1 Introduction

This course provides an introduction to the theory and practice of continuous optimization, with a focus on first-order (gradient-based) methods. The aim of the course is not to provide an extremely rigorous treatment of convex optimization, but instead introduce you to these concepts so that you can get a sense of how to apply it to problems that you are interested in. For the interested reader, there are a lot of excellent books and lecture notes.

The course is roughly split into two parts. Both parts will have a mix of both theory and programming.

1. In Part 1, we will cover the standard theoretical analyses of gradient-based methods for different problem settings, and implement the algorithms on real problems, mostly motivated by applications in machine learning.
2. In Part 2, we will focus on "advanced topics" which build on the material covered in Part 1. Topics include stochastic optimization, acceleration, and derivative-free optimization.

1.1 Why study optimization?

In the most general form, we consider solving the optimization problem:

$$\begin{aligned} \text{minimize}_x \quad & f(x) \\ \text{subject to} \quad & x \in S \subseteq \mathbb{R}^d. \end{aligned}$$

- Here, x denotes the *optimization variable*.
- We also have a constraint $x \in S$, and we call S the *constraint set*.
- Lastly, we call $f(\cdot)$ the *objective function*.

Different fields might use different names for the variables. For example in machine learning it is common to minimize a function L over a variable w or θ.

Many real-world problems in modern engineering, finance, and data science can be framed as optimization problems. Thus, it is useful to develop a common framework for studying such problems. Let's consider some examples.

- Portfolio optimization: Suppose we have some money, and we would like to invest it into d different stocks. The variable x_i for $i \in [d]$ represents the investment into the ith stock, so $x \in \mathbb{R}^d$ represents the total portfolio allocation. However, we might have some constraints on our portfolio allocation. For example, we do not have infinite money, which corresponds to some bound on x, for example $\|x\|_1 \leq C$. We can also have a constraint that the investments are diversified, and this can be captured by assuming that $x_i > c > 0$. The objective function f can capture some desired cost or utility. It can be the total return for the portfolio, or it can be some measure of "risk" which represents how much downside there is to the investment strategy, or maybe some weighted combination.
- Device design: Imagine we have a bunch of devices we want to size in an electronic circuit. The optimization variable is the length and width of each chip, and f can be some measure of power consumption. However, we often have constraints on the size of the circuit components as determined by the manufacturer.

- Operations research: Suppose we are a major shipping company like UPS. Every day we have to determine how to move packages around the country. Suppose we have a fleet of trucks that we want to send from City A to City B, with several intermediate cities along the way. We can decide how many trucks we want to send along each route. But there are constraints: we cannot send more trucks on a route than what the capacity of the road allows (otherwise they end up in a traffic jam). What is the best way to allocate our fleet of vehicles? It turns out that this is a very old problem.

1.2 Our First Algorithm: Grid Search

Let's consider a simple example. Suppose $f : \mathbb{R} \to \mathbb{R}$ is a continuous function in 1D, and suppose $S = [0, 1]$. Our goal is to find the global minimizer $x^\star$. Of course, since $x^\star$ is a real number, we might not be able to exactly find $x^\star$. However, we will usually be content with finding a near-optimal point $\widehat{x}$: that is, $\widehat{x}$ satisfies the property that $f(\widehat{x}) \leq f(x^\star) + \varepsilon$ for some very small value of $\varepsilon > 0$.

We will make the following assumption on f:

Assumption 1. *The function f is L-Lipschitz, i.e., for any $x, y \in [0, 1]$ we have $|f(x) - f(y)| \leq L|x - y|$.*

Roughly speaking Assumption 2 states that small changes in the input only change the function very slightly, up to a factor of L. This means that the slope cannot be too large for differentiable f. Also functions which are nondifferentiable can satisfy the Lipschitz assumption, for example consider $f(x) = |x - x^\star|$.

There is a very simple algorithm for optimization in 1D described in Algorithm 1.

