Lecture 1: Introduction and Convexity

TTIC 31070 / CMSC 35470 / BUSF 36903 / STAT 31015

Convex Optimization

Prof. Zhiyuan Li

Spring 2026

Why Optimization?

An optimization problem packages a decision task into three objects:

- A **decision variable** x in some space E
- A **feasible set** $\Omega \subseteq E$ (what is allowed)
- An **objective** $f : E \to \mathbb{R}$ (what is preferred)

This language covers: resource allocation, profit maximization, machine learning, maximum-likelihood estimation, control, minimum-energy configurations, . . .

Optimization Problem

Definition 1.1 (Optimization problem). Let $f : E \to \mathbb{R}$, and let $\Omega \subseteq E$ be nonempty. The optimization problem associated with (f, Ω) is

$$p^\star := \inf\{f(x) : x \in \Omega\}.$$

A point $x^\star \in \Omega$ is a **global minimizer** if $f(x^\star) = p^\star$, i.e., $x^\star = \arg\min_{x \in \Omega} f(x)$.

In this course we focus on continuous optimization: E is a finite-dimensional real space.

When Do Minimizers Exist?

Before discussing **certificates of optimality**, it is natural to first ask: *does an exact minimizer even exist?*

Theorem 1.1 (Weierstrass). Let $\Omega \subseteq E$ be nonempty and **compact**, and let $f : \Omega \to \mathbb{R}$ be **continuous**. Then there exists $x^\star \in \Omega$ such that

$$f(x^\star) = \min_{x \in \Omega} f(x).$$

What Else Can Happen?

Even with a well-posed specification, several things can go wrong:

1. The feasible set may be **empty** ($p^\star = +\infty$ by convention)
2. The problem may be **unbounded below** ($p^\star = -\infty$)
3. The infimum may be **finite but not attained**
   Example: $\inf_{x>0} x = 0$, but no $x > 0$ achieves 0.

In case (3), exact minimizers do not exist — but near-optimal points do.

Approximate Optimality

Definition 1.2 (Approximate optimality). For every $\varepsilon > 0$, a feasible point $\widehat{x} \in \Omega$ is **ε-optimal** if

$$f(\widehat{x}) \leq p^\star + \varepsilon.$$

Since exact minimizers may not exist, we need a solution concept that does not depend on their existence.

Specification vs. Computation

The formula

$$p^\star = \inf_{x \in \Omega} f(x)$$

specifies *what* we want, but does not yet define a *computational problem*.

To discuss algorithms we must also specify:

- **How is the instance presented?** (e.g., dense vs sparse matrix)
- **Which operations are available?** (e.g., arithmetic over reals, bit operations, floating point operations, . . .)
- **What cost model?** (operation count, GPU time, rental cost, . . .)

Oracle Access

We care about the **total arithmetic cost** of an optimization algorithm. A clean and general way to study this is through an abstract **oracle model**:

Zeroth-order oracle:	query $x \longrightarrow$ receive $f(x)$
First-order oracle:	query $x \longrightarrow$ receive $\nabla f(x)$
Second-order oracle:	query $x \longrightarrow$ receive $\nabla^2 f(x)$

We assume infinite-precision real arithmetic.

The central question: how many oracle queries are needed to find an ε-optimal point?

Multiply by the cost of each oracle call $\longrightarrow$ total computational complexity.

Differentiability

Definition 1.3 (Differentiability on an open set). Let $U \subseteq E$ be open and $f : U \to \mathbb{R}$. The first-order object at $x \in U$ is the **differential**

$$\nabla f(x) := Df(x) \in E^*,$$

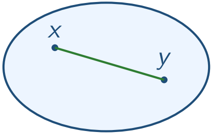
and expressions such as $\langle \nabla f(x), h \rangle := Df(x)[h]$ are **dual pairings**. (no inner product is actually required)

Euclidean special case: if $E = \mathbb{R}^n$ with the standard inner product, $Df(x)$ becomes the familiar gradient vector. We still write $\nabla f(x)$, but the typed meaning is the covector $Df(x) \in E^*$.

