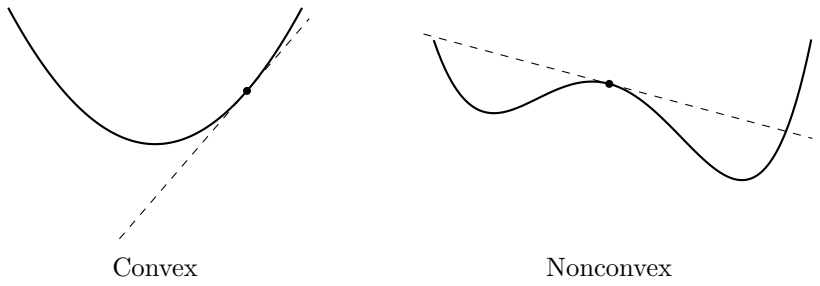


Figure 2: The convex function stays above its tangent line. The nonconvex function falls below the dashed tangent in the right panel.

There are many equivalent definitions of convex functions, and depending on your application, it might be more useful to consider one or the other. We list several below:

Theorem 1.1. *Let f be a differentiable function. Then, the following are equivalent:*

- *f is convex.*

- *For every $x, y \in \mathbb{R}^d$ and every $\lambda \in [0,1]$,*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \; . \tag{2}$$

  That is, the chord between $(x, f(x))$ and $(y, f(y))$ gives an upper bound between those points, see Figure 3.

- *f satisfies* Jensen's inequality*: for any random vector $Z \in \mathbb{R}^d$,*

$$f(\mathbb{E}[Z]) \leq \mathbb{E}[f(Z)] \; .$$

- *The* sublevel *sets of f are convex. For every a, define $S_a = \{x \in \mathbb{R}^d : f(x) \leq a\}$. Then S_a is convex, for all a.*

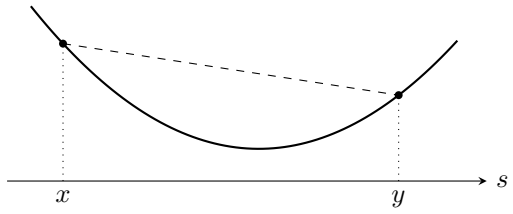


Figure 3: Between x and y, the graph of a convex function lies below the chord joining $(x, f(x))$ and $(y, f(y))$.

Remark 1.1. We will not worry about this too much for this class, but the most general definition of convexity is the chord-based definition, which applies even if f is not differentiable. To deal with non-differentiable functions both here and in gradient descent, there is a well-defined notion of a *sub-gradient* which allows us to extend the theory we will discuss here to general (non-differentiable) convex functions.

Why convexity helps. The key property of convexity is that for a convex function, every local minimum is a global minimum. Indeed, suppose we are at a local minima x, or equivalently, a point x so that $\nabla f(x) = 0$. Then, the tangent condition gives that

$$f(x+h) \geq f(x) + \langle \nabla f(x), h \rangle = f(x) \qquad \text{for every } h \in \mathbb{R}^d \; .$$

For completeness, observe that the statement also holds without differentiability. If there were a point with smaller value, then the chord condition would give points of smaller value arbitrarily close to x along the segment toward that point.

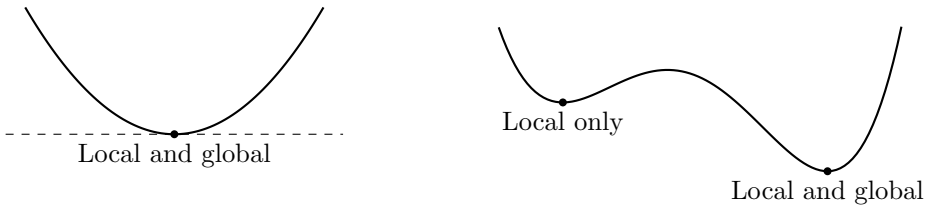


Figure 4: A local minimum of a convex function (left) is also global. The nonconvex function on the right has a shallower local minimum and a deeper global minimum.

2 Gradient descent

This structure implies that all we need to do is find a local minima, that is, a point where the derivative is zero. In one dimension, if we are on the right of a minimizer and the derivative is positive, a small step to the left decreases f. If the derivative is negative, we instead step to the right. This suggests starting at an initial point x_0 and repeatedly moving in the direction of the negative derivative.