Algorithm 1 Grid Search

1: **Require:** accuracy parameter ε, function f.
2: Construct the partition $S' = \{0, \varepsilon/L, 2\varepsilon/L, \cdots, 1\}$
3: For each $x' \in S'$, evaluate $f(x')$.
4: **Return** $\widehat{x} := \operatorname{argmin}_{x' \in S'} f(x')$.

Theorem 1. *Under Assumption 2, Algorithm 1 uses $O(L/\varepsilon)$ function evaluations and returns an ε-optimal point.*

Proof of Theorem 1. Let $\widetilde{x}^\star \in S'$ denote the point that $x^\star$ is closest to in S'. By Assumption 2, we know that

$$|f(\widetilde{x}^\star) - f(x^\star)| \leq L \cdot |\widetilde{x}^\star - x^\star| \leq L \cdot \frac{\varepsilon}{L} = \varepsilon.$$

Therefore we know that $f(\widetilde{x}^\star) \leq f(x^\star) + L\varepsilon$. Furthermore, by optimality of $\widehat{x}$ we have

$$f(\widehat{x}) \leq f(\widetilde{x}^\star) \leq f(x^\star) + \varepsilon,$$

completing the proof of Theorem 1. □

It's also clear that if we don't know the analytic form of f then we need some kind of assumption like Assumption 2. Consider the "needle-in-the-haystack" function. $f(x) = 1 - \mathbb{1}\{x = x^\star\}$. If we are just blinding evaluating points, there is no way we can actually find the minimum.

Grid Search in Higher Dimensions. It is straightforward to extend grid search to work in higher dimensions. We have an analogue of Assumption 2 for multivariate functions.

Assumption 2. *The function f is L-Lipschitz, i.e., for any $x, y \in [0,1]$ we have $|f(x) - f(y)| \leq L\|x - y\|_2$.*

Let's say $S = [0,1]^d$. We can define the box partition:

$$S' \coloneqq \{0, \varepsilon/(L\sqrt{d}), 2\varepsilon/(L\sqrt{d}), \cdots, 1\}^d$$

Following the same analysis we know that the point closest to the optimum $\widetilde{x}^\star$ satisfies

$$f(\widetilde{x}^\star) \leq f(x^\star) + L \cdot \|\widetilde{x}^\star - x^\star\|_2 \leq L\sqrt{d} \cdot \|\widetilde{x}^\star - x^\star\|_\infty \leq \varepsilon.$$

The second inequality uses the fact that for any d-dimensional vector v, we have $\|v\|_2 \leq \sqrt{d}\|v\|_\infty$. So therefore we have that $f(\widehat{x}) \leq f(\widetilde{x}^\star)$, so $\widehat{x}$ is ε-suboptimal.

Why is grid search bad? Observe that in higher dimensions, the grid search algorithm requires us to compute function evaluations at $\left(\frac{L\sqrt{d}}{\varepsilon}\right)^d$ points. Unfortunately, for many optimization problems such as the ones we have already mentioned before, d can be quite large! An exponential dependence on d is undesirable. In the worst case this dependence cannot be removed. However, the rest of this course will be concerned with further restrictions on the objective function f and the constraint set S that enable us to reduce such dependence on d.

Gradient Methods. Instead of grid search, we will mostly study gradient methods for optimization. Gradient descent is a popular technique for minimizing functions. At a high level, it is a form of *local search*, where we maintain some "guess" of the best point $x^\star$, and iteratively refine our guess. To see how to do this, consider minimizing $f : \mathbb{R} \to \mathbb{R}$. By Taylor's theorem, we know that for any δ sufficiently small, we have

$$f(x - \delta) \approx f(x) - \delta f'(x) + \frac{1}{2}\delta^2 f''(x).$$

If δ is very small, then the term $\frac{1}{2}\delta^2 f''(x)$ is negligible. This tells us that if we start at the point x and move to the nearby point $x - \delta$, then we would expect to decrease our function value by the quantity $\delta \cdot f'(x)$. In this course, we will generalize this idea to multivariate functions. In fact, we will show that algorithms based on this idea allow us to solve the optimization problem.