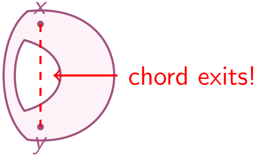
Convex Set

Definition 1.4 (Convex set, convex function, strict convexity). A set $C \subseteq E$ is **convex** if

$$\forall\, x, y \in C,\ \forall\, \theta \in [0,1], \qquad \theta x + (1-\theta) y \in C.$$



convex: chord stays inside

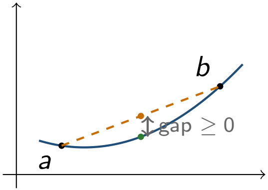


not convex

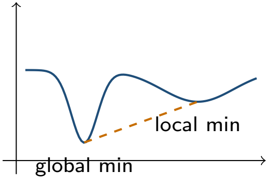
Convex Function

Definition 1.4 (cont.). A function $f : C \to \mathbb{R}$ is **convex** if

$$f\big(\theta x + (1-\theta)y\big) \;\le\; \theta\, f(x) + (1-\theta)\, f(y) \qquad \forall\, x, y \in C,\;\; \theta \in [0,1].$$



✓ convex

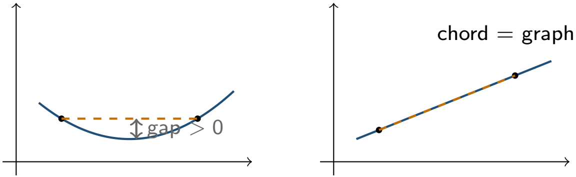


× not convex

Strict Convexity

Definition 1.4 (cont.). f is **strictly convex** if the inequality is strict whenever $x \neq y$ and $\theta \in (0,1)$:

$$f\big(\theta x + (1-\theta)y\big) \;<\; \theta\, f(x) + (1-\theta)\, f(y).$$



✓ strictly convex

✗ convex, not strictly

Strict convexity $\Longrightarrow$ at most one global minimizer. (If there were two, their midpoint would have strictly smaller objective.)

Why Study Convex Optimization? (1/2)

1. **Local information becomes globally meaningful.**
   Without convexity, a gradient near one point says nothing about what happens elsewhere. Under convexity, gradients, subgradients, and dual certificates become *global* lower certificates.

2. **Convexity is broad and stable.**
   Linear programs, least squares, logistic regression, SDPs, regularized learning models are all convex. Convexity is preserved under nonneg. combinations, affine maps, epigraph constructions, partial minimization, and conjugation.

Why Study Convex Optimization? (2/2)

3. **The right baseline for algorithms and complexity.**
   Even for nonconvex applications, convex optimization is the cleanest setting to isolate local primitives, prove global guarantees, and identify complexity barriers.

4. **A source of ideas that survive beyond convexity.**
   Example: modern LLM training is highly nonconvex, yet the default optimizer **AdamW** descends from **AdaGrad**, which emerged from *online convex optimization* theory.

DF] **Adaptive subgradient methods for** online learning and stochastic
ptimization.
uchi, E Hazan, Y Singer - Journal of machine learning research, 2011 - jmlr.org
We present a new family of **subgradient methods** that dynamically incorporate knowledge of
… **adaptive subgradient methods** outperform state-of-the-art, yet non-**adaptive**, **subgradient** …
Save Cite Cited by 16389 Related articles All 26 versions

Adam: A method for stochastic optimization
DP Kingma, J Ba - arXiv preprint arXiv:1412.6980, 2014 - arxiv.org
… **Adam** works well in practice and compares favorably to other stochastic optimization methods.
Finally, we discuss AdaMax, a variant of **Adam** … Overall, we show that **Adam** is a versatile …
Save Cite Cited by 239712 Related articles All 14 versions

Example: Least Squares

Example 1.1 (Least squares). Given data $(a_i, b_i)_{i=1}^m$ with $a_i \in \mathbb{R}^d$:

$$\min_{x \in \mathbb{R}^d} \frac{1}{2m} \sum_{i=1}^m \left(a_i^\top x - b_i\right)^2$$

- Closed-form solution via the normal equation:

$$\hat{x} = (A^\top A)^{-1} A^\top b, \qquad A = \begin{pmatrix} a_1^\top \\ \vdots \\ a_m^\top \end{pmatrix}$$

