SDSC6015 Stochastic Optimization for Machine Learning

Lu Yu

Department of Data Science, City University of Hong Kong

September 4, 2025

First part of the lecture

- What is this class about?
- Administration details

Course Information

This course

- Explores contemporary optimization techniques, with a strong emphasis on their applications in machine learning and data science.
- Focus on both the theoretical foundations and applied analysis of various optimization algorithms.
- Aimed at master-level graduate students, provides a rigorous yet accessible introduction to the field.
- Prerequisites: no formal ones, but assume basic knowledge in
  - calculus, linear algebra, analysis
  - mathematical thinking and modeling

Learning Outcomes

By the end of the course, the students will be able to:

- Evaluate the key algorithms, function classes, and their convergence guarantees.
- Formulate scalable and accurate implementations of the optimization algorithms for machine learning applications.

Course Information

- We will use Canvas for announcements & grades & discussion, etc.
- Lecture slides will be posted on Canvas! Please do let us know about typos you notice and/or any suggestions you might have.
- Office hours will be scheduled prior to the submission of assignments, the midterm exam, and the final exam.
- Questions during lectures are always welcome!

Literatures

Recommended readings will be given for each lecture. The following will be useful throughout the course:

- *Convex Optimization: Algorithms and Complexity*, by Sébastien Bubeck
- *Convex Optimization*, by Stephen Boyd and Lieven Vandenberghe
- *Introductory Lectures on Convex Optimization*, by Yurii Nesterov
- *First-Order and Stochastic Optimization Methods for Machine Learning*, by Guanghui Lan

Requirements and Marking

- Midterm exam: 2 hours; in class on October 9, 2025; worth 30% of course mark.
  - Otherwise, course project (optional, 10%), midterm exam (20%) Please send me an email before Week 8 (Oct 16)
- Final exam: 2-3 hours; date and time TBA; worth 40% of course mark.
- Exam questions are conceptual/theoretical; no coding.
- **Everybody must take the final exam! No exceptions.**

More on Assignments

- Three homework assignments: Combination of pen & paper derivations and coding exercises; equally weighted for a total of 30%.
- The assignments will be posted on Canvas.
- Collaboration on assignments is permitted. After attempting the problems individually, you may discuss and work on the homework with up to two classmates. However, you must independently write your own code and provide your own written solutions. Additionally, you must explicitly list any collaborators at the top of your submission.
- Assignments should be handed in by the deadline; a late penalty of 10% **per day** will be assessed thereafter (up to 3 days, then the submission is blocked).

More on Assignments (cont'ed)

- Extensions will be granted only under exceptional circumstances. To request an extension, you must submit your case along with supporting documentation via AIMS and **inform the instructor** accordingly.
- You will be using Python on assignments.

Provisional Calendar (tentative)

- Week 1, Sep 4
  - Introduction
  - Preliminaries of Stochastic Optimization
- Week 2, Sep 11
  - Convex Function
  - Convex Optimization Problems
- Week 3, Sep 18
  - Gradient Descent
  - Projected/Proximal Gradient Descent
  - Assignment 1 released on Sep 18, due on Oct 1
- Week 4, Sep 25
  - Subgradient Descent
  - Mirror Descent

Provisional Calendar (cont'ed)

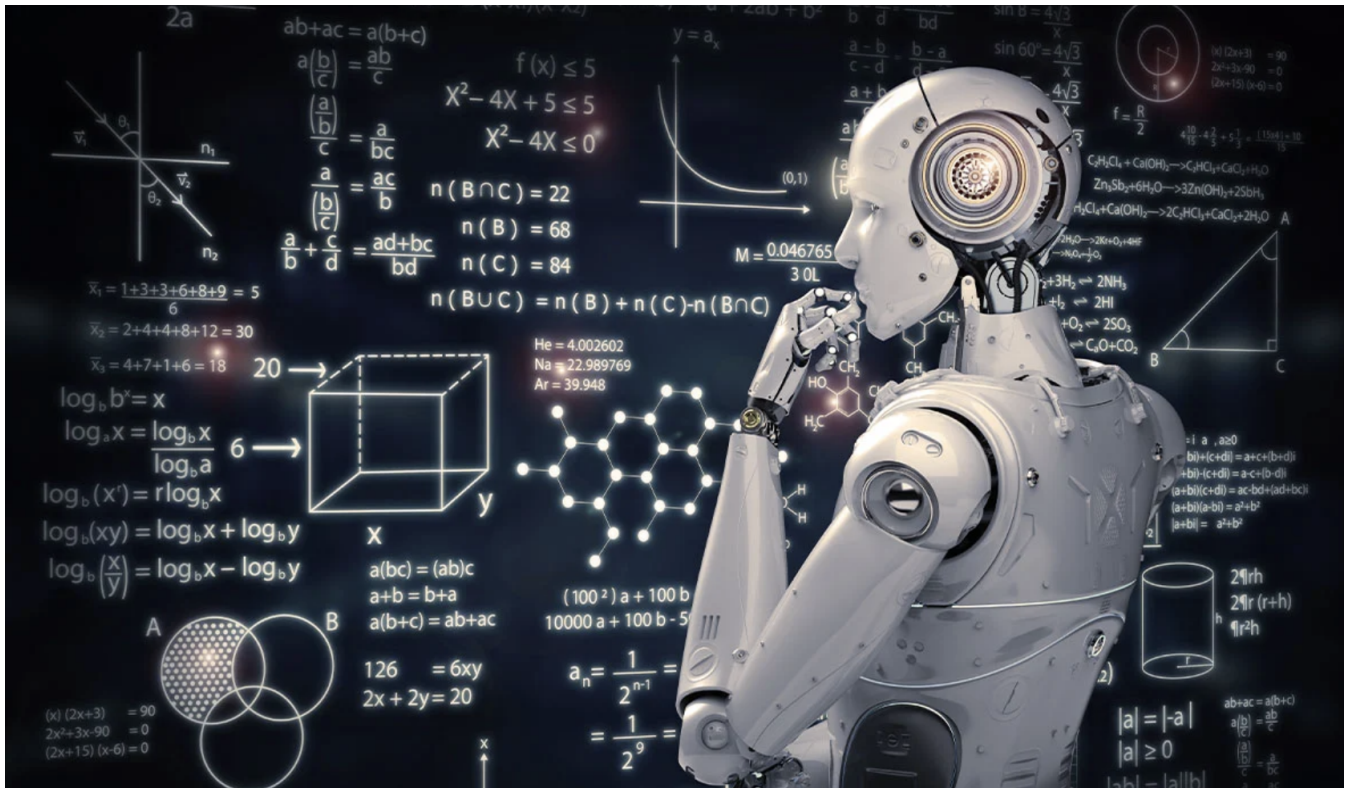
- Week 5, Oct 2
  - Stochastic Gradient Descent (SGD)
  - Momentum-based Methods
- Week 6, Oct 9
  - Midterm exam
- Week 7, Oct 16
  - Adaptive Learning Rates Methods
  - Other Variants of SGD
  - Assignment 2 released on Oct 16, due on Oct 29
- Week 8, Oct 23
  - Coordinate Descent
  - Newton Method, Quasi-Newton Method

Provisional Calendar (cont'ed)

- Week 9, Oct 30
  - Zero-th Order Optimization
  - Parallel and Distributed Optimization
- Week 10, Nov 6
  - Nonconvex Optimization
  - Applications in Machine Learning
- Week 11, Nov 13
  - Robust and Adversarial Optimization
  - Assignment 3 released on Nov 13, due on Nov 26
- Week 12, Nov 20
  - Advanced Topics in Optimization for Machine Learning
  - Final exam review
- Week 13, Nov 27, recap
- TBA: Final Exam

What is Machine Learning?

- Machine learning approach: program an algorithm to automatically learn patterns from data or experience.



What is Machine Learning? (cont'ed)

- Example 1: Recognizing handwritten numbers after being shown many examples.

What is Machine Learning? (cont'ed)

- Example 2: Predicting house prices based on features such as size, location, and number of bedrooms.

Optimization for Machine Learning

Optimization problems underlie nearly everything we do in Machine Learning.

In many courses, you learn how to translate the conceptual idea into an optimization problem:

$$P \ : \ \min_{\theta \in \mathcal{D}} f(\theta)$$

This course: how to solve P, and why this is a good skill to have

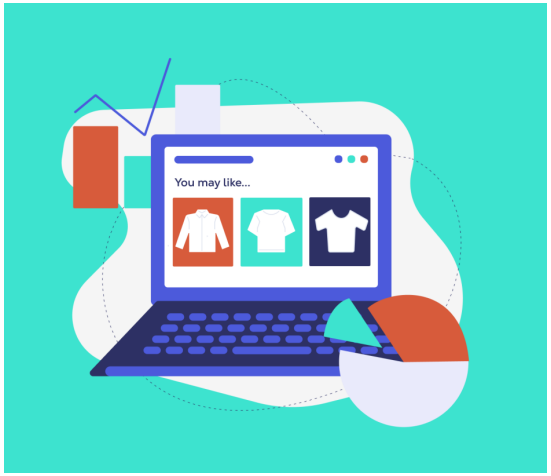
Stochastic Optimization for Machine Learning

- Stochastic optimization is a method/algorithm used to efficiently train machine learning models, especially on large datasets.
- Instead of using all the data at once, stochastic optimization uses small, randomly selected parts of the data to update the model step by step.
- This approach makes training faster and more practical!