2 Convexity

In this section, we will introduce the concept of convexity (both for functions and sets).

2.1 Convex Sets

Definition 1. *A set $K \subseteq \mathbb{R}^d$ is* convex *if the line segment between any two points in K is also contained in K. Formally for any $x, y \in K$ and scalars $\gamma \in [0,1]$ we have $\gamma x + (1-\gamma)y \in K$.*

Some examples.

1. Linear spaces $\{x \in \mathbb{R}^d : Ax = 0\}$ and halfspaces $\{x \in \mathbb{R}^d : \langle a, x\rangle \geq 0\}$.

2. Affine transformation of convex sets. If K is convex, then so is $\{Ax + b : x \in K\}$.
3. Norm balls $\{x \in \mathbb{R}^d : \|x\|_p \leq B\}$ for any $p \geq 1$.
4. Intersections of convex sets.
5. Positive Semidefinite Matrices: $S_+^d = \{A \in \mathbb{R}^{d \times d} : A \succeq 0\}$. By $A \succeq 0$ we mean that $x^\top Ax \geq 0$ for all $x \in \mathbb{R}^d$. To see this let A be the set of all symmetric matrices.
6. Polyhedra: $\{x \in \mathbb{R}^d : Ax = b, Cx \leq d\}$.

2.2 Convex Functions

Definition 2. *A function $f : K \to \mathbb{R}$ is* convex *if for any $x, y \in K$ and scalars $\gamma \in [0, 1]$, we have*

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

There is a relationship between convex functions and sets.

Definition 3. *The* epigraph *of a function $f : K \to \mathbb{R}$ is defined as*

$$\text{epi}(f) = \{(x, t) : f(x) \leq t\}.$$

Proposition 1. *A function is convex if and only if its epigraph is convex.*

Some examples.

- Affine functions $f(x) = ax + b$
- Exponential: $f(x) = e^{ax}$.
- Powers: $f(x) = |x|^p$ for any $p \geq 1$.
- Negative log $f(x) = -\log x$ and negative entropy $f(x) = -x \log x$.
- Norms: $\|x\|_p$ for any $p \geq 1$.
- max: $f(x) = \max\{x_1, \cdots, x_d\}$.
- Quadratic functions: $f(x) = \langle x, Ax \rangle + \langle b, x \rangle + c$ for any $A \succeq 0$.

Convex functions have the following nice property.

Proposition 2. *For any convex f, all local minima are also global minima.*

Proof. Let $x \in K$ be any local minima of f. We know that every point close to f must have larger function value. Now pick any $y \in K$. We can always pick γ small enough so that

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

The first inequality follows by the local minima property, and the second inequality follows by convexity. Therefore, rearranging this equation we get that $f(x) \leq f(y)$ for all $y \in K$, meaning that x is actually a global minima. □

2.3 First Order Characterization

Next we examine the first order condition that you might have seen in calculus class for univariate functions. For multivariate $f : K \to \mathbb{R}$ we can define the gradient at point $x \in K$ as the vector of partial derivatives:

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_d} \right) \in \mathbb{R}^d.$$

At this point it might be good to review some basic material on matrix calculus.

- Gradient of a linear function. $f(x) = \langle b, x \rangle$
- Gradient of a quadratic function. $f(x) = \langle x, Ax \rangle$.
- Gradient of norms.

We can relate linear functions of the gradients to 1D derivatives. For any $f : K \to \mathbb{R}$ that is differentiable, and any $x, y \in K$ we have

$$\langle \nabla f(x), y \rangle = \left. \frac{\partial f(x + \eta y)}{\partial \eta} \right|_{\eta = 0}.$$

For convex functions we have the following fact.