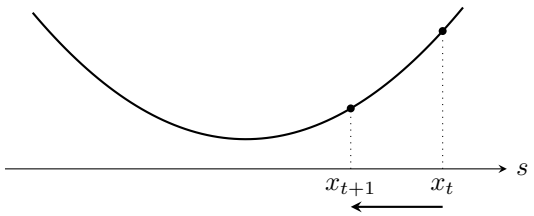


Figure 5: Here $f'(x_t) > 0$, so a sufficiently small step in the negative derivative direction moves left and decreases f.

In one dimension this yields the update rule $x_{t+1} = x_t - \eta_t f'(x_t)$, for some *step-size schedule* η_t. In higher dimensions the same idea gives

$$x_{t+1} = x_t - \eta_t \nabla f(x_t) \; . \tag{3}$$

The remaining question is: how do we choose good step-size schedule to guarantee that this sequence of updates will converge?

What does it mean to converge? Actually, before that, we need to answer a quick question: what does it even mean to converge? There are two natural objectives:

1. Choose a minimizer $x^\star$ and ask that the distance $\|x_T - x^\star\|_2$ be small.
2. Ask for a small objective gap, $f(x_T) - \min_x f(x) \leq \varepsilon$. This is often sufficient in practice, even when we cannot single out one minimizer.

In some sense, the second guarantee is usually weaker than the former. However, for some functions, it's all we can really hope for: this is because for some convex functions, the minimizer is not uniquely defined, and so it's hopeless to try to find one. For instance, $f(x) = (\max\{0, -x\})^2$ has a whole ray of minimizers. More generally, one can find examples where even though the minimizer is uniquely defined, it's more or less unrealistic to ever hope to actually recover it (why?).

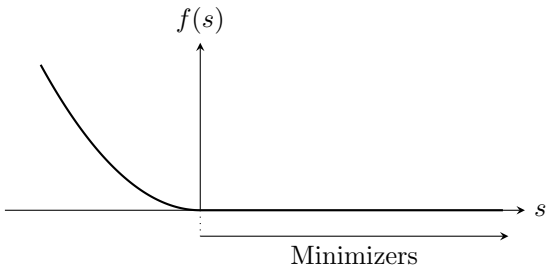
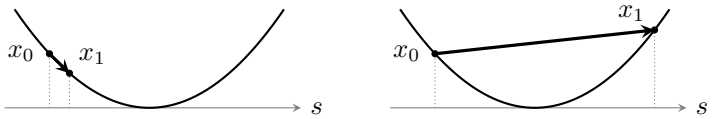


Figure 6: The convex function $f(s) = (\max\{0, -s\})^2$ is minimized at every $s \geq 0$.

3 Smoothness and strong convexity

There are two counterbalancing effects we need to consider when choosing η_t. If our η_t is too small, then our step will make little progress. On the other hand, if η_t is too large, we can overshoot the minimizer and even increase the objective; see Figure 7. While assumptions of this sort are not strictly speaking necessary, it turns out that there are two additional properties of convex functions will help us understand this balance, namely, *smoothness* and *strong convexity*.



Small step Overshoot

Figure 7: One step on $f(s) = s^2/2$ from $x_0 = -1$. With $\eta = 1/5$, the iterate moves only to $-4/5$. With $\eta = 11/5$, it crosses the minimum and lands at $6/5$, where the objective is higher.

Managing overshooting: smoothness At its heart, the worry about overshooting is that gradient descent is performing a *local* update

Definition 3.1. A differentiable function $f\colon \mathbb{R}^d \to \mathbb{R}$ is *L-smooth* if for all $x, y \in \mathbb{R}^d$,

$$\|\nabla f(x) - \nabla f(y)\|_2 \le L \|x - y\|_2 \ .$$

Intuitively, smoothness gives a quadratic upper bound on how far the function can rise above its tangent approximation, as illustrated in Figure 8.