Why is Stochastic Optimization Important?

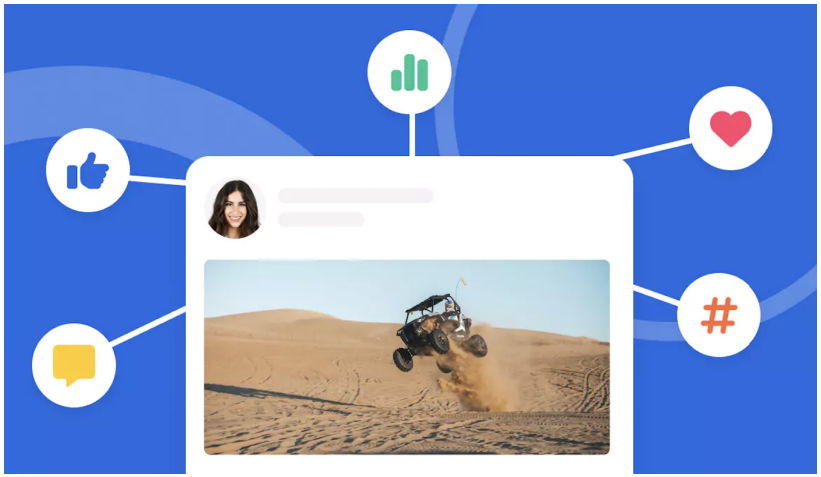
- Handles Large Datasets: Instead of working with massive data all, it focuses on small, manageable parts.
- Efficient Learning: By making updates step by step, it learns patterns quickly without the full dataset.
- Real-World Applications: It's widely used in tasks like training recommendation systems, self-driving cars, and speech recognition, etc.

Online Shopping Recommendation System



- **Challenge:** The dataset includes billions of user-product interactions, which is computationally expensive to process all at once.
- **Stochastic Optimization in Action:** Stochastic optimization methods use small random samples of data (e.g., a few user interactions).
  - The algorithm updates the model incrementally.
  - Recommendations improve as more samples are processed.

Social Media Feed Ranking



- **Challenge:** The platforms analyze billions of posts, likes, and shares to decide what content to show each user. Calculating rankings for everyone in real-time is computationally infeasible.
- **Stochastic Optimization in Action:** The ranking algorithm uses random samples of user interactions (e.g., likes, comments) to train the model:
  - It learns what types of posts users engage with.
  - The algorithm updates ranking rules incrementally using small subsets of data.

Navigation Apps

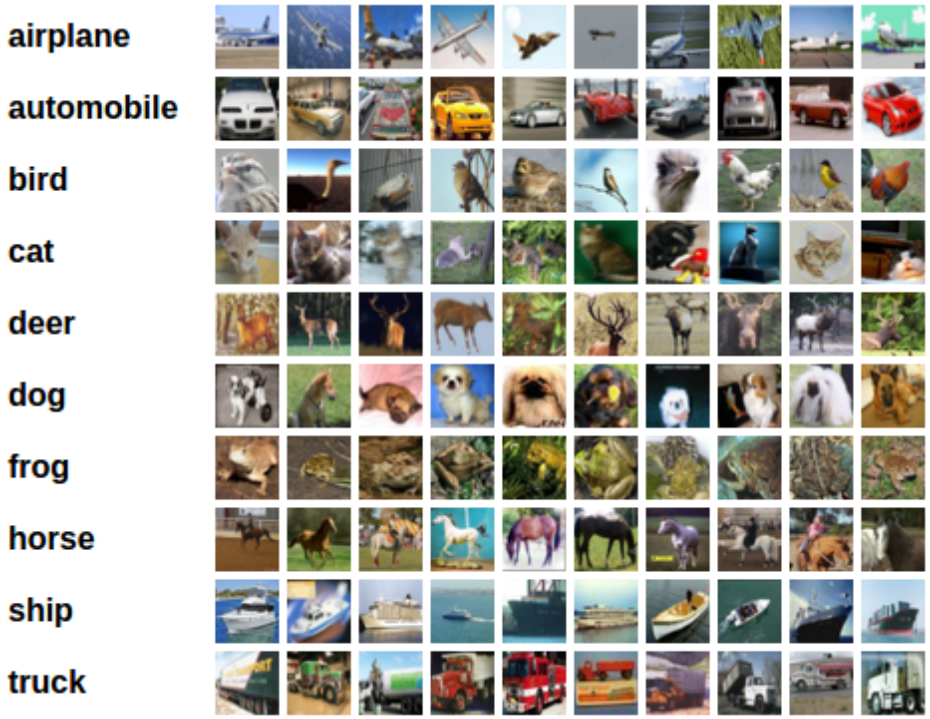
- **Challenge:** Traffic data changes every second, and there are millions of users navigating simultaneously. Calculating the optimal route for every user at once is too slow.
- **Stochastic Optimization in Action:** The app collects real-time traffic samples (e.g., a random subset of cars on the road) to estimate congestion levels.
  - It updates route recommendations in near real-time.
  - It doesn't need to process all traffic data at once.

Key Takeaways

- **Stochastic Optimization = Small Steps with Random Data**: Instead of processing the entire dataset, stochastic optimization works with small random subsets.
- **Why It Works in Real Life:**
  - Datasets in real-world applications are massive (e.g., millions of users or sensors).
  - Stochastic optimization helps machine learning models learn efficiently without processing all the data at once.
- **Analogy:** Consider learning to cook by preparing one randomly chosen recipe at a time, rather than reading every recipe in a cookbook. Gradually, the cooking skills will be improved through practice without the need to study the entire book!

Questions?

Main problem

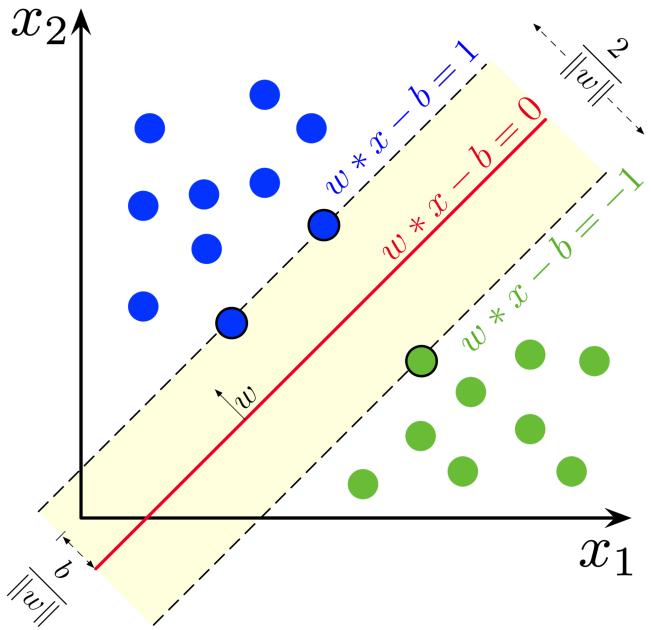


Given a set of labeled training data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times \mathcal{Y}$, find the set of weights θ which classifies the data via

$$\text{minimize } f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, (\boldsymbol{x}_i, y_i)), \quad n \text{ large.}$$

Example: Classification

- **Soft Margin SVM (support vector machine)**



$$\min \left\{ \frac{1}{n} \sum_{i=1}^{n} \max \left(0, 1 - y_i \left(\mathbf{w}^\top \boldsymbol{x}_i - b \right) \right) \right) + \lambda \left\| \mathbf{w} \right\|^2 \right\},$$

- $\boldsymbol{x}_i$: predictor vector (feature)
- $y_i \in \{-1, 1\}$: label/class
- n: number of data points, $\lambda > 0$
- $\mathbf{w}$: parameter to be learned

Example: Image Denoising

- **Total Variation Denoising Model**

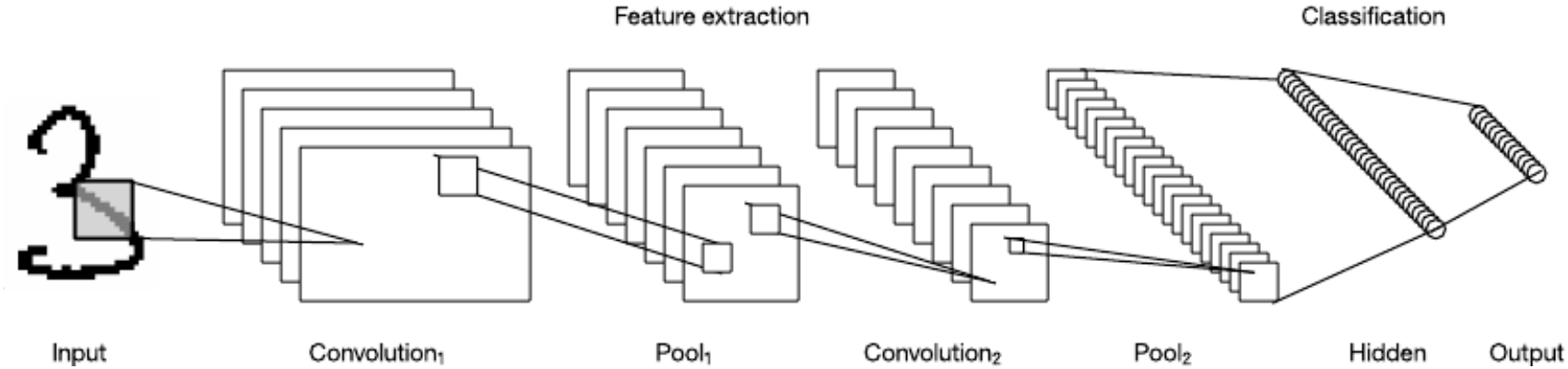
$$\min_X \sum_{(i,j)\in P} |X_{ij} - O_{ij}|^2 + \lambda \cdot TV(X)\,,$$

