Optimization
(18-660/18-460)

Instructor: Guannan Qu

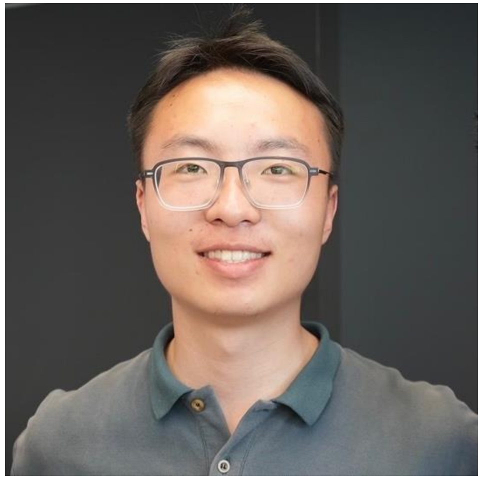
gqu@andrew.cmu.edu

Course Setup

- Instructor: Guannan Qu, Assit. Prof. at ECE
- Email: gqu@andrew.cmu.edu
- Lecture: Tuesdays and Thursdays, 11 AM -12:20 PM (will be recorded)
- Instructor Office Hour: Thursdays 2PM – 3PM, hybrid (zoom + in person at Porter B22)

TA Team

Alex Deweese
mdeweese@andrew.cmu.edu
Thursday 9:30AM - 10:30AM



Chaoyi Pan
chaoyip@andrew.cmu.edu
Monday 9AM - 10AM

Vidur Sinha
vidursin@andrew.cmu.edu
Tuesday 4PM – 5PM

Stephen Oduh
soduh@andrew.cmu.edu
Wednesday 11AM - 12PM

Mark Phillip Matovic
markphim@andrew.cmu.edu
Friday 11AM - 12PM

Check canvas for OH zoom link and locations

Optional Recitation Sessions

- Purpose
  - Help cover some extra materials
  - Review for tests
- ~5 in the semester
  - week 2: Review of linear algebra and calculus
  - week 5: CVX (for homework 2)
  - week 6: Review for test 1
  - week 9: Review for homework 3
  - week 11: Review for test 2
- Time: Poll online for time
- Zoom only (will be recorded)

Lectures

- Will use slides (with annotation/writing)
- Slides will be made available on website on the day of lecture
- Course will be recorded

Reading Materials

- No official textbook

- A few good resources
  - Boyd and Vanderberghe, *Convex Optimization* (free online!)
  - Nesterov, *Introductory Lectures on Convex Optimization*
  - Dimitri P. Bertsekas, *Nonlinear Programming*
  - Bubeck, *Convex Optimization: Algorithms and Complexity* (free online!)

- I will post reading materials (optional) for each lecture

Prerequisite

- Linear algebra, calculus, basic real analysis
- Programming (MATLAB or Python)
- Ability to do mathematical proofs

- First review session will cover linear algebra and calculus

Evaluation

- Homework: 30%
- Tests: 40%
- Project: 20%
- Quizzes: 10%

Homework (30%)

- 4 HWs, each 2 weeks
- 5 late days in total, and 3 late days max for single HW
- If there is an emergency, contact me ASAP before the due date.

Homework (30%)

- **Collaboration policy**: You may discuss homework problems with other students at a high level (e.g., strategy, intuition). You must clearly acknowledge all collaborators and describe the nature of the collaboration. All derivations, calculations, and written solutions must be produced independently. You may not look at or copy another student's solutions.
- **GenAI policy**: You may use GenAI tools (e.g., ChatGPT) as a learning aid, e.g., to clarify lecture material, review relevant theory, or ask for conceptual hints. You MAY NOT use GenAI to generate, check, or complete solutions to any homework problem, subproblem, or derivation.
- **Other online resources**: You may consult online resources covering general concepts from the course. You may not search for or consult solutions or worked examples that directly correspond to any homework problem or close variant.
- **Reporting requirements**: In each homework submission, you will be asked to report (1) the collaborators you have in solving the homework and the nature of the collaboration (2) a summary of how you used GenAI in completing the homework, or a statement that you have not used GenAI in completing the homework. You can *optionally* attach your chat history with GenAI (3) citations to any other online resources you have used in solving the homework.

The goal of this policy is to help you learn optimization deeply and independently, while allowing you to use modern GenAI tools responsibly as a learning aid, not a shortcut.

