Georgia Tech College of Computing

CX4240 Spring 2026

Brief Intro to Optimization

Bo Dai
School of CSE, Georgia Tech
bodai@cc.gatech.edu

Basic / Prerequisites

- Probability
  - Distributions, densities, marginalization, conditioning
- Statistics
  - Mean, variance, maximum likelihood estimation
- Linear Algebra and Optimization
  - Vector, matrix, multiplication, inversion, eigen-value decomposition
- Coding Skills
  - Pytorch and/or JAX

Machine Learning for Apartment Hunting

- Suppose you are to move to Atlanta
- And you want to find the most reasonably priced apartment satisfying your needs:

Living area (ft^2)	# bedroom	Monthly rent ($)
230	1	900
506	2	1800
433	2	1500
190	1	800
…		
150	1	?
270	1.5	?

Linear Regression Model

- Assume y is a linear function of x (features) plus noise ϵ

$$y = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n + \epsilon$$

where ϵ is an error model as Gaussian $N(0, \sigma^2)$ Probability

- Let $\theta = (\theta_0, \theta_1, \dots, \theta_n)^\top$, and augment data by one dimension

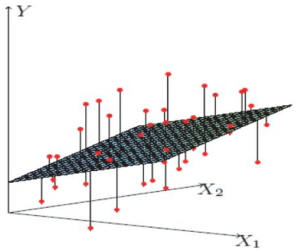
$$x \leftarrow (1, x)^\top$$

Linear algebra

Then $y = \theta^\top x + \epsilon$

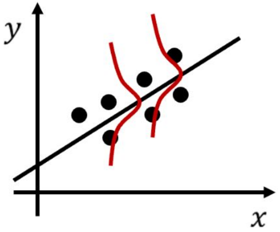
Linear algebra

Gaussian Likelihood



- Assume y is a linear in x plus noise ϵ

$$y = \theta^\top x + \epsilon$$



- Assume ϵ follows a Gaussian $N(0, \sigma)$

$$p(y^i \mid x^i; \theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^i - \theta^\top x^i)^2}{2\sigma^2}\right)$$

- By independence assumption, likelihood is

$$L(\theta) = \prod_i^m p(y^i \mid x^i; \theta) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^m \exp\left(-\frac{\sum_i^m (y^i - \theta^\top x^i)^2}{2\sigma^2}\right)$$

Probability

MLE

$$L(\theta) = \prod_{i}^{m} p\left(y^i \middle| x^i; \theta\right) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^m \exp\left(-\frac{\sum_i^m \left(y^i - \theta^\top x^i\right)^2}{2\sigma^2}\right)$$

MLE

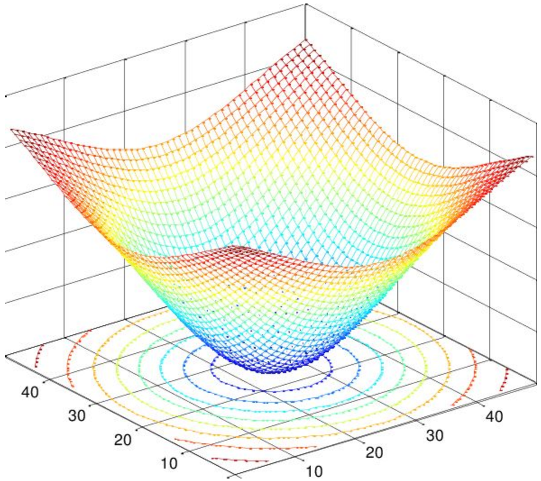
$$L(\theta) = \prod_{i}^{m} p(y^i|x^i;\theta) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^m \exp\left(-\frac{\sum_i^m (y^i - \theta^\top x^i)^2}{2\sigma^2}\right)$$

$$\max_{\theta} \; \log L(\theta) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y^i - \theta^\top x^i)^2 - m\log(\sqrt{2\pi}\sigma)$$

Optimization

Least Mean Square for Linear Regression

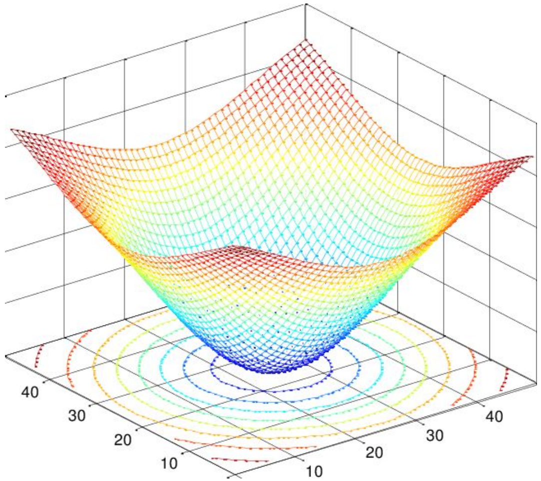
Optimization Problem



$$\text{minimize } f(\theta)$$

- $\theta \in \mathbf{R}^d$ is the variable or decision variable
- $f: \mathbf{R}^d \to \mathbf{R}$ is the objective function
- goal is to choose θ to minimize f

Optimization Problem

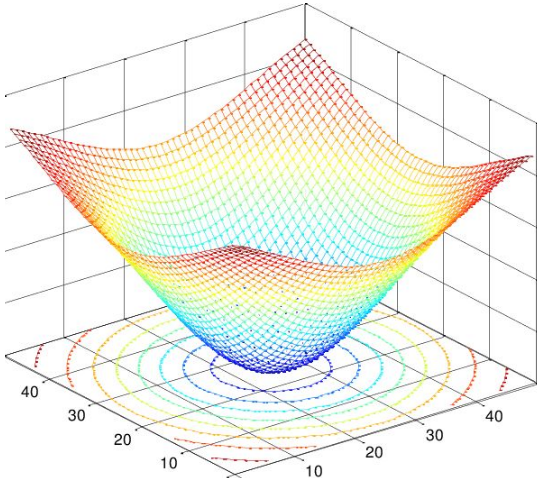


$$\text{minimize } f(\theta)$$

- $\theta \in \mathbf{R}^d$ is the variable or decision variable
- $f: \mathbf{R}^d \to \mathbf{R}$ is the objective function
- goal is to choose θ to minimize f

$$\max_{\theta} \ \log L(\theta) = -\frac{1}{2\sigma^2} \sum_{i=1}^{m} (y^i - \theta^\top x^i)^2 - m \log(\sqrt{2\pi}\sigma)$$