Proposition 3. *Assume $f : K \to \mathbb{R}$ is differentiable. Then f is convex if for all $x, y \in K$ we have*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle.$$

Proof. First let us assume that f is convex. Then we have for any $x, y \in K$

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

Rearranging this implies that

$$\begin{aligned} f(y) &\geq \frac{f(\gamma y + (1 - \gamma) x) - (1 - \gamma) f(x)}{\gamma} \\ &= f(x) + \frac{f(x + \gamma(y - x)) - f(x)}{\gamma} \\ &\to f(x) + \langle \nabla f(x), y - x \rangle \quad \text{as } \gamma \to 0. \end{aligned}$$

Now we show the other direction. Let's fix two points x, y as well as γ. Define $z = \gamma x + (1 - \gamma) y$ we can get that

$$f(x) \geq f(z) + \langle \nabla f(z), x - z \rangle, \quad \text{and} \quad f(y) \geq f(z) + \langle \nabla f(z), y - z \rangle.$$

If we add these two inequalities after scaling by γ and $1 - \gamma$ respectively we get that

$$\gamma f(x) + (1 - \gamma) f(y) \geq f(z) + \langle \nabla f(z), \gamma x + (1 - \gamma) y - z \rangle = f(\gamma x + (1 - \gamma) y).$$

This establishes convexity. □

A way to think about this is that for convex functions, the gradient of f at a given point x gives us a linear lower bound on the function at any other point. As a corollary, if K is taken to be the entire space $\mathbb{R}^d$ we know that x is a global minimum if and only if $\nabla f(x) = 0$.

3 Gradient Descent for Quadratics

Now we will introduce the gradient descent algorithm and analyze it for quadratic objective functions. Quadratic objective functions are already a very interesting class of functions to optimize, and they have many applications.

3.1 Gradient Descent Algorithm

Let us first think about how we want to build an algorithm to solve optimization. For now, we will consider unconstrained optimization. Imagine we have some function f which we want to minimize. The basic idea of an iterative algorithm is that we start with some *initial point* x_0 and we construct a sequence of points $x_1, x_2, \cdots$ which satisfies the property that

$$f(x_{t+1}) < f(x_t), \quad t = 0, 1, \cdots$$

Thus, eventually we will reach the minimizer $x^\star \in \operatorname{argmin}_x f(x)$. Concretely, to construct such a sequence of points, in every iteration, we will define the next iterate by searching along a direction, as $x_{t+1} = x_t + \eta_t d_t$, where d_t is a *descent direction* at x_t and η_t is a step size.

The method of **gradient descent** is an example of this general template. It can be traced back to Augustin Louis Cauchy in 1847. The idea is to pick the descent direction to be the negative of the gradient of f:

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

In other words, this is the direction of *steepest descent*, since we know that:

$$\operatorname*{argmin}_{d:\|d\|_2 \le 1} \left.\frac{\partial f(x+\eta d)}{\partial \eta}\right|_{\eta=0} = \operatorname*{argmin}_{d:\|d\|_2 \le 1} \langle \nabla f(x), d \rangle = -\frac{\nabla f(x)}{\|\nabla f(x)\|_2}.$$

The last equality is due to Cauchy-Schwarz inequality. In words, this says that the direction of descent which results in the greatest decrease in function value f is exactly $-\frac{\nabla f(x)}{\|\nabla f(x)\|_2}$.

Now that we have established the gradient descent methods, there are a few questions we would like to understand about it. The main question is one about iteration complexity: how many steps of gradient descent do we need to convergence to some fixed suboptimality? We did this kind of analysis already for the grid search algorithm.

3.2 Quadratic Minimization

Let's begin by optimizing quadratic objective functions:

$$\min_{x \in \mathbb{R}^d} f(x) \coloneqq \frac{1}{2}(x - x^\star)^\top Q (x - x^\star).$$

Here, we have $Q \succ 0$ is some $n \times n$ matrix. It is clear that the optimal value of this function is 0, and is obtained at the point $x = x^\star$.