- But this requires forming and inverting $A^\top A$:
  - d large $\Rightarrow$ $O(d^3)$ time and $O(d^2)$ space
  - m large $\Rightarrow$ forming $A^\top A$ requires a full pass over all data

$\Rightarrow$ Iterative algorithms (gradient descent, SGD) are needed in the large-data regime

Example: Logistic Regression

Example 1.2 (Logistic regression). Given labels $y_i \in \{\pm 1\}$:

$$\min_{x \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 + e^{-y_i a_i^\top x}\right) + \frac{\lambda}{2} \|x\|_2^2$$

- Convex, but usually **no closed-form** minimizer
- Value and gradient are both natural to evaluate
- **Convexity $\neq$ closed form**; it means local information can become globally meaningful

Example: Constrained Quadratic Program

Example 1.3 (Constrained QP). Let $Q \succeq 0$, $b \in \mathbb{R}^n$, $\Omega := \{x \in \mathbb{R}^n : Cx = d,\ x \geq 0\}$.

$$\min\left\{\frac{1}{2}x^\top Qx + b^\top x \ : \ x \in \Omega\right\}$$

- Convex optimization with both objective geometry and explicit constraints
- Previews later themes: constrained optimality, dual variables, KKT conditions

First Consequences: the Certificate Viewpoint

Under convexity, local information becomes a **certificate of global optimality**. The next four results make this precise:

1. **Local $=$ global:** a local minimizer is already a global minimizer.
2. **First-order necessary condition:** at a minimizer, the directional derivative is nonnegative in every feasible direction.
3. **Global linear lower bound:** differentiability $+$ convexity give $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$ everywhere.
4. **First-order $\Leftrightarrow$ optimal:** the sign condition is not only necessary but also sufficient.

Local Minima Are Global

Theorem 1.2 (Local minima are global). Let Ω be nonempty and convex, $f : \Omega \to \mathbb{R}$ convex. If $x^\star$ is a **local** minimizer, then $x^\star$ is a **global** minimizer.

Proof idea. Suppose some $x \in \Omega$ has $f(x) < f(x^\star)$. Form $x_\theta = \theta x + (1-\theta)x^\star$. By convexity, $f(x_\theta) < f(x^\star)$. For small θ, $\|x_\theta - x^\star\| < r$ — contradicting local minimality. □

First-Order Necessary Condition

Lemma 1.3 (First-order necessary condition). Let Ω be nonempty and convex, $f : E \to \mathbb{R}$ differentiable, $x^\star$ a global minimizer of f over Ω. Then

$$\forall\, x \in \Omega, \qquad \langle \nabla f(x^\star), x - x^\star \rangle \geq 0.$$

Proof idea. Restrict f to the line segment $\phi(t) = f\big(x^\star + t(x - x^\star)\big)$. Global minimality forces $\phi(t) \geq \phi(0)$ on $[0, 1]$, so $\phi'_+(0) = \langle \nabla f(x^\star), x - x^\star \rangle \geq 0$. $\square$

Global Linear Lower Bound

Lemma 1.4 (Gradient lower bound). Let $\Omega \subseteq E$ be nonempty and convex, $f : E \to \mathbb{R}$ differentiable and convex on Ω. Then

$$\forall\, x, y \in \Omega, \qquad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle.$$

Proof idea. The one-variable function $\psi(t) = f(x + t(y - x))$ is convex on $[0, 1]$. A convex function of one variable satisfies $\psi(1) \geq \psi(0) + \psi'(0)$. $\square$

First-Order Characterization

Theorem 1.5 (Differentiable convex first-order characterization). Let $\Omega \subseteq E$ be nonempty and convex, $f : E \to \mathbb{R}$ differentiable and convex on Ω, $x^\star \in \Omega$. Then:

1. $f(x^\star) \leq f(x)$ for every $x \in \Omega$ ($x^\star$ is a global minimizer)
2. $\langle \nabla f(x^\star), x - x^\star \rangle \geq 0$ for every $x \in \Omega$ (first-order condition)

are equivalent.