Optimization Problem



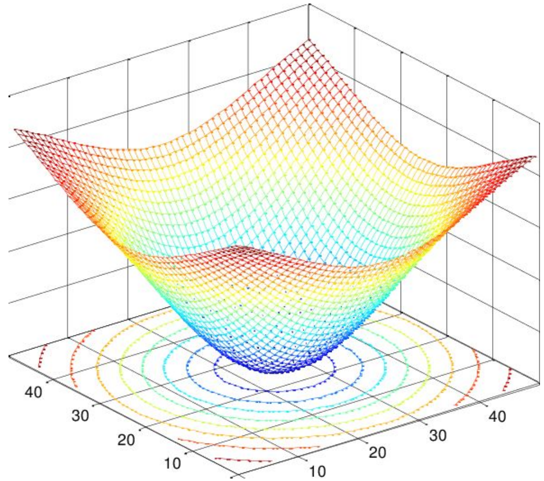
$$\text{minimize } f(\theta)$$

- $\theta \in \mathbf{R}^d$ is the variable or decision variable
- $f: \mathbf{R}^d \to \mathbf{R}$ is the objective function
- goal is to choose θ to minimize f

$$\max_{\theta} \; \log L(\theta) = -\frac{1}{2\sigma^2} \sum_{i=1}^{m} (y^i - \theta^\top x^i)^2 - m \log(\sqrt{2\pi}\sigma)$$

$$\min_{\theta} \; -\log L(\theta) = \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y^i - \theta^\top x^i)^2 + m \log(\sqrt{2\pi}\sigma)$$