We have already calculated the gradient of the function:

$$\nabla f(x) = Q(x - x^\star).$$

Convergence for constant stepsizes. We show this result for gradient descent with constant step size. We will use $\lambda_i(Q)$ to denote the ith largest eigenvalue of Q.

Theorem 2. *Suppose we pick $\eta_t = \eta = \frac{2}{\lambda_1(Q)+\lambda_n(Q)}$, then*

$$\|x_t - x^\star\|_2 \leq \left(\frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}\right)^t \|x_0 - x^\star\|_2.$$

Observe that $\lambda_1, \cdots \lambda_n$ are all positive and in strictly decreasing order, so the RHS is further upper bounded by $(\lambda_1/\lambda_n)^t \|x_0 - x^\star\|_2$. We call λ_1/λ_n the *condition number*, and it plays an important role in optimization! This convergence rate is extremely fast; and it is often called linear convergence or geometric convergence. The name "linear" comes from the fact that if you plot the error on a log-linear plot vs. iteration count, it lies below a line.

Let's prove this result.

Proof. By the GD update rule we have

$$x_{t+1} - x^\star = x_t - x^\star - \eta_t \nabla f(x_t) = (I - \eta_t Q)(x_t - x^\star).$$

Taking norms of both sides we get that

$$\|x_{t+1} - x^\star\|_2 \leq \|I - \eta_t Q\| \cdot \|x_t - x^\star\|_2.$$

Now observe that

$$\|I - \eta_t Q\| = \max(|1 - \eta_t \lambda_1(Q)|, |1 - \eta_t \lambda_n(Q)|).$$

Recall that under our choice of η_t we get that

$$\|I - \eta_t Q\| = 1 - \frac{2\lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)} = \frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}.$$

Plugging this back into our bound and applying recursion completes the proof. □

We also have the following corollary.

Corollary 1. *Under the choice $\eta_t = \eta = \frac{2}{\lambda_1(Q)+\lambda_n(Q)}$, gradient descent achieves the suboptimality guarantee*

$$f(x_t) - f(x^\star) \leq \frac{\lambda_1(Q)}{2}\left(\frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}\right)^{2t} \|x_0 - x^\star\|_2.$$

Proof. This follows by Cauchy Schwarz. □

Convergence for exact line search. A downside of the previous result is that we need to exactly know what $\lambda_1(Q)$ and $\lambda_n(Q)$ are for the matrix Q. In some cases, this might require some preliminary experimentation. Another strategy is the *exact line search rule*:

$$\eta_t = \operatorname*{argmin}_{\eta \geq 0} f(x_t - \eta \nabla f(x_t)).$$

That is, we will search in the direction $\nabla f(x_t)$ to find the step size that results in the largest function decrease. While exactly computing the step size is not possible, observe that this is actually a 1D optimization problem, so it can be reasonably solved with something like binary search!

Here is a convergence rate guarantee for exact line search.

Theorem 3. *Suppose we pick $\eta_t = \operatorname{argmin}_{\eta \geq 0} f(x_t - \eta \nabla f(x_t))$, then we have*

$$f(x_t) - f(x^\star) \leq \left(\frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}\right)^{2t} (f(x_0) - f(x^\star)).$$

This theorem has a very similar convergence rate to what we proved before.