- X is an image matrix
- O is the noisy image, P is the ob observed entries
- $TV(X)$ is the total variation

$$TV(X) = \sum_{i,j} |X_{i+1,j} - X_{ij}| + |X_{i,j+1} - X_{ij}|$$

Example: Neural Networks

- **Convolutional Neural Network**



$$\text{minimize } f(\mathbf{w}, \mathbf{b}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{b}, (\boldsymbol{x}_i, y_i)),$$

- $\ell(\mathbf{w}, \mathbf{b}, (\boldsymbol{x}_i, y_i)) = \|\mathbf{w}_3^\top \cdot Q(\mathbf{w}_2^\top \cdot Q(\mathbf{w}_1^\top \boldsymbol{x}_i - \mathbf{b}_1) - \mathbf{b}_2) - y_i\|^2$
- $Q(\mathbf{w})_i = \max(0, \mathbf{w}_i)$ (ReLU)
- $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3$ are weight matrices
- $\mathbf{b}_1, \mathbf{b}_2$ are bias vectors
- $\boldsymbol{x}_i$: input data, y_i : output targets

Simple Numerical Example - Least Square Regression

Problem Setup

Consider the least squares problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) := \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{y}\|^2 ,$$

where

- $A \in \mathbb{R}^{n\times d}$ is a data matrix
- $\boldsymbol{y} \in \mathbb{R}^n$ is the observation vector
- $\boldsymbol{x} \in \mathbb{R}^d$ is the variable to optimize

Simple Numerical Example - Least Square Regression

Key idea

- **Gradient Descent (GD)** computes the gradient using all n data points at each step:

$$\nabla f(\boldsymbol{x}) = A^\top (A\boldsymbol{x} - \boldsymbol{y})\,.$$

  This is computationally expensive for large n.

- **Stochastic Gradient Descent (SGD)** approximates the gradient using only one randomly chosen data point at each step:

$$\nabla f_i(\boldsymbol{x}) = a_i^\top (a_i \boldsymbol{x} - y_i)\,,$$

  where a_i is the i-th row of A and y_i is i-th entry of $\boldsymbol{y}$.
  This reduces computation per step from $O(nd)$ to $O(d)$, making SGD efficient for large datasets.

Simple Numerical Example - Least Square Regression

Key idea

- **Gradient Descent (GD)** makes use of the full gradient $\nabla f(\boldsymbol{x})$ via

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \nabla f(\boldsymbol{x}), \qquad t = 0, 1, \ldots$$

- **Stochastic Gradient Descent (SGD)** makes use of the stochastic gradient $\nabla f_t(\boldsymbol{x})$ via

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \nabla f_t(\boldsymbol{x}), \qquad t = 0, 1, \ldots$$

- $\eta > 0$ denotes the step size (learning rate)

Simple Numerical Example - Least Square Regression

We simulate a problem with:

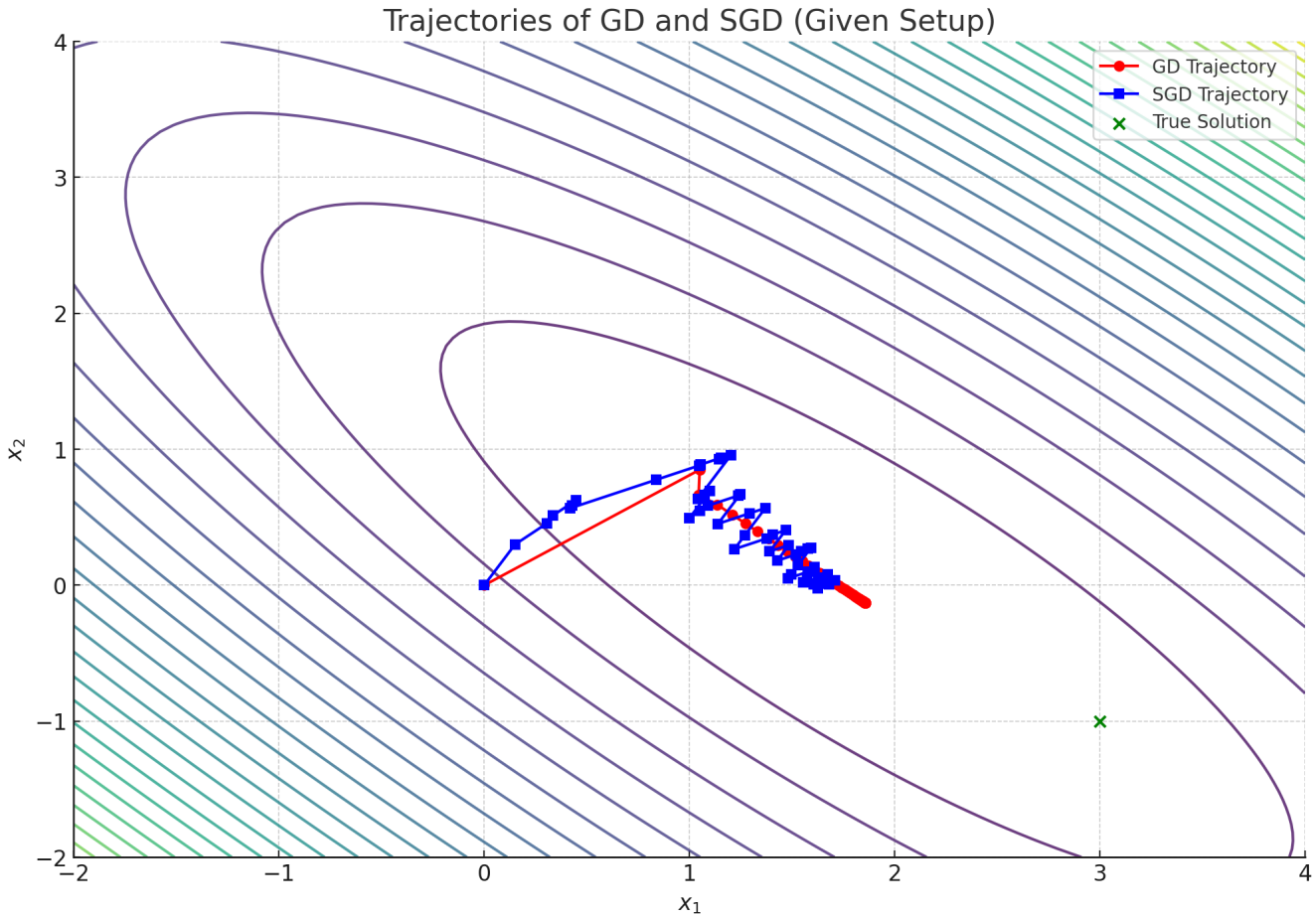
- $n = 3$ (small sample size)
- $d = 2$ (low dimension)
-

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 1 \end{bmatrix}, \quad y = \begin{bmatrix} 1.5 \\ 3.5 \\ 2.0 \end{bmatrix}$$