Optimization Problem

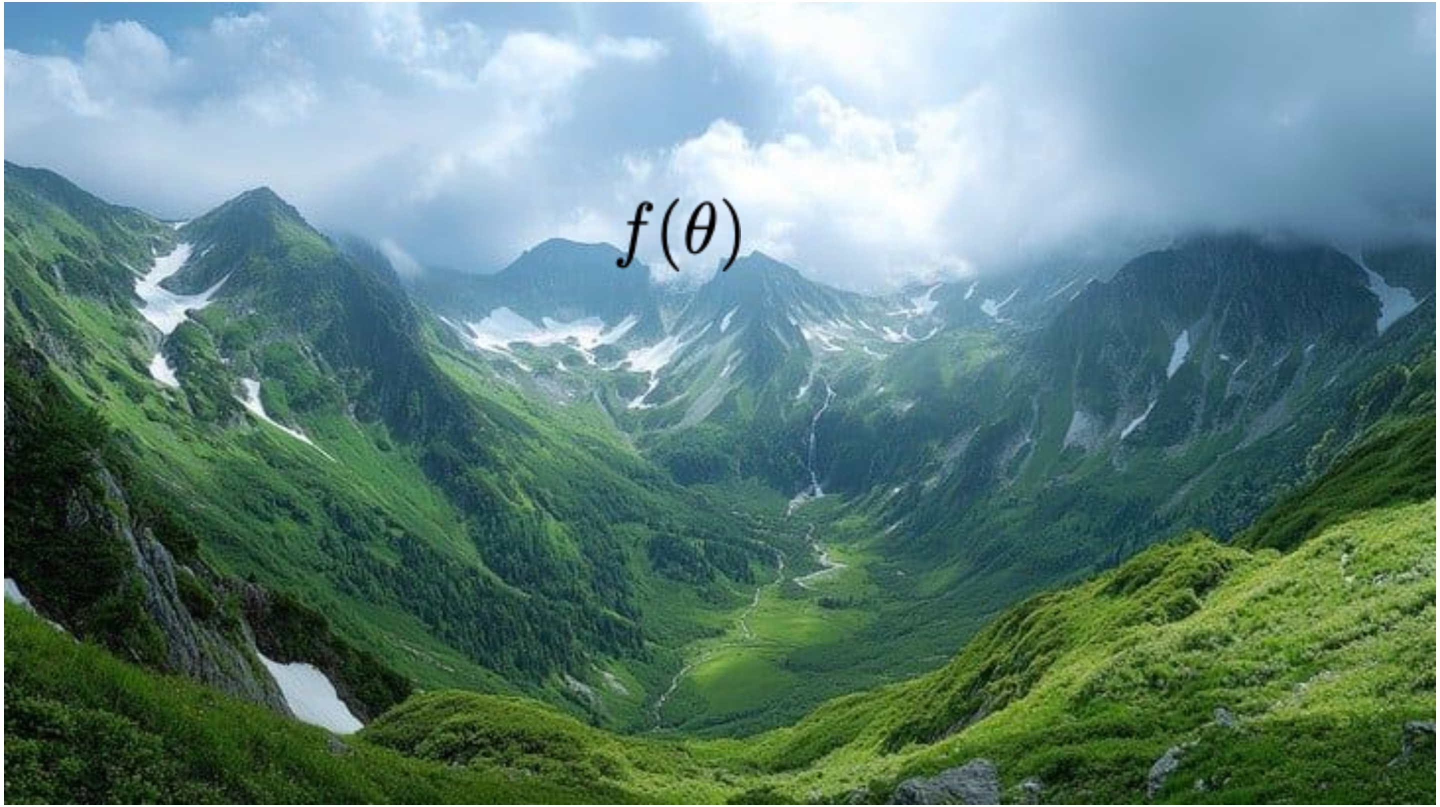


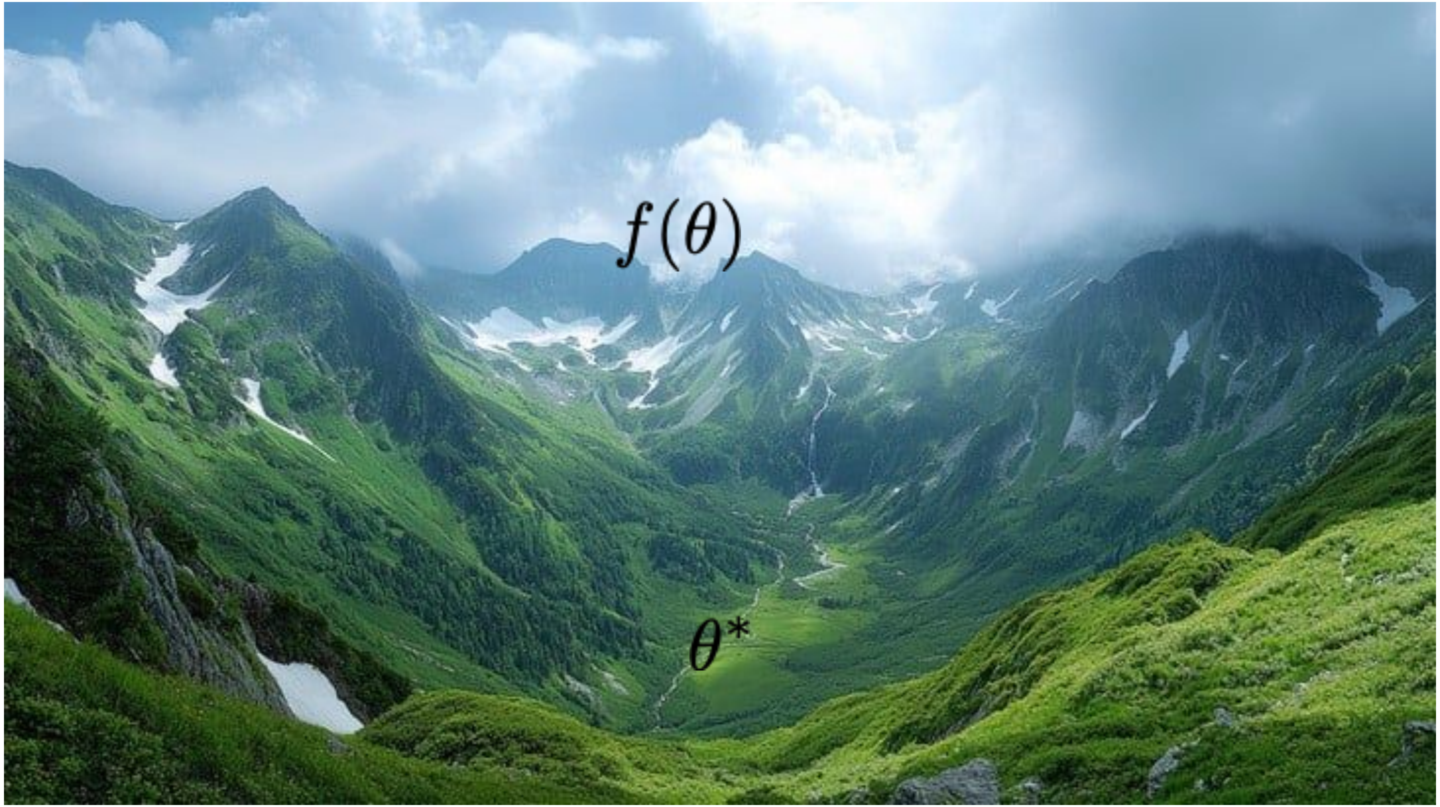
$$\text{minimize } f(\theta)$$

- $\theta \in \mathbf{R}^d$ is the variable or decision variable
- $f: \mathbf{R}^d \to \mathbf{R}$ is the objective function
- goal is to choose θ to minimize f
- θ^* is optimal means that for all $\theta, f(\theta) \geq f(\theta^\star)$
- $f^\star = f(\theta^\star)$ is the optimal value of the problem

$$\theta^* = \arg\min_\theta \; f(\theta)$$

$$f^* = \min_\theta \; f(\theta)$$


$f(\theta)$


$f(\theta)$
θ^*

Random Search ?!

Random Search ?!

Random Search ?!

Random Search ?!

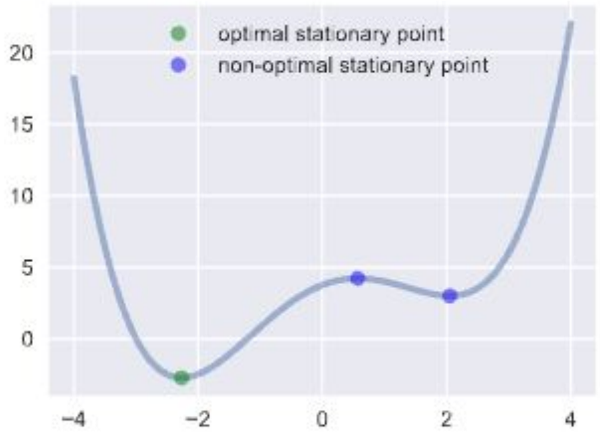
Random Search ?!

Random Search ?!

Random Search ?!

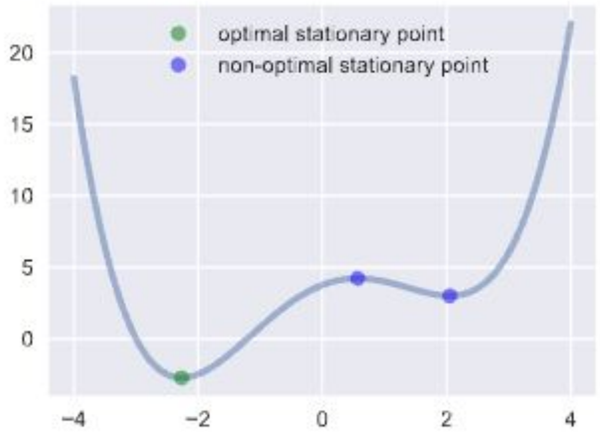
Random Search ?!