Tests: 40%

- 2 Tests, on Feb 26 and April 7
- In-class test (11 AM – 12:20 pm)
- Open-book, open-notes (you can use all the materials within the course, including the lecture slides, homework, reading materials)

- To be environmentally friendly and prevent excessive printing, you may use your electronic device (e.g. laptop, iPad) to view lecture materials during the test, but you are not allowed to access any other resources during the test (e.g. search online or use GenAI).

Evaluation

- Project: 20%
  - Form groups of 2-3 students
  - Topic: Paper reading and reproducibility study; applications of optimization; independent research
  - Presentation at the end of the course
  - Final report
  - Evaluation method: each project will be randomly assigned a few anonymous peer students for review
  - More details coming later…

Evaluation

- Quizzes: 10%
  - Small conceptual questions, just to check understanding
  - Out after every lecture, due before next lecture.
  - No quiz today. The first quiz will be released at the end of the next lecture.

Difference between 18-660 and 18-460

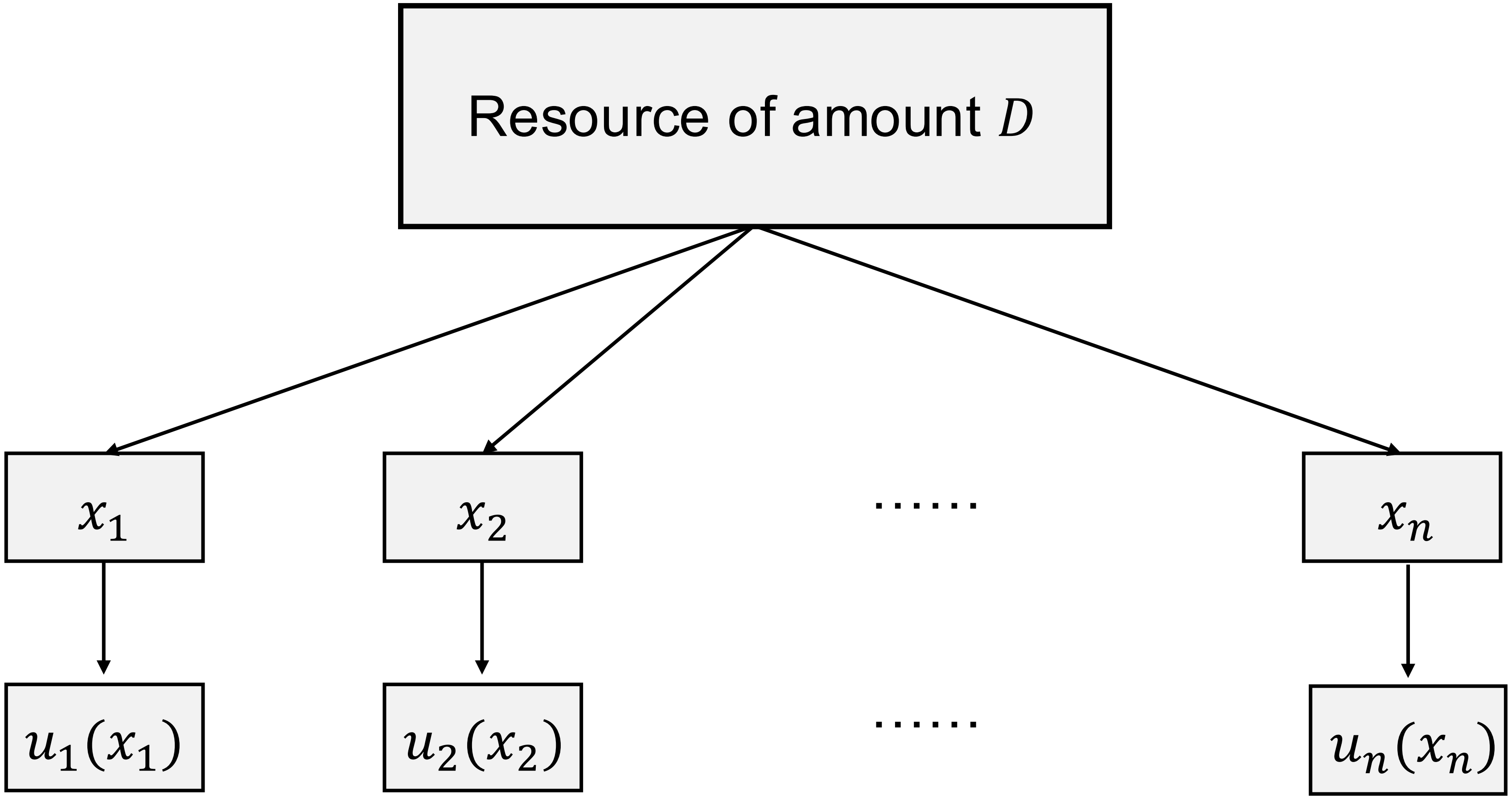
18-460 has:

- Fewer homework problems
- Fewer exam questions

What is optimization?