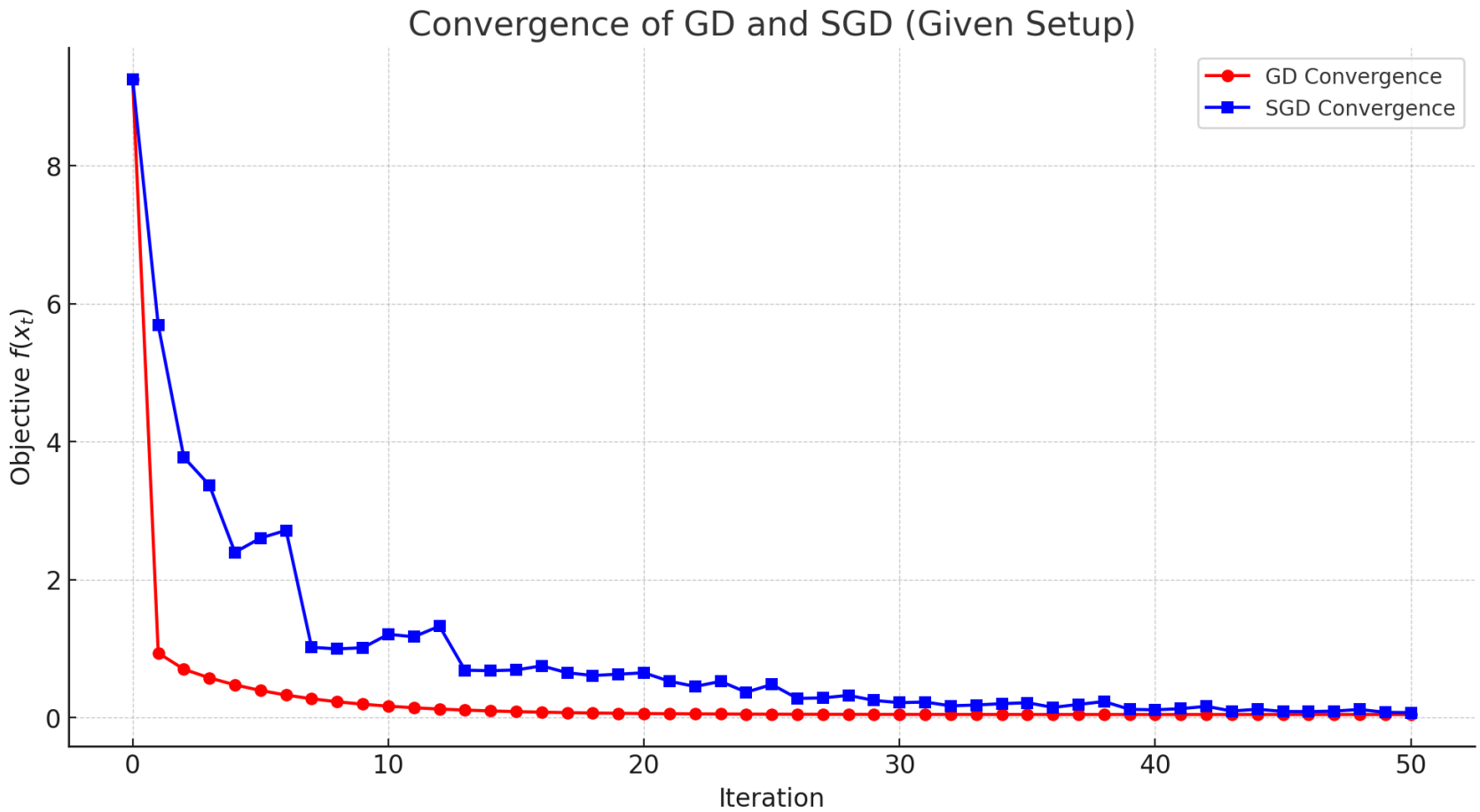
We use:

- Step size: $\eta = 0.1$
- Initialization: $\boldsymbol{x}_0 = (0, 0)$

Simple Numerical Example - Least Square Regression



Simple Numerical Example - Least Square Regression



Simple Numerical Example - Least Square Regression

Key takeaways

- **Gradient Descent:**
  - Computes full gradients at every step.
  - Converges smoothly but can be computationally expensive.
- **Stochastic Gradient Descent:**
  - Uses one random data point per step, reducing computation.
  - Converges with more noise but can reach the solution faster in terms of total computation.

Preliminaries of Stochastic Optimization

Warmup: The Cauchy-Schwarz inequality

Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$. Cauchy-Schwarz inequality

$$|\mathbf{u}^\top \mathbf{v}| \leqslant \|\mathbf{u}\| \|\mathbf{v}\|$$

Notation:

- $\mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_d \end{pmatrix}, \mathbf{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_d \end{pmatrix}$, d-dimensional column vectors with real entries.
- $\mathbf{u}^\top = (u_1, \cdots, u_d)$, transpose of $\mathbf{u}$, a d-dimensional row vector.
- $\mathbf{u}^\top \mathbf{v} = \sum_{i=1}^d u_i v_i$, scalar (or inner) product of $\mathbf{u}$ and $\mathbf{v}$.
- $|\mathbf{u}^\top \mathbf{v}|$, absolute value of $\mathbf{u}^\top \mathbf{v}$.
- $\|\mathbf{u}\| = \sqrt{\mathbf{u}^\top \mathbf{u}} = \sqrt{\sum_{i=1}^d u_i^2}$, Euclidean norm of $\mathbf{u}$.

Warmup: The Cauchy-Schwarz inequality

Cauchy-Schwarz inequality: $|\mathbf{u}^\top \mathbf{v}| \leqslant \|\mathbf{u}\|\|\mathbf{v}\|, \qquad \mathbf{u}, \mathbf{v} \in \mathbb{R}^d.$

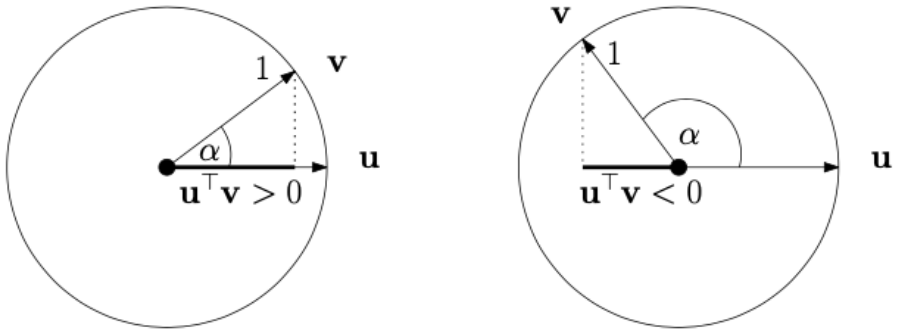
For nonzero vectors, this is equivalent to

$$-1 \leqslant \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \leqslant 1\,.$$

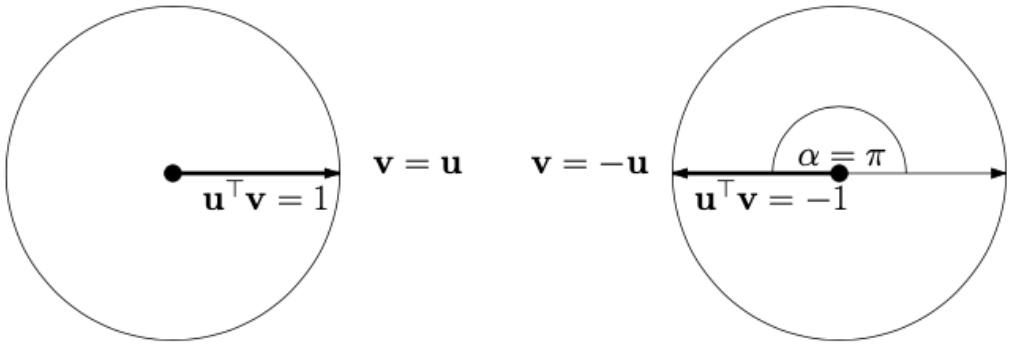
Fraction can be used to define the angle α between $\mathbf{u}$ and $\mathbf{v}$:

$$\cos(\alpha) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

Warmup: The Cauchy-Schwarz inequality



Examples for unit vectors ($\|\mathbf{u}\| = \|\mathbf{v}\| = 1$)



Equality in Cauchy-Schwarz if and only $\mathbf{u} = \mathbf{v}$ or $\mathbf{u} = -\mathbf{v}$.

Hölder's Inequality

Let $p, q > 1$ such that $\frac{1}{p} + \frac{1}{q} = 1$, and let $\boldsymbol{x}, \boldsymbol{y}$ be vectors or functions. Hölder's inequality states:

$$\sum_{i=1}^{n} |x_i y_i| \leq \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \left(\sum_{i=1}^{n} |y_i|^q \right)^{1/q},$$

where x_i is the i-th element of $\boldsymbol{x}$, y_i is the i-th element of $\boldsymbol{y}$,

- The Cauchy-Schwarz inequality is a special case of Hölder's inequality when $p = q = 2$.

Hölder's Inequality

$$\sum_{i=1}^{n} |x_i y_i| \leq \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \left(\sum_{i=1}^{n} |y_i|^q \right)^{1/q}.$$

- The ℓ_p-norm of a vector $\boldsymbol{x}$ is defined as:

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}.$$

  The larger p, the more weight is given to the larger components of $\boldsymbol{x}$.

- Intuition: Hölder's inequality provides a way to bound the interaction between two vectors or functions $\boldsymbol{x}$ and $\boldsymbol{y}$ by measuring their "sizes" in terms of ℓ_p-norm and ℓ_q-norm.