Lemma 3.1. *If f is L-smooth, then for all $x, y \in \mathbb{R}^d$,*

$$f(y) \le f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|_2^2 \ . \tag{4}$$

Proof. Let $h = y - x$. By the fundamental theorem of calculus along the segment from x to y,

$$f(y) - f(x) - \langle \nabla f(x), h \rangle = \int_0^1 \langle \nabla f(x + sh) - \nabla f(x), h \rangle \, ds \ .$$

Cauchy–Schwarz and L-smoothness bound the integrand above by $Ls \|h\|_2^2$. Integrating over s gives (4). □

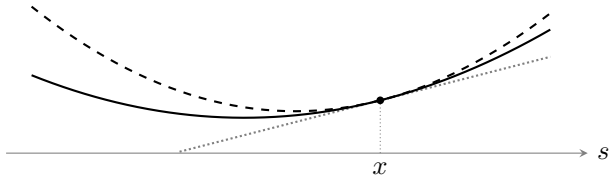


Figure 8: An L-smooth function (solid) lies below the quadratic model (dashed) formed by adding $\frac{L}{2}(s-x)^2$ to the tangent at x (dotted).

Managing small steps: strong convexity Just as smoothness yields an upper bound on how quickly the gradient can change, it will be useful as well to have a lower bound on how quickly it can change. Intuitively, such a bound allows us to guarantee that the gradient will not be close to zero unless our current iterate is actually close to the optimal point. Formally, this is expressed via the notion of *strong convexity*:

Definition 3.2. A differentiable function $f\colon \mathbb{R}^d \to \mathbb{R}$ is *ℓ-strongly convex* for $\ell > 0$ if for all $x, h \in \mathbb{R}^d$,

$$f(x+h) \geq f(x) + \langle \nabla f(x), h \rangle + \frac{\ell}{2} \|h\|_2^2 \ .$$

Equivalently, substituting $h = y - x$, we have for all $x, y \in \mathbb{R}^d$,

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\ell}{2} \|y - x\|_2^2 \ . \tag{5}$$

Strong convexity puts a quadratic lower bound beneath the graph, which meets at x (Figure 9). In particular, at a minimizer $x^\star$ the gradient is zero, so $f(y) - f(x^\star) \geq (\ell/2)\|y - x^\star\|_2^2$. Thus a minimizer, if it exists, is unique.

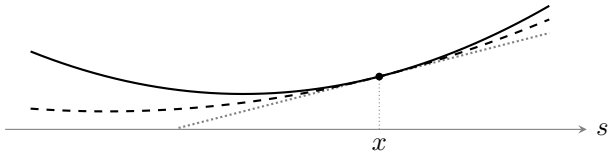


Figure 9: An ℓ-strongly convex function (solid) lies above the quadratic model (dashed) formed by adding $\frac{\ell}{2}(s-x)^2$ to the tangent at x (dotted).

The Hessian interpretation. If f is convex and twice continuously differentiable, its Hessian has nonnegative eigenvalues. Then, it turns out that f is L-smooth exactly when every Hessian eigenvalue is at most L at every point, and f is ℓ-strongly convex exactly when every Hessian eigenvalue is at least ℓ at every point.

4 Convergence of gradient descent

We now have enough to prove that the iterates of gradient descent approach the minimizer. In particular, we will show that for smooth and strongly convex functions, it suffices to take constant stepsize.

Theorem 4.1. *Let $f\colon \mathbb{R}^d \to \mathbb{R}$ be L-smooth and ℓ-strongly convex, where $0 < \ell \leq L$, and let $x^\star$ be its minimizer. Run gradient descent with $\eta_t = \eta = \ell/(2L^2)$. Then for every integer $T \geq 0$,*

$$\|x_T - x^\star\|_2^2 \leq \left(1 - \frac{\ell^2}{4L^2}\right)^T \|x_0 - x^\star\|_2^2 \ . \tag{6}$$

Proof. Write $g_t = \nabla f(x_t)$. Expanding the square in (3), we get

$$\|x_{t+1} - x^\star\|_2^2 = \|x_t - x^\star\|_2^2 \underbrace{- 2\eta \langle g_t, x_t - x^\star \rangle}_{\text{progress term}} + \underbrace{\eta^2 \|g_t\|_2^2}_{\text{quadratic term}} \ . \tag{7}$$

For the progress term, apply strong convexity with $x = x_t$ and $y = x^\star$ and rearrange. This gives

$$\langle g_t, x_t - x^\star \rangle \geq f(x_t) - f(x^\star) + \frac{\ell}{2}\|x_t - x^\star\|_2^2 \geq \frac{\ell}{2}\|x_t - x^\star\|_2^2 \ .$$