Proof. First we show a closed form for the exact line search step size. Let $g_t \coloneqq Q(x_t - x^\star)$. Then we can verify that

$$\eta_t = \operatorname*{argmin}_{\eta \geq 0} \frac{1}{2}(x_t - x^\star - \eta g_t)^\top Q(x_t - x^\star - \eta g_t) = \frac{\langle g_t, g_t \rangle}{\langle g_t, Q g_t \rangle}.$$

To solve the minimization problem, we took derivatives of the objective and set it to zero. Therefore we have

$$\begin{aligned}
f(x_{t+1}) &= \frac{1}{2}(x_t - x^\star - \eta_t g_t)^\top Q(x_t - x^\star - \eta_t g_t) \\
&= \frac{1}{2}(x_t - x^\star)^\top Q(x_t - x^\star) - \eta_t \|g_t\|_2^2 + \frac{\eta_t^2}{2} g_t^\top Q g_t \\
&= \frac{1}{2}(x_t - x^\star)^\top Q(x_t - x^\star) - \frac{\|g_t\|_2^4}{2 g_t^\top Q g_t} \\
&= \left(1 - \frac{\|g_t\|_2^4}{g_t^\top Q g_t \cdot g_t^\top Q^{-1} g_t}\right) f(x_t).
\end{aligned}$$

The last line used the fact that $f(x_t) = \frac{1}{2}(x_t - x^\star)^\top Q(x_t - x^\star) = \frac{1}{2} g_t^\top Q^{-1} g_t$.

Now we use Kantorovich's inequality to get that

$$\frac{\|g_t\|_2^4}{g_t^\top Q g_t \cdot g_t^\top Q^{-1} g_t} \geq \frac{4\lambda_1(Q)\lambda_n(Q)}{(\lambda_1(Q) + \lambda_n(Q))^2}.$$

So therefore we have

$$f(x_{t+1}) \leq \left(1 - \frac{4\lambda_1(Q)\lambda_n(Q)}{(\lambda_1(Q) + \lambda_n(Q))^2}\right) f(x_t) = \left(\frac{\lambda_1(Q) - \lambda_n(Q)}{\lambda_1(Q) + \lambda_n(Q)}\right)^2 f(x_t).$$

Since $f(x^\star) = 0$ this concludes the proof. □

Proof of Kantorovich's inequality. Without loss of generality, we can assume that Q is a diagonal matrix with entries $\lambda_1, \cdots, \lambda_n$. This is because if Q can always be written as $V \Sigma V^\top$, so we can always reparameterize g_t as $V g_t'$ for some g_t'. Also suppose that $\|g_t\|_2 = 1$. Let α_i be the squared ith entry of g_t. The inequality can be rewritten as

$$\left(\sum_i \alpha_i \cdot \lambda_i\right)\left(\sum_i \frac{\alpha_i}{\lambda_i}\right) \leq \frac{(\lambda_1 + \lambda_n)^2}{4\lambda_1 \lambda_n}$$

Probabilistic Proof. We prove the following continuous statement: if X is a random variable in $[a, b]$ we have

$$\mathbb{E}[X]\,\mathbb{E}[X^{-1}] \leq \frac{(a+b)^2}{4ab}.$$

By Cauchy Schwarz we know that

$$\mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] \leq \big(\mathbb{E}\big[(X - \mathbb{E}[X])^2\big]\big)^{1/2}\big(\mathbb{E}\big[(Y - \mathbb{E}[Y])^2\big]\big)^{1/2}$$

If we take $Y = -X^{-1}$ we get

$$\mathbb{E}[X]\,\mathbb{E}\big[X^{-1}\big] \leq 1 + \big(\mathbb{E}\big[(X - \mathbb{E}[X])^2\big]\big)^{1/2}\big(\mathbb{E}\big[(X^{-1} - \mathbb{E}[X^{-1}])^2\big]\big)^{1/2}$$

Now we use the fact that for any random variable Z we have $\mathbb{E}\big[(X - \mathbb{E}[X])^2\big] = \inf_a \mathbb{E}[(Z-a)^2]$. So we can get

$$\mathbb{E}[X]\,\mathbb{E}\big[X^{-1}\big] \leq 1 + \left(\frac{b-a}{2}\right) \cdot \left(\frac{b-a}{2ab}\right) = \frac{(a+b)^2}{4ab}.$$

□

References