Convex Sets

A set $\mathcal{C}$ is convex if the line segment between any two points of $\mathcal{C}$ lies in $\mathcal{C}$, i.e., if for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$ and any λ with $0 \leqslant \lambda \leqslant 1$, we have

$$\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in \mathcal{C}$$

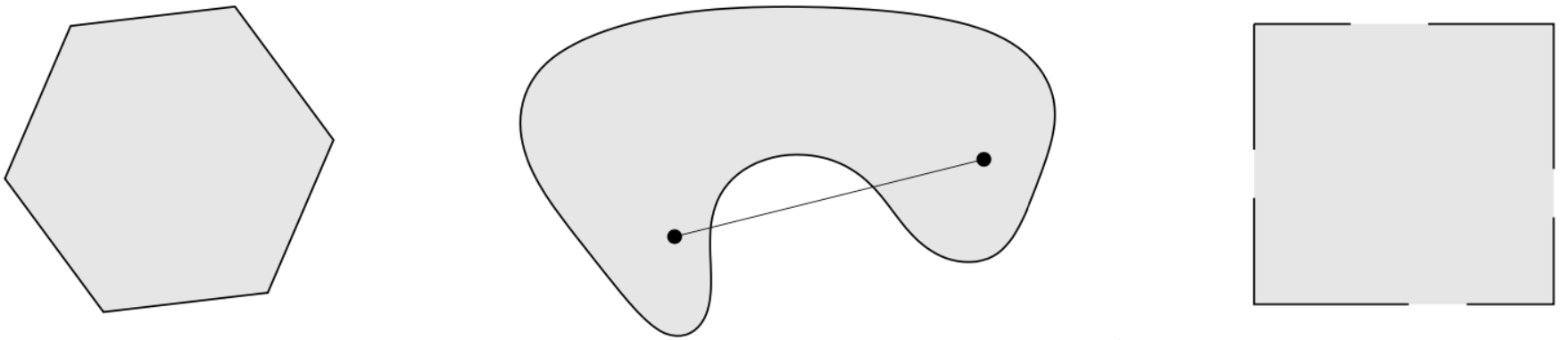


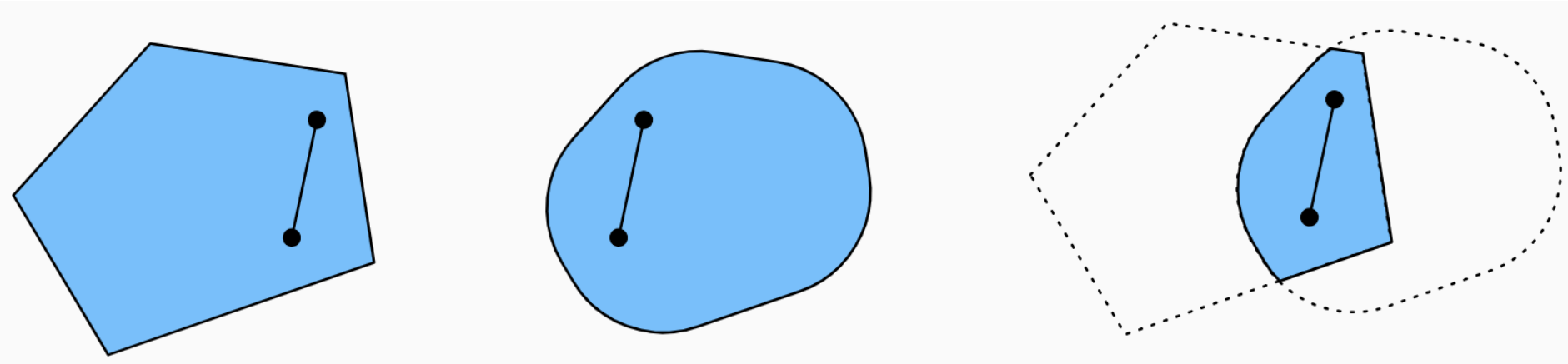
*Figure 2.2 from S. Boyd, L. Vandenberghe

Left: Convex.
Middle: Not convex, since the line segment not in set.
Right: Not convex, since some, but not all boundary points are contained in the set.

Properties of Convex Sets

- Intersections of convex sets are convex.

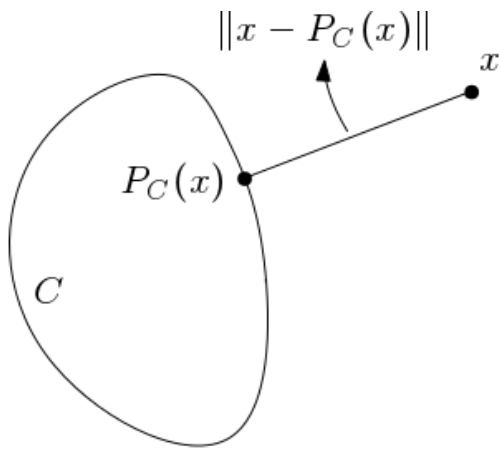
  Obervation: Let $\mathcal{C}_i, i \in I$ be convex sets, where I is a (possibly infinite) index set. Then, the set $\mathcal{C} = \bigcap_{i \in I} \mathcal{C}_i$ is convex.



Properties of Convex Sets

- Definition: projection onto the convex set $\mathcal{C}$

$$\mathrm{P}_{\mathcal{C}}(x) = \arg\min_{y \in \mathcal{C}} \|y - x\|$$



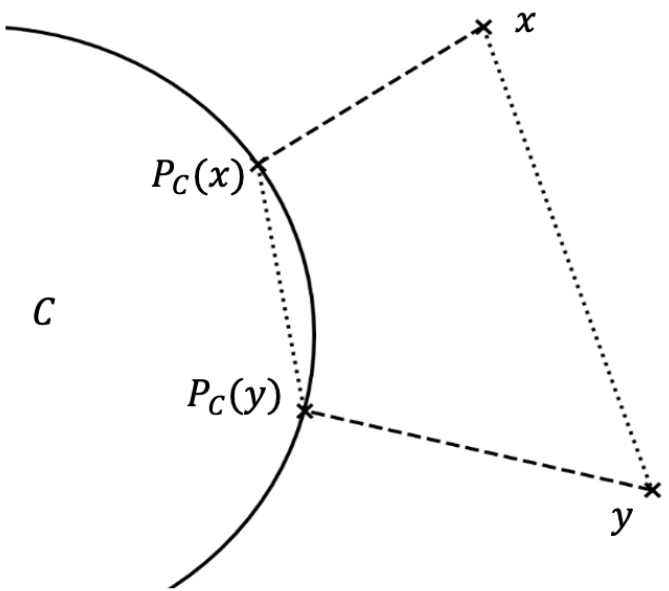
- Projections onto nonempty, closed, and convex sets are unique.

Properties of Convex Sets

- For a projection onto a convex set $\mathcal{C}$, it holds that

$$\|P_{\mathcal{C}}(x) - P_{\mathcal{C}}(y)\| \leqslant \|x - y\|, \qquad \forall x, y$$

- This property is called non-expansiveness: the projection does not increase the distance between any two points.



Lipschitz continuity

Definition

A function $f : \mathbb{R}^d \to \mathbb{R}$ is called L-Lipschitz if for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, we have

$$\|f(\boldsymbol{x}) - f(\boldsymbol{y})\| \leqslant L\|\boldsymbol{x} - \boldsymbol{y}\| .$$

- Intuitively, L is a measure of how fast the function can change.
- Example: $f(x) = \sqrt{x^2 + 5}, \quad f(\boldsymbol{x}) = \|\boldsymbol{x}\|$

Differentiable Functions

- A function $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable at a point $\boldsymbol{x}_0$ if it can be well-approximated by a linear function near that point.
- There exists a gradient $\nabla f(\boldsymbol{x}_0)$ such that

$$f(\boldsymbol{x}_0 + \mathbf{h}) \approx f(\boldsymbol{x}_0) + \nabla f(\boldsymbol{x}_0) \cdot \mathbf{h}\,,$$

where $\mathbf{h}$ is a small change around $\boldsymbol{x}_0$.

Differentiable Functions

- If f is differentiable at every point in its domain, it is called a differentiable function.
- The graph of a differentiable function has a non-vertical tangent line at each interior point in its domain.

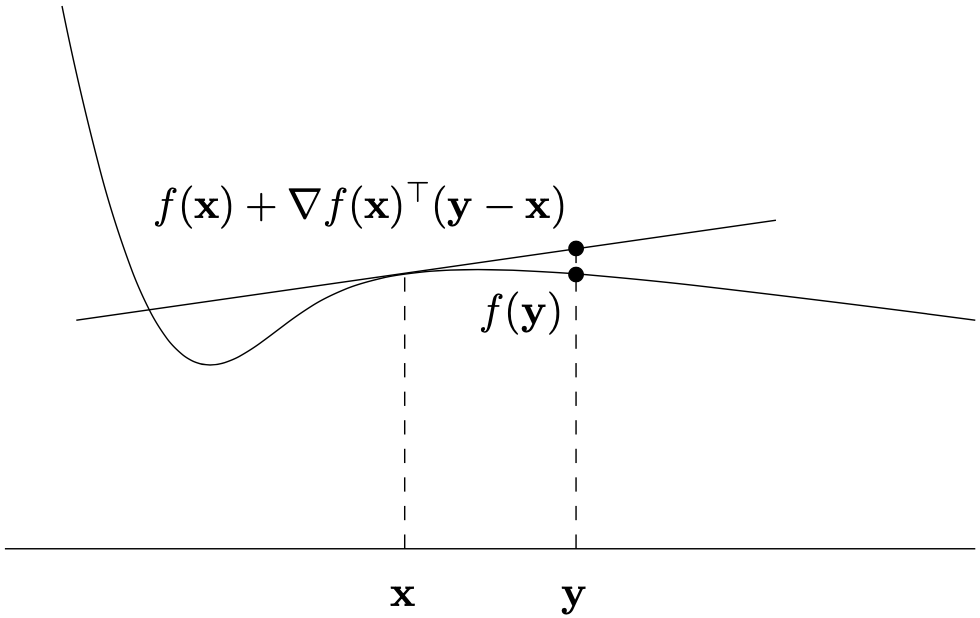
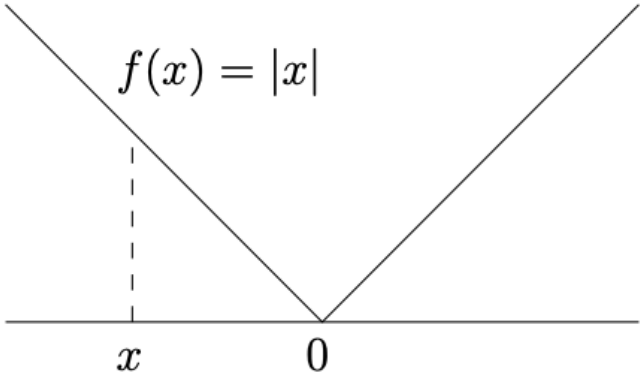