Optimality Condition



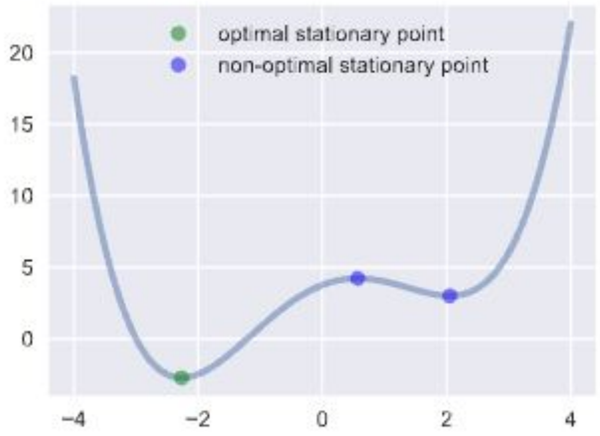
- let's assume that f is differentiable, i.e., partial derivatives $\frac{\partial f(\theta)}{\partial \theta_i}$ exist
- if $\theta^\star$ is optimal, then $\nabla f(\theta^\star) = 0$
- $\nabla f(\theta) = 0$ is called the optimality condition for the problem

Optimality Condition



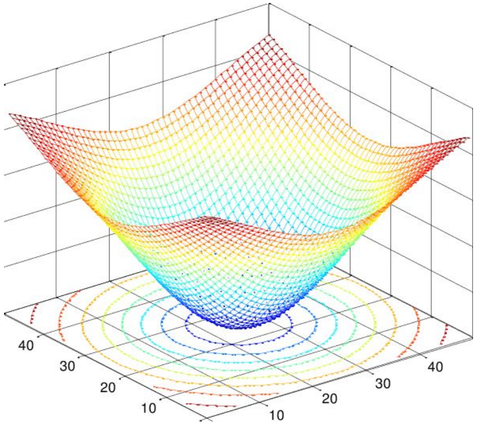
- let's assume that f is differentiable, i.e., partial derivatives $\frac{\partial f(\theta)}{\partial \theta_i}$ exist
- if $\theta^\star$ is optimal, then $\nabla f(\theta^\star) = 0$ Necessary Conditions
- $\nabla f(\theta) = 0$ is called the optimality condition for the problem
- there can be points that satisfy $\nabla f(\theta) = 0$ but are not optimal
- we call points that satisfy $\nabla f(\theta) = 0$ stationary points
- not all stationary points are optimal

Optimality Condition



- let's assume that f is differentiable, i.e., partial derivatives $\frac{\partial f(\theta)}{\partial \theta_i}$ exist
- if $\theta^\star$ is optimal, then $\nabla f(\theta^\star) = 0$
- $\nabla f(\theta) = 0$ is called the optimality condition for the problem
- there can be points that satisfy $\nabla f(\theta) = 0$ but are not optimal
- we call points that satisfy $\nabla f(\theta) = 0$ stationary points
- not all stationary points are optimal

Optimality Condition



- let's assume that f is differentiable, i.e., partial derivatives $\frac{\partial f(\theta)}{\partial \theta_i}$ exist
- if $\theta^\star$ is optimal, then $\nabla f(\theta^\star) = \mathbf{0}$
- $\nabla f(\theta) = \mathbf{0}$ is called the optimality condition for the problem
- there can be points that satisfy $\nabla f(\theta) = \mathbf{0}$ but are not optimal
- we call points that satisfy $\nabla f(\theta) = \mathbf{0}$ stationary points
- not all stationary points are optimal

Solving Optimization Problems

- in some cases, we can solve the problem analytically

Solving Optimization Problems

- in some cases, we can solve the problem analytically
- e.g., least squares: minimize $f(\theta) = \|X\theta - y\|_2^2$

Solving Optimization Problems

- in some cases, we can solve the problem analytically
- e.g., least squares: minimize $f(\theta) = \|X\theta - y\|_2^2$
- optimality condition is $\nabla f(\theta) = 2X^\top(X\theta - y) = 0$

Solving Optimization Problems

- in some cases, we can solve the problem analytically
- e.g., least squares: minimize $f(\theta) = \|X\theta - y\|_2^2$
- optimality condition is $\nabla f(\theta) = 2X^\top(X\theta - y) = 0$
  this has unique solution $\theta^\star = (X^\top X)^1 X^\top y = X^\dagger y$ (when columns of X are linearly independent)

Solving Optimization Problems

- in some cases, we can solve the problem analytically
- e.g., least squares: minimize $f(\theta) = \|X\theta - y\|_2^2$
- optimality condition is $\nabla f(\theta) = 2X^\top(X\theta - y) = 0$
  this has unique solution $\theta^\star = (X^\top X)^1 X^\top y = X^\dagger y$ (when columns of X are linearly independent)

What if optimality condition is difficult to be solved?

Local Search

Local Search

Local Search

Local Search

Local Search

Local Search

Local Search

Local Search

Local Search

Iterative Algorithm

- iterative algorithm computes a sequence $\theta^1, \theta^2, \ldots$
- θ^k is called the k th iterate
- θ^1 is called the starting point

$$f(\theta^{k+1}) < f(\theta^k), k = 1,2, \ldots$$

i.e., each iterate is better than the previous one

Iterative Algorithm

- iterative algorithm computes a sequence $\theta^1, \theta^2, \ldots$
- θ^k is called the k th iterate
- θ^1 is called the starting point

$$f(\theta^{k+1}) < f(\theta^k), k = 1,2, \ldots$$

i.e., each iterate is better than the previous one

- this means that $f(\theta^k)$ converges, but not necessarily to $f^\star$

Local Search

Local Search

Local Search

Local Search

Local Search

Iterative Algorithm

- iterative algorithm computes a sequence $\theta^1, \theta^2, \ldots$
- θ^k is called the k th iterate
- θ^1 is called the starting point

$$f(\theta^{k+1}) < f(\theta^k), k = 1,2, \ldots$$

i.e., each iterate is better than the previous one

- this means that $f(\theta^k)$ converges, but not necessarily to $f^\star$

Local Search

Gradient Descent

Gradient Descent

Gradient Descent

Iterative Algorithm

- iterative algorithm computes a sequence $\theta^1, \theta^2, \dots$
- θ^k is called the k th iterate
- θ^1 is called the starting point
- many iterative algorithms are descent methods, which means

$$f(\theta^{k+1}) < f(\theta^k), k = 1,2, \dots$$

i.e., each iterate is better than the previous one

- this means that $f(\theta^k)$ converges, but not necessarily to $f^\star$

Iterative Algorithm

- iterative algorithm computes a sequence $\theta^1, \theta^2, \ldots$
- θ^k is called the k th iterate
- θ^1 is called the starting point
- many iterative algorithms are descent methods, which means

$$f(\theta^{k+1}) < f(\theta^k), k = 1,2, \ldots$$

i.e., each iterate is better than the previous one

- this means that $f(\theta^k)$ converges, but not necessarily to $f^\star$

Gradient Descent

Gradient Descent

Gradient Method Summary

choose an initial $\theta^1 \in \mathbf{R}^d$ and $h^1 > 0$ (e.g., $\theta^1 = 0, h^1 = 1$)
for $k = 1,2, \dots, k^{\text{max}}$