Making some decisions to perform “well” to some metric.

Example: Resource Allocation

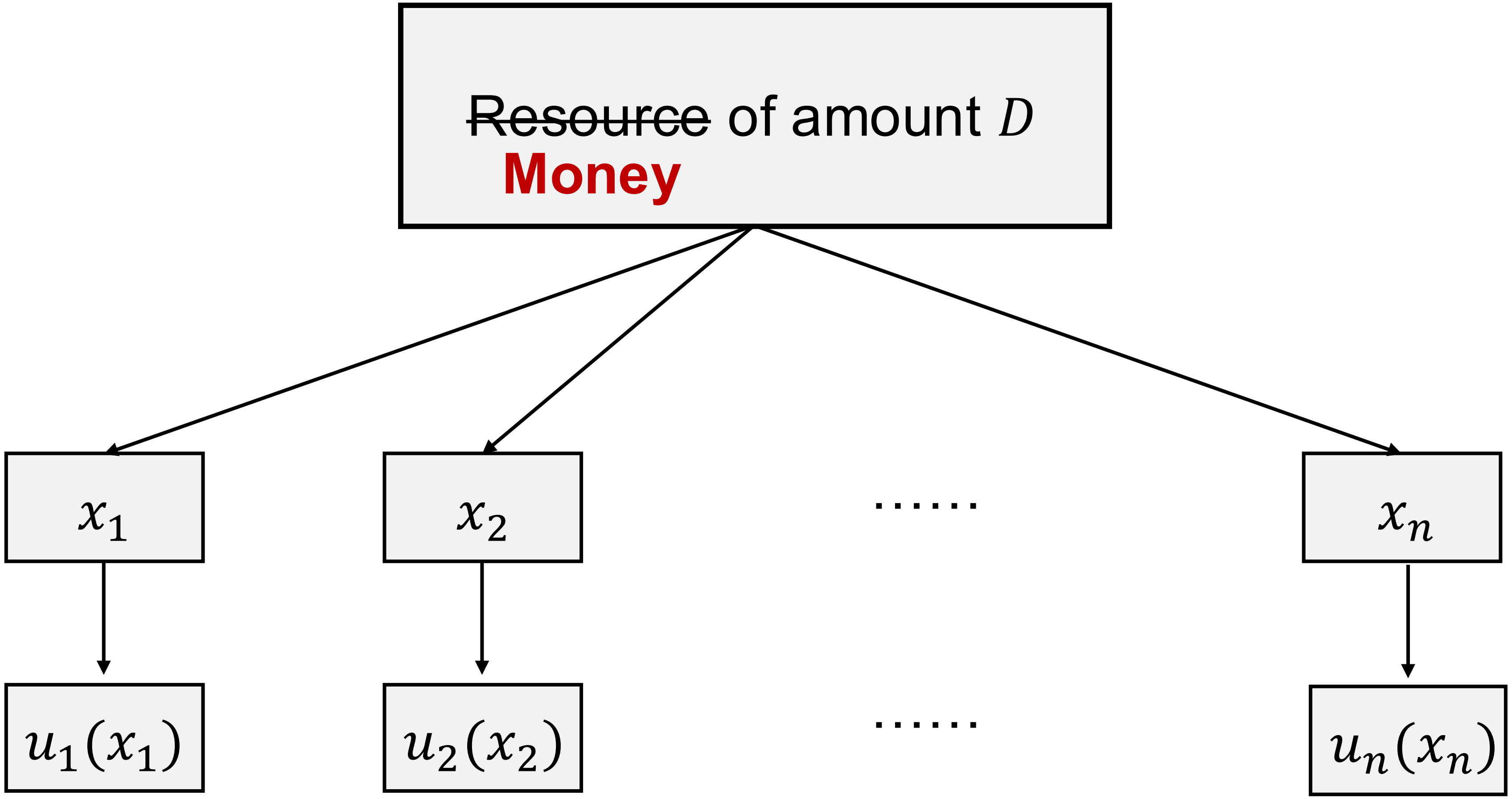


Divide into n pieces

Each piece generates utility

Goal: find an allocation $x_1, \dots, x_n$ to maximize total utility $u_1(x_1) + u_2(x_2) + \cdots + u_n(x_n)$