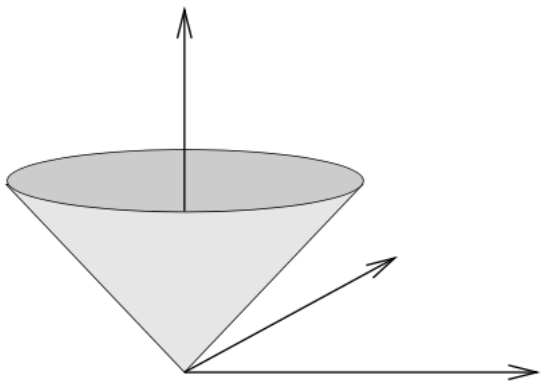
Figure: Graph of the affine function $f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$ is a tangent hyperplane to the graph of f at $(\boldsymbol{x}, f(\boldsymbol{x}))$.

Nondifferentiable Functions

are also relevant in practice.



More generally, $f(\boldsymbol{x}) = \|\boldsymbol{x}\|$ (Eucliean norm). For $d = 2$, graph is the ice cream cone:



L-smoothness

Definition

A function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth if

1. f is continuously differentiable (i.e., the gradient ∇f exists and is continuous).
2. The gradient ∇f is L-Lipschitz

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leqslant L\|\boldsymbol{x} - \boldsymbol{y}\|, \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$$

- the gradient of f (vector of partial derivatives)

$$\nabla f(\boldsymbol{x}) := \left(\frac{\partial f}{\partial x_1}(\boldsymbol{x}), \ldots, \frac{\partial f}{\partial x_d}(\boldsymbol{x}) \right)$$

- $\nabla f(\boldsymbol{x})$ is continuous if

$$\lim_{\boldsymbol{y} \to \boldsymbol{x}} \|\nabla f(\boldsymbol{y}) - \nabla f(\boldsymbol{x})\| = 0, \qquad \forall \boldsymbol{x} \in \mathbb{R}^d.$$

L-smoothness

- Example: $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|^2 = \frac{1}{2}\sum_{i=1}^{d} x_i^2$ is 1-smooth function.

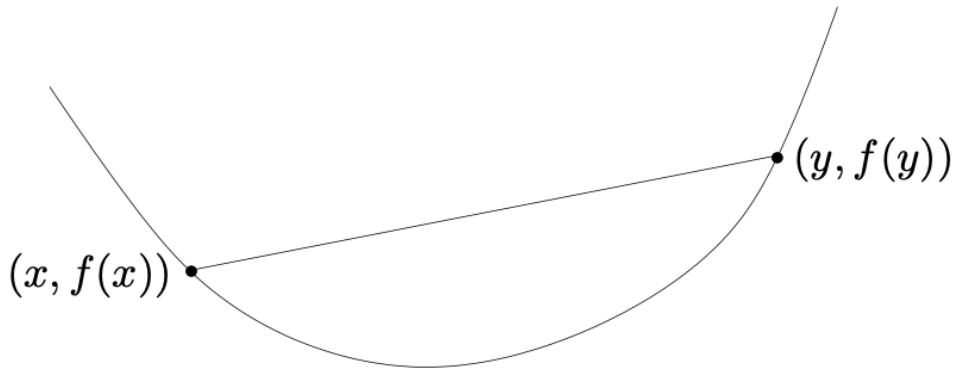
Questions?

Convex function

Convex Functions

Definition: A function $f : \mathbb{R}^d \to \mathbb{R}$ is **convex** if
(i) $\mathbf{dom}(f)$ is a convex set and
(ii) for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{dom}(f)$ and λ with $0 \leqslant \lambda \leqslant 1$, we have

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leqslant \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}).$$



Geometrically: The line segment between $(\boldsymbol{x}, f(\boldsymbol{x}))$ and $(\boldsymbol{y}, f(\boldsymbol{y}))$ lies above the graph of f.

Convex Functions

Examples of convex functions

- Linear functions: $f(\boldsymbol{x}) = \mathbf{a}^\top \boldsymbol{x}$.
- Affine functions: $f(\boldsymbol{x}) = \mathbf{a}^\top \boldsymbol{x} + b$.
- Exponential: $f(x) = e^{\alpha x}$
- Norms: Every norm on $\mathbb{R}^d$ is convex.

Convexity of a norm:

By the triangle inequality $\|\boldsymbol{x} + \boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ and homogeneity of a norm $\|a\boldsymbol{x}\| = |a|\|\boldsymbol{x}\|$ (a is a scalar):

$$\|\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\| \leqslant \|\lambda \boldsymbol{x}\| + \|(1-\lambda)\boldsymbol{y}\| = \lambda\|\boldsymbol{x}\| + (1-\lambda)\|\boldsymbol{y}\|.$$

We used the triangle inequality for the inequality and homogeneity for the equality.

Convex Functions & Sets

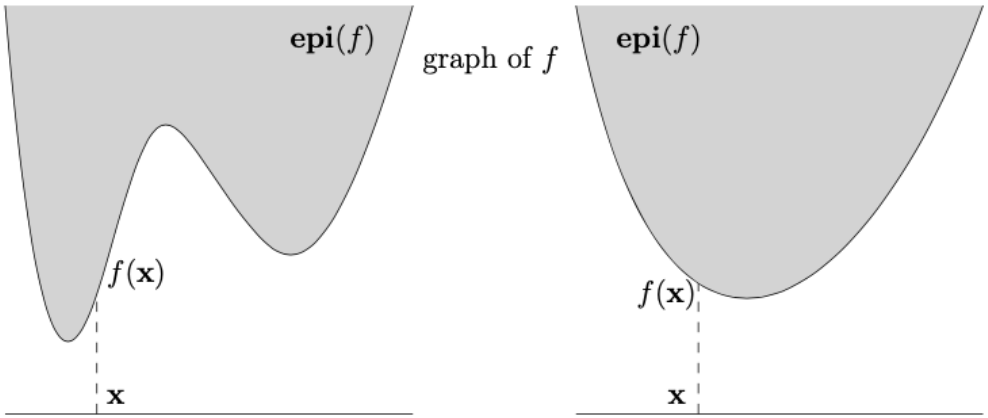
The **graph** of a function $f : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\{(\boldsymbol{x}, f(\boldsymbol{x}))|\boldsymbol{x} \in \textbf{dom}(f)\}$$

The **epigraph** of a function $f : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\textbf{epi}(f) := \{(\boldsymbol{x}, \alpha) \in \mathbb{R}^{d+1} | \textbf{x} \in \textbf{dom}(f), \alpha \geqslant f(\boldsymbol{x})\}$$

Observation 1: A function is convex if and only if its epigraph is a convex set.



Proof:

$\Rightarrow$: Suppose that f is convex, and $(\boldsymbol{x}, t)$ and $(\boldsymbol{y}, s)$ belong to **epi**(f).
To show that **epi**(f) is convex, it suffices to show that the line segment joining $(\boldsymbol{x}, t)$ and $(\boldsymbol{y}, s)$ belongs to **epi**(f), which is equivalent to

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leqslant \lambda t + (1-\lambda)s\,.$$

This can be seen easily from the convexity of f.

$\Rightarrow$: Suppose that **epi**(f) is convex.
Clearly, we have $(\boldsymbol{x}, f(\boldsymbol{x})), (\boldsymbol{y}, f(\boldsymbol{y})) \in$ **epi**(f). As **epi**(f) is convex, the line segment joining $(\boldsymbol{x}, f(\boldsymbol{x}))$ and $(\boldsymbol{y}, f(\boldsymbol{y}))$ belongs to **epi**(f), i.e.

$$(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}, \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y})) \in \mathbf{epi}(f).$$

The convexity of f follows immediately by the definition of $\mathbf{epi}(f)$.