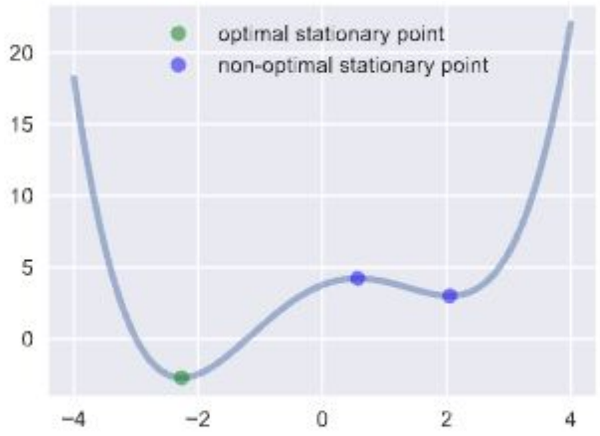
1. compute $\nabla f(\theta^k)$; quit if $\|\nabla f(\theta^k)\|_2$ is small enough
2. form tentative update $\theta^{\text{tent}} = \theta^k - h^k \nabla f(\theta^k)$
3. if $f(\theta^{\text{tent}}) < f(\theta^k)$, set $\theta^{k+1} = \theta^{\text{tent}}, h^{k+1} = 1.2h^k$
4. else set $h^k := 0.5h^k$ and go to step 2

Gradient Method Summary

choose an initial $\theta^1 \in \mathbf{R}^d$ and $h^1 > 0$ (e.g., $\theta^1 = 0, h^1 = 1$)
for $k = 1,2, \dots, k^{\max}$

1. compute $\nabla f(\theta^k)$; quit if $\|\nabla f(\theta^k)\|_2$ is small enough
2. form tentative update $\theta^{\text{tent}} = \theta^k - h^k \nabla f(\theta^k)$
3. if $f(\theta^{\text{tent}}) < f(\theta^k)$, set $\theta^{k+1} = \theta^{\text{tent}}, h^{k+1} = 1.2h^k$
4. else set $h^k := 0.5h^k$ and go to step 2

Optimality Condition



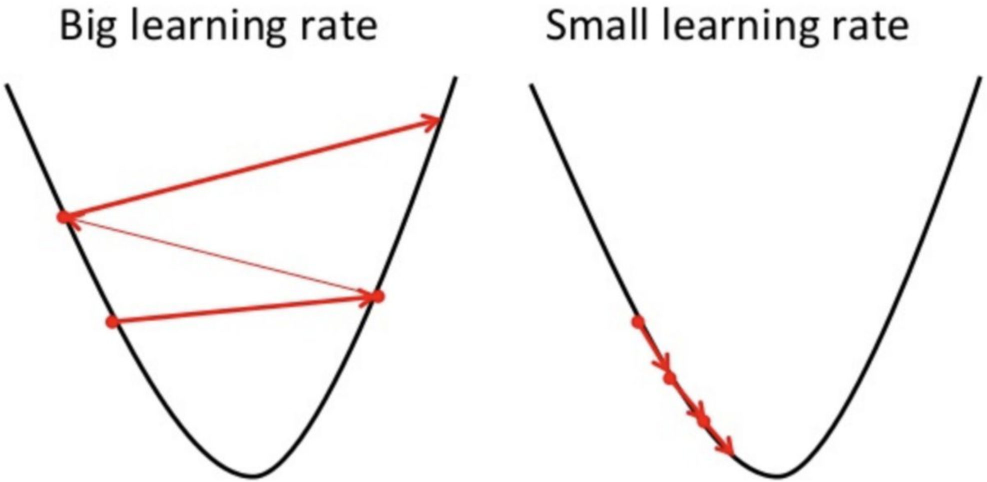
- let's assume that f is differentiable, i.e., partial derivatives $\frac{\partial f(\theta)}{\partial \theta_i}$ exist
- if $\theta^\star$ is optimal, then $\nabla f(\theta^\star) = 0$ Necessary Conditions
- $\nabla f(\theta) = 0$ is called the optimality condition for the problem
- there can be points that satisfy $\nabla f(\theta) = 0$ but are not optimal
- we call points that satisfy $\nabla f(\theta) = 0$ stationary points
- not all stationary points are optimal

Gradient Method Summary

choose an initial $\theta^1 \in \mathbf{R}^d$ and $h^1 > 0$ (e.g., $\theta^1 = 0, h^1 = 1$)
for $k = 1,2,\dots,k^{\max}$

1. compute $\nabla f(\theta^k)$; quit if $\left\|\nabla f(\theta^k)\right\|_2$ is small enough
2. form tentative update $\theta^{\text{tent}} = \theta^k - h^k \nabla f(\theta^k)$
3. if $f(\theta^{\text{tent}}) < f(\theta^k)$, set $\theta^{k+1} = \theta^{\text{tent}}, h^{k+1} = 1.2h^k$
4. else set $h^k := 0.5h^k$ and go to step 2

Step-size Matters



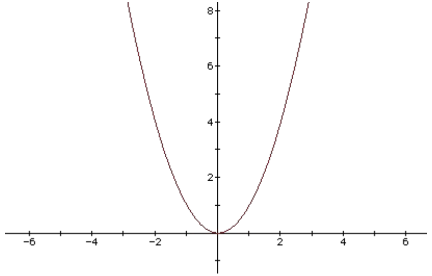
Stopping Criterion

- in practice, we stop after a finite number K of steps
- typical stopping criterion: stop if $\left\|\nabla f(\theta^k)\right\|_2 \leq \epsilon$ or $k = k^{\max}$
- ϵ is a small positive number, the stopping tolerance
- $k^{\max}$ is the maximum number of iterations