For the quadratic term, apply smoothness at the same pair of points and use $\nabla f(x^\star) = 0$ to get

$$\|g_t\|_2^2 = \|\nabla f(x_t) - \nabla f(x^\star)\|_2^2 \le L^2 \|x_t - x^\star\|_2^2 \ .$$

Substituting both inequalities into (7), we get

$$\|x_{t+1} - x^\star\|_2^2 \le (1 - \eta\ell + \eta^2 L^2) \|x_t - x^\star\|_2^2 = \left(1 - \frac{\ell^2}{4L^2}\right) \|x_t - x^\star\|_2^2 \ .$$

Iterating this inequality T times gives (6). □

Remark 4.1. In the numerical optimization literature, this sort of convergence (where the rate of convergence is geometrically fast to the true point), is somewhat confusingly referred to as *linear convergence*.

In fact, this bound is loose, and it turns out that with step size $\eta = 1/L$, we can sharpen the bound to

$$\|x_T - x^\star\|_2^2 \le \left(1 - \frac{\ell}{L}\right)^T \|x_0 - x^\star\|_2^2 \qquad \text{for } T \ge 1.$$

5 Step sizes and convergence rates

This is only one of many settings considered in the literature for the convergence of gradient descent, and different assumptions on f lead to different convergence rates. The table below summarizes typical guarantees after T updates. For the nonsmooth cases, we use *subgradient* steps, which replace the gradient by a subgradient g satisfying

$$f(y) \ge f(x) + \langle g, y - x \rangle \qquad \text{for all } y.$$

Let $R = \|x_0 - x^\star\|_2$ and assume the subgradients used by the method have norm at most G. Let $\overline{x}$ denote an appropriate average of the iterates, as specified:

	Nonsmooth	**L-smooth**
Convex	$\eta = R/(G\sqrt{T})$, uniform average. $f(\overline{x}) - f(x^\star) = O(GR/\sqrt{T})$.	$\eta = 1/L$, final iterate. $f(x_T) - f(x^\star) = O(LR^2/T)$.
ℓ-strongly convex	$\eta_t = 2/(\ell(t+2))$, average with weights $t+1$ for $t = 0, \dots, T-1$. $f(\overline{x}) - f(x^\star) = O(G^2/(\ell T))$.	$\eta = 1/L$, final iterate. $\|x_T - x^\star\|_2^2 \le (1 - \ell/L)^T R^2$.

Step sizes in practice. The theoretical step sizes depend on quantities like L that can be hard to estimate. In practice, step-size scheduling is often closer to an art than a science, and we often rely on expensive hyperparameter sweeps to try to find better schedules. Below, we list three common approaches, but this is by no means an extensive list!

- *Three-stage schedule.* Divide the run into three stages. Within each stage, use a constant step size within each stage and reduce it at each transition.
- *Cosine annealing.* Given an initial step size η_0 and a horizon T, use the schedule

  $$\eta_t = \frac{\eta_0}{2}\left(1 + \cos\left(\frac{\pi t}{T}\right)\right) \qquad \text{for } 0 \le t \le T.$$

  The schedule starts at η_0 and tapers to zero at T (Figure 10).
- *Sweep over candidates.* Try several step sizes with the same iteration budget and compare how well they perform (Figure 11). In a learning problem, we can choose the candidate with the lowest validation loss.

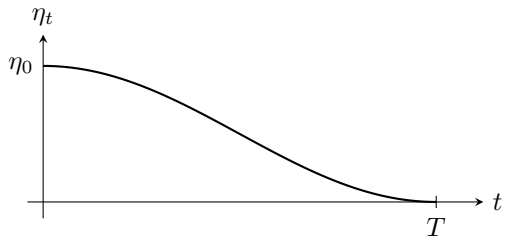


Figure 10: Cosine annealing decreases the step size from η_0 to zero over T iterations.

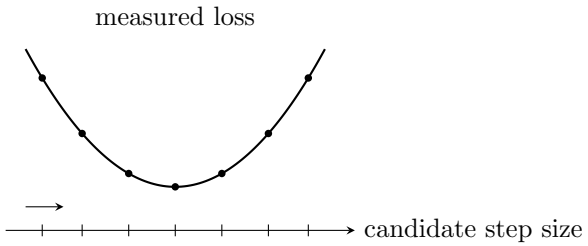


Figure 11: A step size sweep compares measured losses across candidate step sizes.