Jensen's Inequality

Lemma 1(Jensen's Inequality)

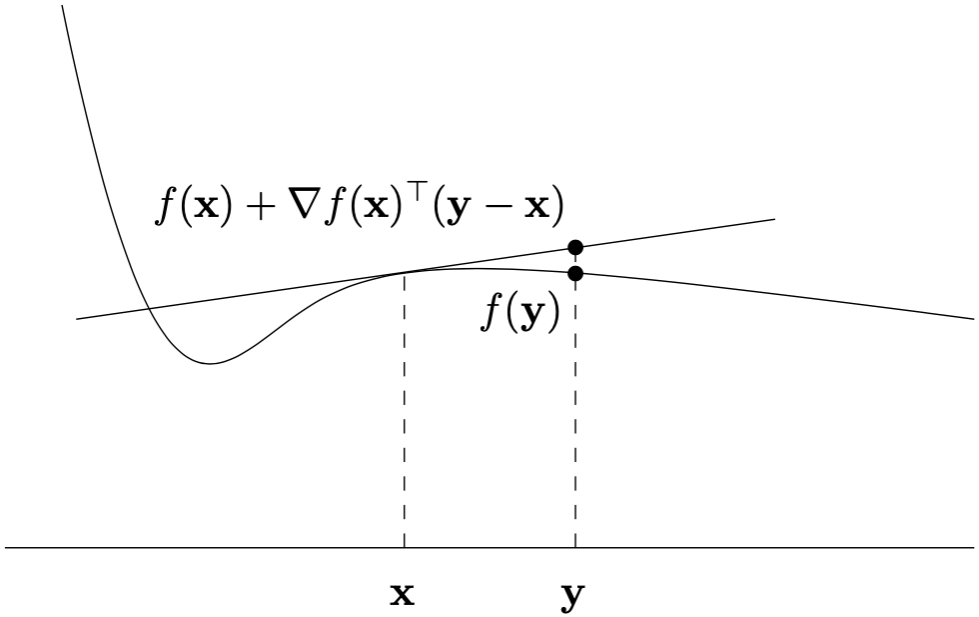
Let f be convex, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathbf{dom}(f), \lambda_1, \ldots, \lambda_m \in \mathbb{R}_+$ such that $\sum_{i=1}^m \lambda_i = 1$. Then,

$$f\Big(\sum_{i=1}^m \lambda_i \boldsymbol{x}_i\Big) \leqslant \sum_{i=1}^m \lambda_i f(\boldsymbol{x}_i)\,.$$

For $m = 2$, this is convexity. The proof of the general case is left as a homework exercise.

Differentiable Functions

Graph of the affine function $f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$ is a tangent hyperplane to the graph of f at $(\boldsymbol{x}, f(\boldsymbol{x}))$.



First-order Characterization of Convexity

Lemma 2 [BV04, 3.1.3]

Suppose $\mathbf{dom}(f)$ is open and that f is differentiable; in particular, the gradient of f (vector of partial derivatives)

$$\nabla f(\boldsymbol{x}) := \left(\frac{\partial f}{\partial x_1}(\boldsymbol{x}), \ldots, \frac{\partial f}{\partial x_d}(\boldsymbol{x}) \right)$$

exists at every point $x \in \mathbf{dom}(f)$. Then, f is convex if and only if $\mathbf{dom}(f)$ is convex and

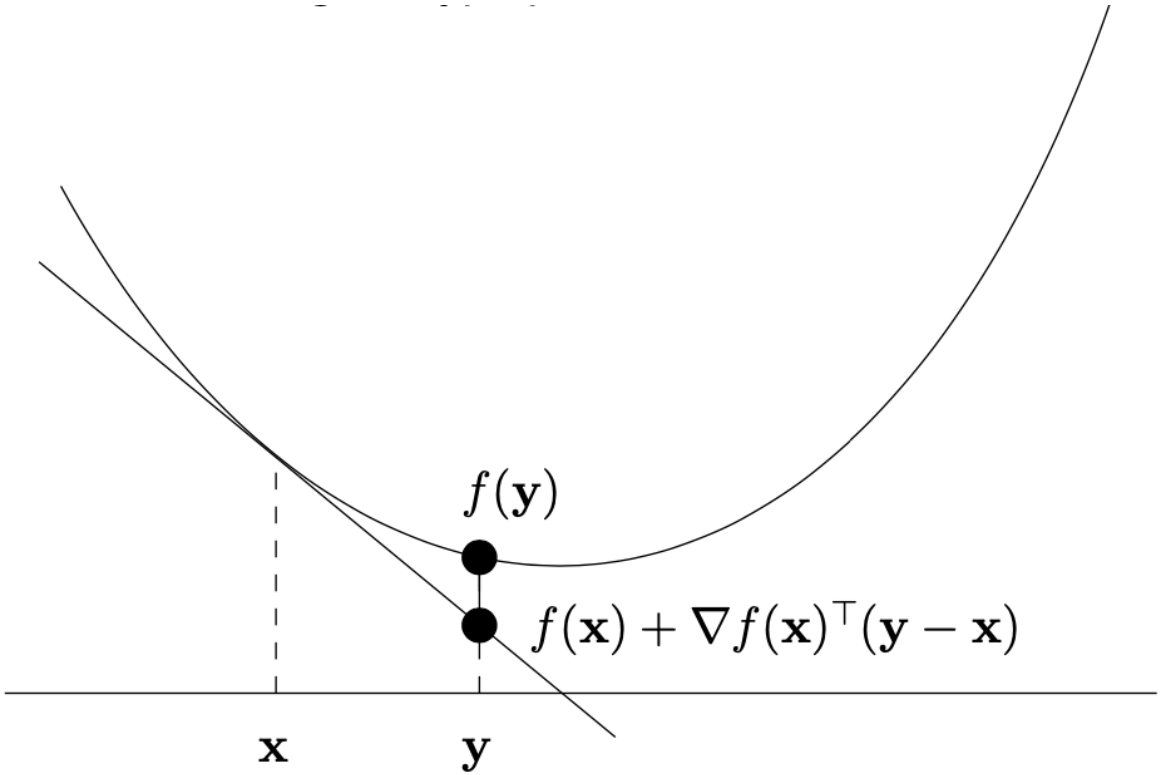
$$f(\boldsymbol{y}) \geqslant f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$$

holds for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbf{dom}(f)$.

First-order Characterization of Convexity

$$f(\boldsymbol{y}) \geqslant f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x}), \qquad \boldsymbol{x}, \boldsymbol{y} \in \mathbf{dom}(f).$$

Graph of f is above all its tangent hyperplanes.



Second-order Characterization of Convexity

Lemma 3 [BV04, 3.1.4] Suppose that $\mathbf{dom}(f)$ is open and that f is twice differentiable; in particular, the **Hessian** (matrix of second partial derivatives):

$$\nabla^2 f(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(\boldsymbol{x}) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\boldsymbol{x}) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_d}(\boldsymbol{x}) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\boldsymbol{x}) & \frac{\partial^2 f}{\partial x_2 \partial x_2}(\boldsymbol{x}) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_d}(\boldsymbol{x}) \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1}(\boldsymbol{x}) & \frac{\partial^2 f}{\partial x_d \partial x_2}(\boldsymbol{x}) & \cdots & \frac{\partial^2 f}{\partial x_d \partial x_d}(\boldsymbol{x}) \end{pmatrix}$$

exists at every point $x \in \mathbf{dom}(f)$ and is symmetric. Then f is convex if and only if $\mathbf{dom}(f)$ is convex, and for all $x \in \mathbf{dom}(f)$, we have

$$\nabla^2 f(\boldsymbol{x}) \succeq 0 \quad (\text{i.e. } \nabla^2 f \text{ is positive semidefinite}) .$$

(A symmetric matrix M is positive semidefinite if $\boldsymbol{x}^\top M \boldsymbol{x} \geqslant 0$ for all $\boldsymbol{x}$, and positive definite if $\boldsymbol{x}^\top M \boldsymbol{x} > 0$ for all $\boldsymbol{x} \neq \mathbf{0}$.)

Second-order Characterization of Convexity

Example: $f(x_1, x_2) = x_1^2 + x_2^2$.

$$\nabla^2 f(\boldsymbol{x}) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \succeq 0\,.$$

Operations that Preserve Convexity

Lemma 4 (homework exercise)

(i) Let $f_1, f_2, \ldots, f_m$ be convex functions, $\lambda_1, \lambda_2, \ldots, \lambda_m \in \mathbb{R}_+$. Then $f := \sum_{i=1}^m \lambda_i f_i$ is convex on $\mathbf{dom}(f) := \bigcap_{i=1}^m \mathbf{dom}(f_i)$.

(ii) Let f be a convex function with $\mathbf{dom}(f) \subseteq \mathbb{R}^d$, $g : \mathbb{R}^m \to \mathbb{R}^d$ an affine function, meaning that $g(\boldsymbol{x}) = A\boldsymbol{x} + \mathbf{b}$, for some matrix $A \in \mathbb{R}^{d \times m}$ and some vector $\mathbf{b} \in \mathbb{R}^d$. Then the function $f \circ g$ (that maps $\boldsymbol{x}$ to $f(A\boldsymbol{x} + \mathbf{b})$ is convex on $\mathbf{dom}(f \circ g) := \{\boldsymbol{x} \in \mathbb{R}^m : g(\boldsymbol{x}) \in \mathbf{dom}(f)\}$.

Local Minima are Global Minima

Definition

A local minimum of $f : \mathbf{dom}(f) \to \mathbb{R}$ is a point $\boldsymbol{x}$ such that there exists $\varepsilon > 0$ with

$$f(\boldsymbol{x}) \leqslant f(\boldsymbol{y}), \quad \forall \boldsymbol{y} \in \mathbf{dom}(f) \text{ satisfying } \|y - x\| \leqslant \varepsilon.$$

Lemma 5

Let $\boldsymbol{x}^*$ be a **local minimum** of a convex function $f : \mathbf{dom}(f) \to \mathbb{R}$. Then $\boldsymbol{x}^*$ is a **global minimum**, meaning that $f(\boldsymbol{x}^*) \leqslant f(\boldsymbol{y}), \forall \boldsymbol{y} \in \mathbf{dom}(f)$.