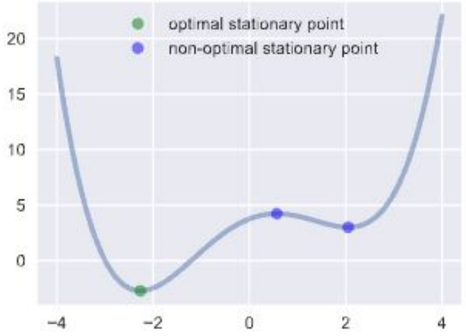
Gradient Method Convergence

- (assuming some technical conditions hold) we have

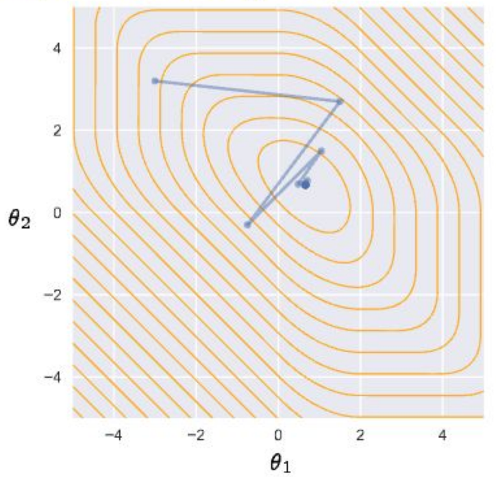
$$\left\|\nabla f\left(\theta^k\right)\right\|_2 \to 0 \text{ as } k \to \infty$$

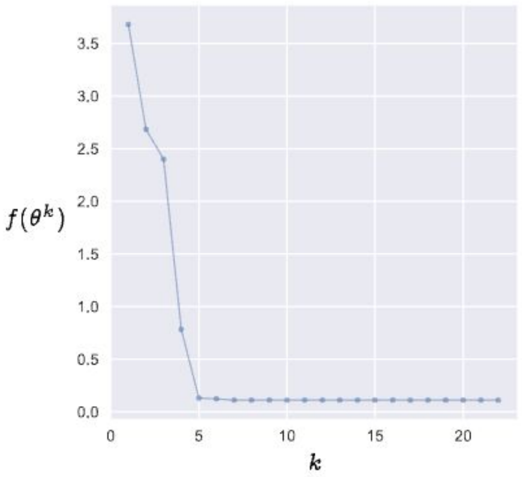


- i.e., the gradient method always finds a stationary point
- for convex problems
  - ➢ gradient method is non-heuristic
  - ➢ for any starting point $\theta^1, f\left(\theta^k\right) \to f^\star$ as $k \to \infty$
- for non-convex problems
  - ➢ gradient method is heuristic
  - ➢ we can (and often do) have $f\left(\theta^k\right) \nrightarrow f^\star$

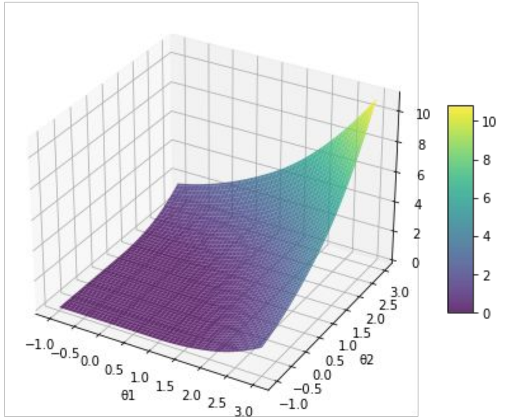


Example: Convex Objective

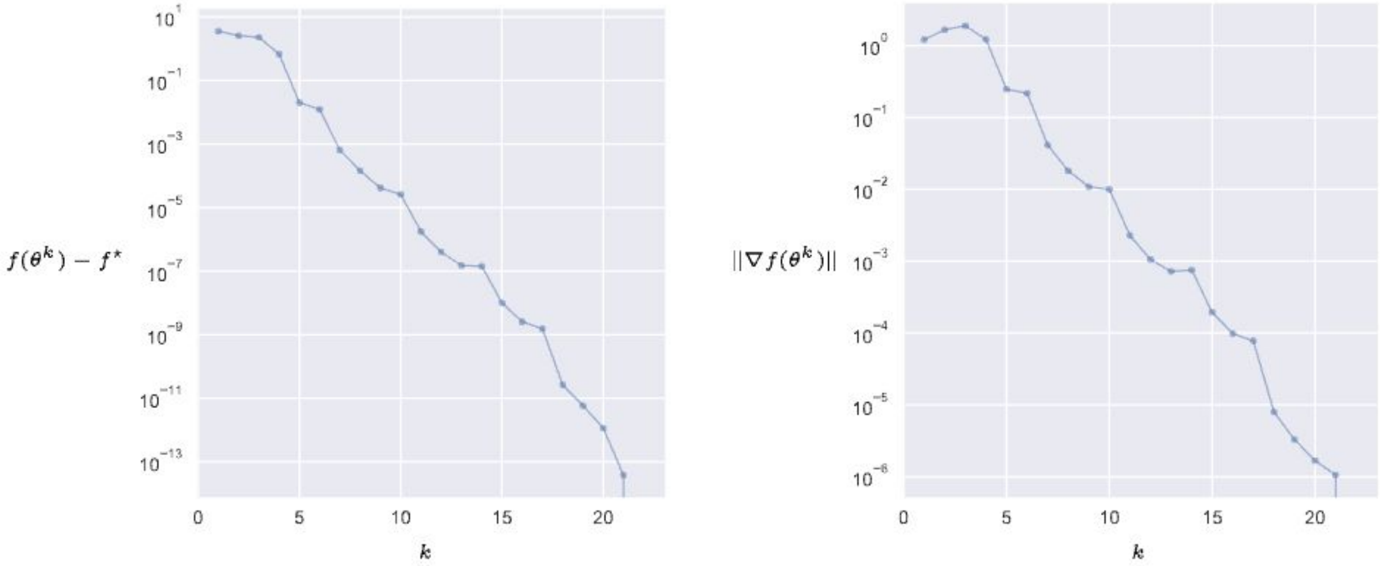




- $f(\theta) = \frac{1}{3}\left(p^{\text{hub}}(\theta_1 - 1) + p^{\text{hub}}(\theta_2 - 1) + p^{\text{hub}}(\theta_1 + \theta_2 - 1)\right)$
- f is convex
- optimal point is $\theta^\star = (2/3, 2/3)$, with $f^\star = 1/9$