Proof idea. (1) $\Rightarrow$ (2): apply Lemma 1.3. (2) $\Rightarrow$ (1): apply Lemma 1.4 to get $f(x) \geq f(x^\star) + \langle \nabla f(x^\star), x - x^\star \rangle \geq f(x^\star)$. $\square$

Summary & What's Next

Today:

- Optimization = variable + feasible set + objective
- Existence under compactness (Weierstrass)
- Convexity: local $\Rightarrow$ global; first-order condition $\Leftrightarrow$ optimal
- Oracle model: how many queries to reach ε-optimality?

Next lecture:

- Separation theorems and supporting hyperplanes
- Subgradients: first-order information without smoothness
- The geometric backbone of convex optimization

Course Goals

This course covers three things:

1. **Formalize** concrete optimization problems as convex programs
2. **Analyze** structural properties of convex optimization
   (duality, separation, KKT conditions, conic geometry)
3. **Design and analyze** first-order and second-order algorithms
   (convergence rates, oracle complexity, lower bounds)

Course Roadmap (18 Lectures)

Part I — Duality (L1–6) Convexity, separation, LP, conjugates, KKT, conic optimization

Part II — First-Order Methods (L7–15) Cutting planes, steepest descent, mirror descent, stochastic methods, adaptive methods, Frank–Wolfe, lower bounds, acceleration

Part III — Second-Order Methods (L16–18) Newton's method, self-concordance, interior-point methods

Schedule is subject to change.

Course Information

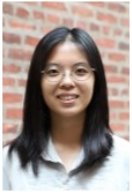
TTIC 31070 / CAAM 31015 / CMSC 35470 / BUSF 36903 / STAT 31015

- Course website:
  `https://zhiyuan-li.github.io/teaching/convex-optimization-2026/`
- Canvas:
  `https://canvas.uchicago.edu/courses/71971`

Teaching Team

Instructor: Zhiyuan Li (`https://zhiyuanli.ttic.edu/`)

TAs:



Shuo Xie

Marko Medvedev

Beining Wu

Schedule & Office Hours

- **Lectures:** Tuesday & Thursday, 2:00–3:20 PM, TTIC 530
- **Instructor OH:** Tu & Th, 3:20–3:50 PM, TTIC 508
  (right after class — just walk over with me)
- **TA OH:** TBD (will be announced on Canvas)
- No recitation this quarter

Grading & Homework

- Homework will be posted regularly
- We assume everyone can score full marks with the help of language models, so there is no need to grade it

- Evaluation:

$$\underbrace{40\%}_{\text{Homework } (=100\%)} + 60\% \text{ Final Exam } + \text{ Bonus points}$$

- The final exam will very likely sample from the homework

Communication

- **Course material questions** $\rightarrow$ Canvas, OH, or in class
  Do NOT send direct emails — we will not answer content questions via email.

- **Homework clarifications** $\rightarrow$ Canvas (preferred) or TA OH
- **Homework help** $\rightarrow$ TA office hours

- **General feedback** $\rightarrow$ Canvas (can be anonymous)
- **Staff email:** `ttic-31070-convex-optimization-2026@ttic.edu`

LLM Usage Policy

You are welcome to use LLMs (ChatGPT, Claude, Gemini, . . .) **freely** throughout this course, unless explicitly stated otherwise.

We **encourage** it as an active learning tool:

- Ask an LLM to explain a concept you find confusing
- Request worked examples or step-by-step proof walkthroughs
- Follow threads depth-first: when one answer raises a new question, dig deeper before moving on

Caveat: LLMs can produce plausible-sounding but incorrect mathematics. Always verify non-trivial claims against the notes or your own reasoning.

Formal Verification

As an experiment this year, I am attempting to **formally verify** all major theorems in the lecture notes using **Lean 4**, a formal verification tool based on dependent type theory.

- Every theorem statement and proof is **machine-checked**
  — if it compiles, the proof is mathematically correct
  — still need to check whether the statement is faithful to the notes
- **Bonus points** for spotting formalization bugs and contributing formal proofs
- Verified proofs are linked from the **course website**

Special tutorial on using Lean this Friday, 3–4 PM, by **Richard Xu**