Proof.

Suppose there exists $\boldsymbol{y} \in \mathbf{dom}(f)$ such that $f(\boldsymbol{y}) < f(\boldsymbol{x}^*)$.
Define $\boldsymbol{y}' := \lambda \boldsymbol{x}^* + (1 - \lambda)\boldsymbol{y}$ for $\lambda \in (0, 1)$.
From convexity, we get that that $f(\boldsymbol{y}') < f(\boldsymbol{x}^*)$. Choosing λ close to 1 that $\|\boldsymbol{y}' - \boldsymbol{x}^*\| < \varepsilon$ yields a contradiction to $\boldsymbol{x}^*$ being a local minimum. $\square$

Critical Points are Global Minima

Lemma 6 Suppose that f is convex and differentiable over an open domain $\mathbf{dom}(f)$. Let $x \in \mathbf{dom}(f)$. If $\nabla f(\boldsymbol{x}) = \mathbf{0}$ (**critical point**), then $\boldsymbol{x}$ is a **global minimum**.

Proof.

Suppose that $\nabla f(\boldsymbol{x}) = \mathbf{0}$. According to our Lemma 2 on the first-order characterization of convexity, we have...

Geometrically, tangent hyperplane is horizontal at $\boldsymbol{x}$.

Strictly Convex Functions

Definition [BV04, 3.1.1]

A function $f : \mathbf{dom}(f) \to \mathbb{R}$ is **strictly convex** if (i) $\mathbf{dom}(f)$ is convex and (ii) for all $\boldsymbol{x} \neq \boldsymbol{y} \in \mathbf{dom}(f)$ and all $\lambda \in (0, 1)$, we have

$$f(\lambda \boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) < \lambda f(\boldsymbol{x}) + (1 - \lambda) f(\boldsymbol{y}) .$$

convex, but not strictly convex

strictly convex

Lemma 7

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be strictly convex. Then f has at most one global minimum.

Constrained Minimization

Definition

Let $f : \mathbf{dom}(f) \to \mathbb{R}$ be convex and let $X \subseteq \mathbf{dom}(f)$ be a convex set. A point $\boldsymbol{x} \in X$ is a minimizer of f over X if

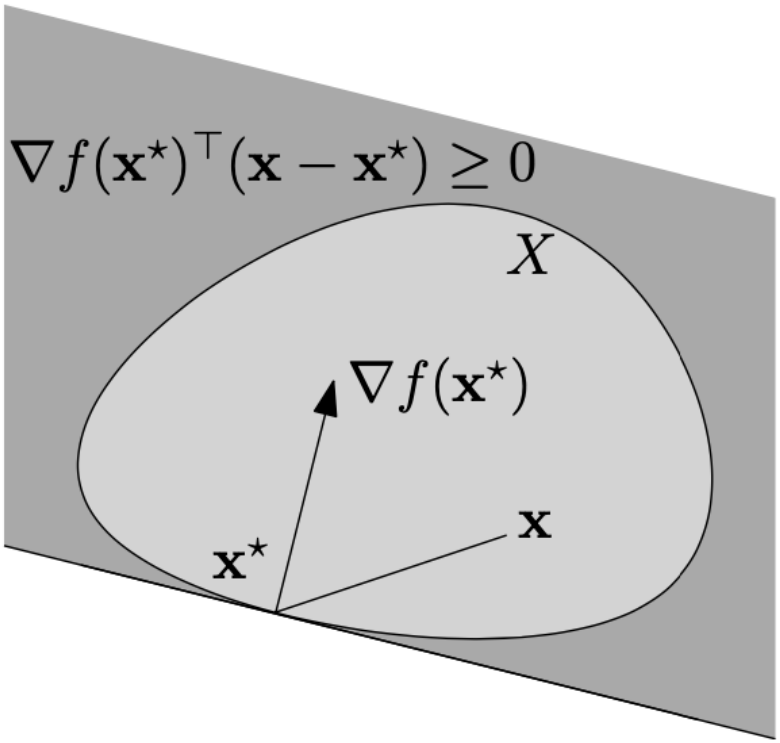
$$f(\boldsymbol{x}) \leqslant f(\boldsymbol{y}), \qquad \forall \boldsymbol{y} \in X\,.$$

Lemma 8

Suppose that $f : \mathbf{dom}(f) \to \mathbb{R}$ is convex and differentiable over an open domain $\mathbf{dom}(f) \subseteq \mathbb{R}^d$, and let $X \subseteq \mathbf{dom}(f)$ be a convex set. Point $\boldsymbol{x}^* \in X$ is a minimizer of f over X if and only if

$$\nabla f(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*) \geqslant 0, \quad \forall \boldsymbol{x} \in X.$$

Constrained Minimization



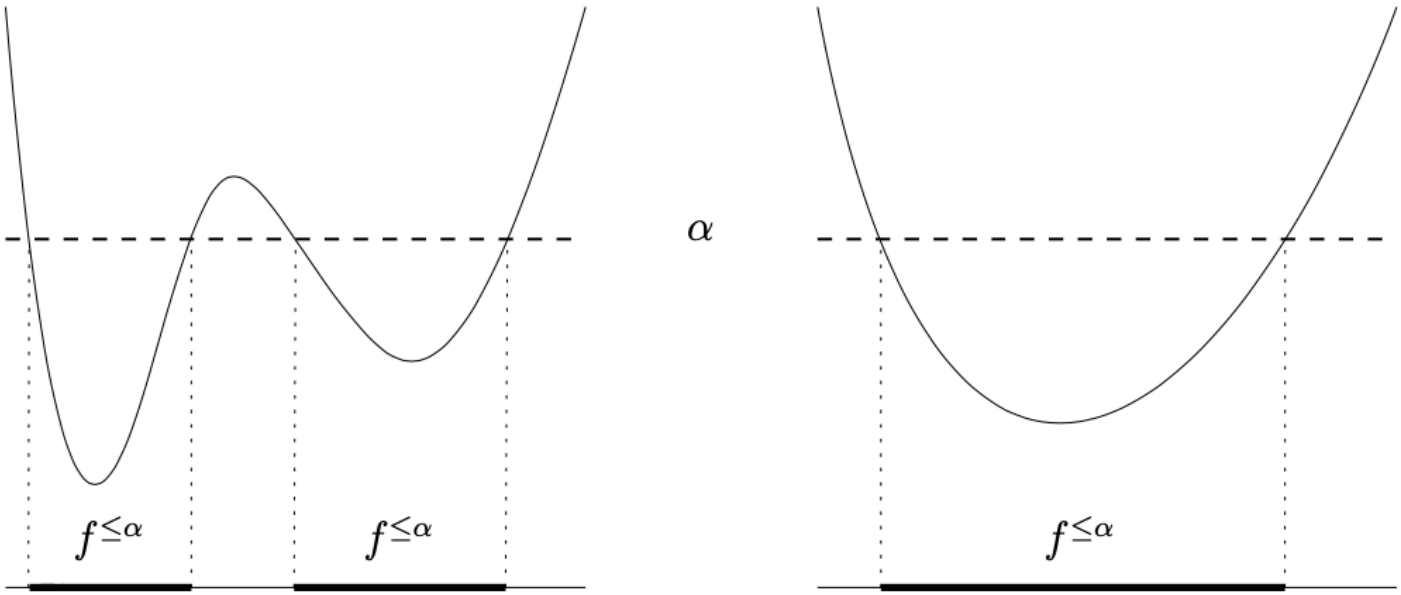
Existence of a minimizer

How do we know that a global minimum exists?

Not necessarily the case, even if f bounded from below ($f(x) = e^x$)

Definition

Let $f : \mathbb{R}^d \to \mathbb{R}, \alpha \in \mathbb{R}$. The set $f^{\leqslant \alpha} := \{\boldsymbol{x} \in \mathbb{R}^d : f(\boldsymbol{x}) \leqslant \alpha\}$ is the α-sublevel set of f.



Convex Optimization

Convex Optimization Problems are of the form

$$\min_{x \in \mathcal{D}} f(x),$$

where

- f is a convex function.
- $\mathcal{D} \subset \mathbf{dom}(f)$ is a convex set (note: $\mathbb{R}^d$ is convex)

Crucial Property of Convex Optimization Problems:

- Every local minimum is a global minimum.

Solving Convex Optimization - Provably

For convex optimization problems, the algorithms

- Coordinate Descent, Gradient Descent, Stochastic Gradient Descent, Projected and Proximal Gradient Descent,...

do **converge** to the global optimum! (assuming f is differentiable)

Example: The convergence rate for the gradient descent is proportional to $1/t$, i.e.

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*) \leqslant \frac{c}{t},$$

where $\boldsymbol{x}^*$ is some optimal solution to the problem.

Meaning: **Approximation error** converges to 0 over time.

References

- Sébastien Bubeck, *Convex optimization: Algorithms and complexity*, Foundations and Trends in Machine Learning **8** (2015), no. 3-4, 231–357.
- Stephen P Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.
- Arkadij Semenovič Nemirovskij and David Borisovich Yudin, *Problem complexity and method efficiency in optimization*, Wiley-Interscience, 1983.
- Stephen J Wright and Benjamin Recht, *Optimization for data analysis*, Cambridge University Press, 2022.