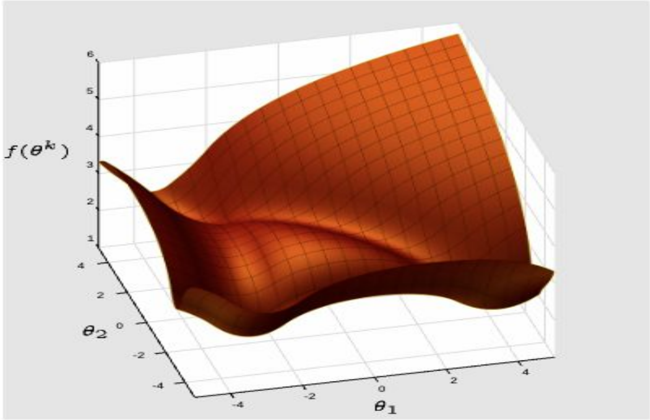


Example: Convex Objective



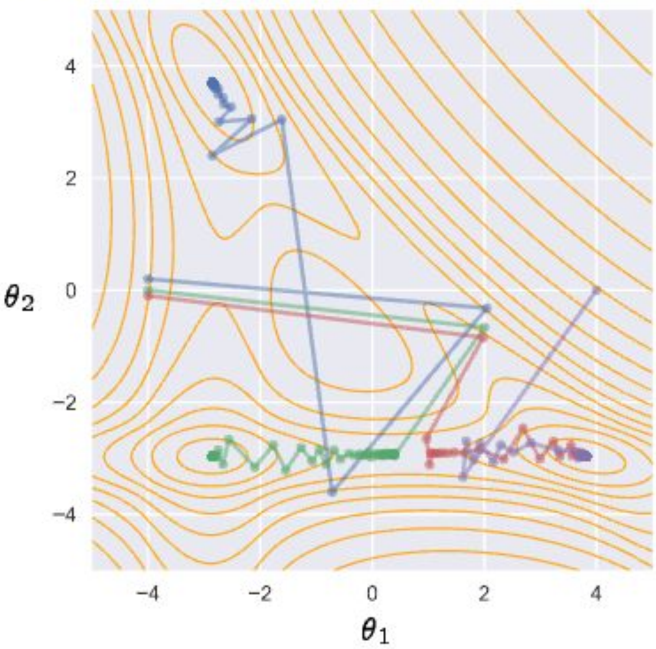
- $f(\theta^k)$ is a decreasing function of k, (roughly) exponentially
- $\|\nabla f(\theta^k)\| \to 0$ as $k \to \infty$

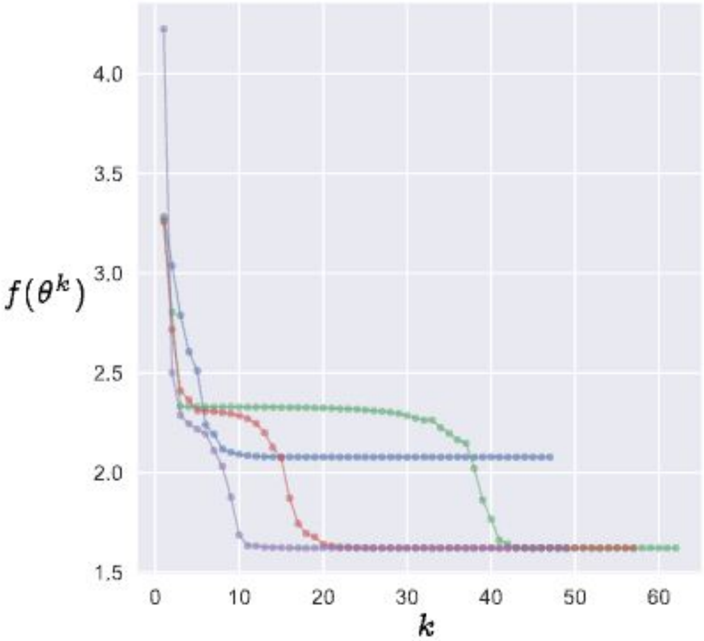
Example: Non-Convex Objective



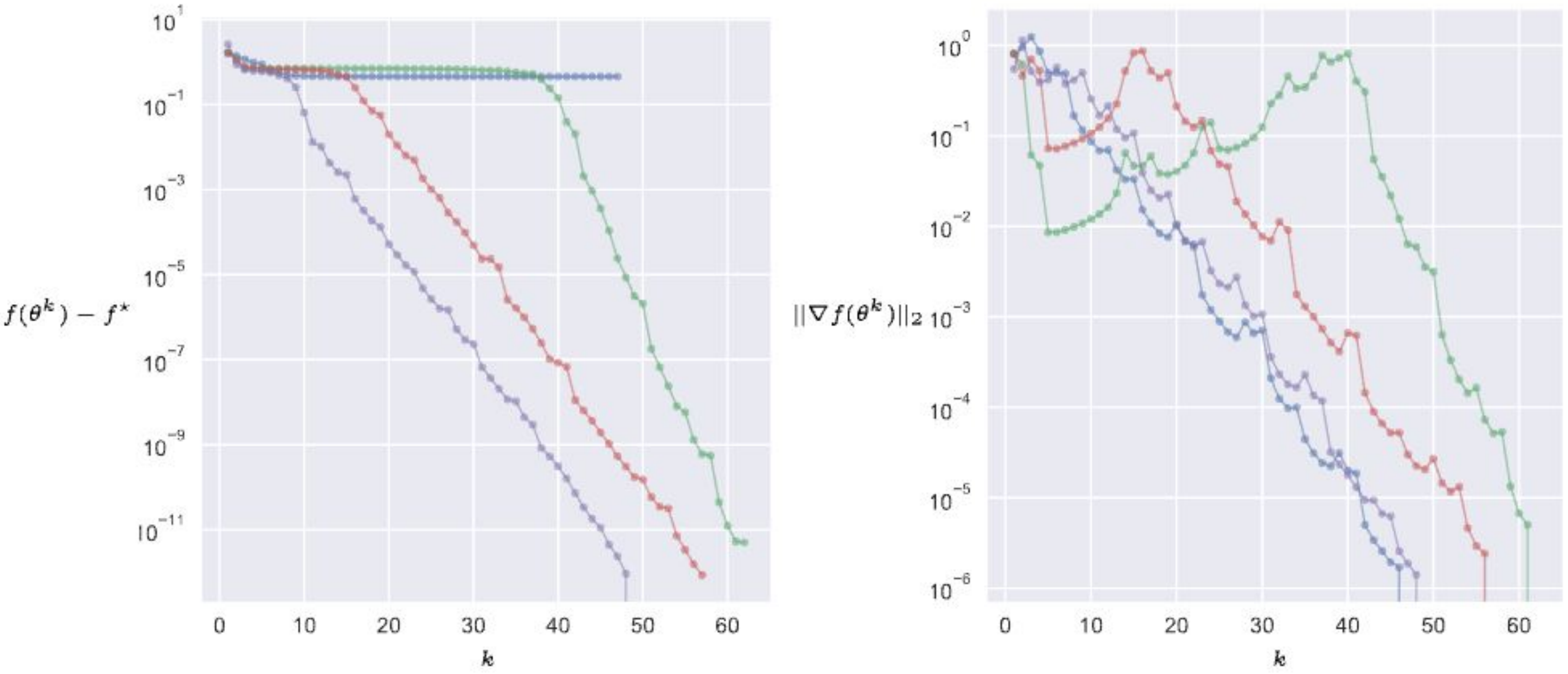
- gradient algorithm converges, but limit depends on initial guess

Example: Non-Convex Objective





Example: Non-Convex Objective



Stochastic Gradient Descent

Goal: minimize $f(\theta) = \frac{1}{n}\sum_{i=1} f(x_i, y_i; \theta)$

Initialize $\theta^0 \in \mathbb{R}^d$ randomly
Iterate until convergence:

- $\nabla f(\theta)|_{\theta=\theta^t} = \frac{1}{n}\sum_{i=1}^{n} \nabla f(x_i, y_i, \theta)|_{\theta=\theta^t}$
- $\theta^{t+1} = \theta^t - \eta \nabla f(\theta)|_{\theta=\theta^t}$

Stochastic Gradient Descent

Goal: minimize $f(\theta) = \frac{1}{n}\sum_{i=1} f(x_i, y_i; \theta)$

Initialize $\theta^0 \in \mathbb{R}^d$ randomly
Iterate until convergence:

- Randomly sample a point (x_i, y_i) from the n data points
- Compute noisy gradient $\tilde{g}^t = \nabla f(x_i, y_i; \theta)|_{\theta=\theta^t}$
- Update $\theta^{t+1} = \theta^t - \eta\tilde{g}^t$

Intuition of why Stochastic GD can work

Claim: the random noisy gradient is an unbiased estimate of the true gradient

Note the point (x_i, y_i) is uniformly random sampled from n data points, we have:

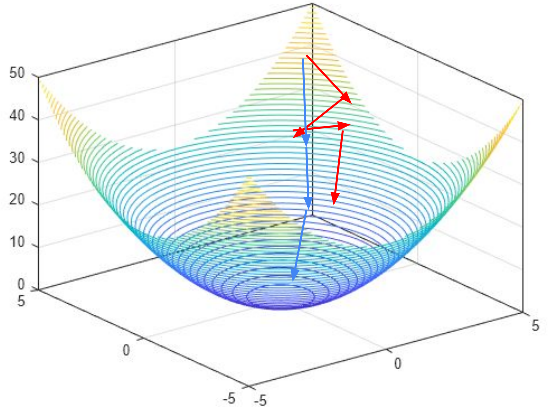
$$\mathbb{E}[\nabla f(x_i, y_i; \theta)]$$

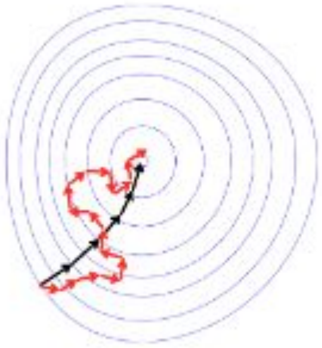
$$= \frac{1}{n}\sum_{i=1}^{n} \nabla f(x_i, y_i; \theta) = \nabla \left[\frac{1}{n}\sum_{i=1}^{n} f(x_i, y_i; \theta)\right] = \nabla f(\theta)$$

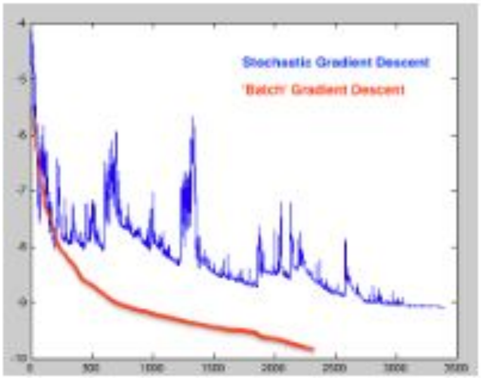
Stochastic gradient descent generally makes more iterations than gradient descent.

Each iteration is much cheaper (by a factor of n).

$$\vec{\nabla} f(\vec{\theta}) = \vec{\nabla} \sum_{j=1}^{n} f_j(\vec{\theta}) \text{ vs. } \vec{\nabla} f_j(\vec{\theta})$$







Apply GD and SGD to LMS

$$\max_{\theta} \; \log L(\theta) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(y^i - \theta^\top x^i)^2 - m\log(\sqrt{2\pi}\sigma)$$

- The gradient of LMS is

$$\frac{1}{m}\nabla_\theta \log L(\theta) = \frac{1}{m\sigma^2}\sum_{i=1}^{m}(y^i - \theta^\top x^i)\, x^i$$

- The stochastic gradient of LMS is

$$\nabla_\theta \log L(\theta)\Big|_{\text{sample } i} = \frac{1}{\sigma^2}\left(y^i - \theta^\top x^i\right) x^i$$

Summary

- Random Search
- Closed-form
- Iterative methods:
  - Local Search
  - Gradient Descent
  - Stochastic Gradient Descent

- Homework 1 is released
- Due: 11:59PM EST, 02/04/2026

Q&A