Example: Resource Allocation



Divide into n pieces
to invest in stocks

Each piece
generates ~~utility~~
return

Goal: find an allocation $x_1, \ldots, x_n$ to maximize total ~~utility~~ $u_1(x_1) + u_2(x_2) + \cdots + u_n(x_n)$
return

Example: Resource Allocation

Decision Variable: $x_1, \dots, x_n \in \mathbb{R}$

Objective Function: $g(x_1, \dots, x_n) = u_1(x_1) + u_2(x_2) + \cdots + u_n(x_n)$

Constraint:

$x_i \geq 0, \forall i = 1,2, \dots, n$ "cannot allocate negative resource"

$x_1 + x_2 + \cdots + x_n \leq D$ "total resource cannot exceed D"

Example: Resource Allocation

$$\max_{x_1,\dots,x_n} g(x_1,\dots,x_n) = u_1(x_1) + u_2(x_2) + \cdots + u_n(x_n)$$

s.t.

$$x_i \geq 0, \forall i = 1,2,\dots,n$$

$$x_1 + x_2 + \cdots + x_n \leq D$$

Example: Resource Allocation

$$\max_x g(x)$$

s.t. $x \in C$

Constraint represented as a set

Equivalent!

$$\min_x f(x) = -g(x)$$

s.t. $x \in C$

In this course we will take this “min” convention

Three components of optimization

Decision variable

$$\min_{x} f(x)$$

$$\text{s.t.} \quad x \in C$$

Objective/cost function

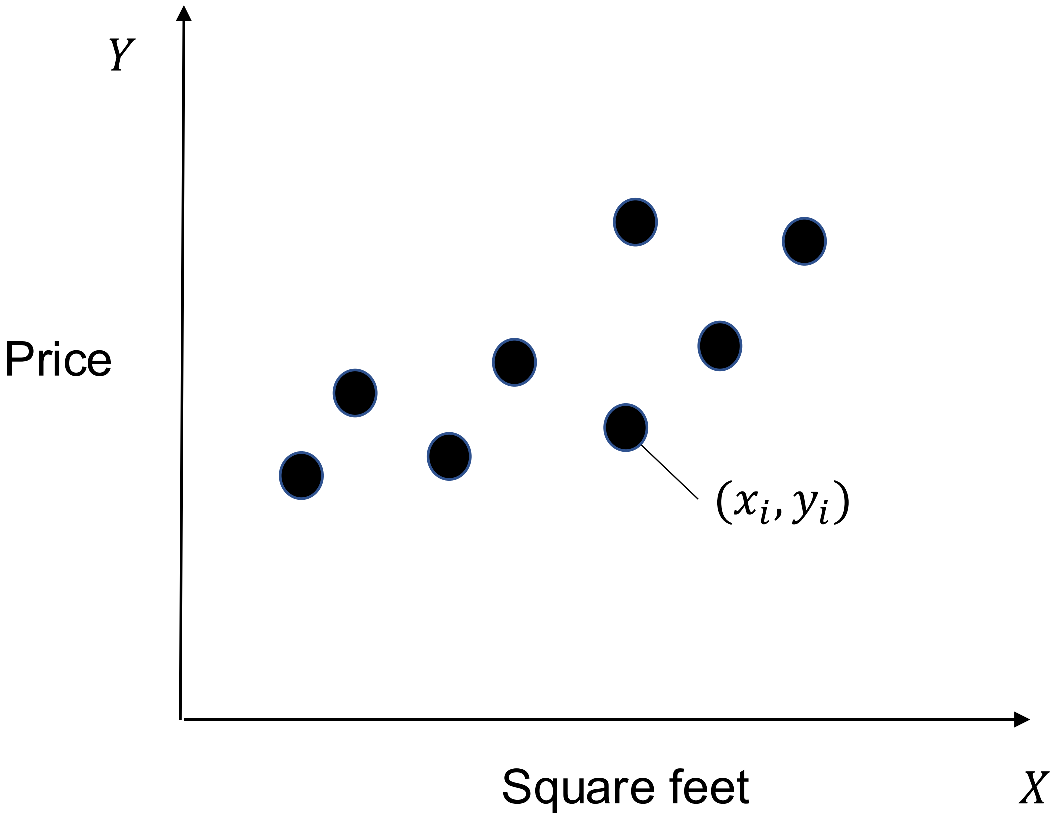
Constraint (set)

Optimization powers many real-world engineering systems

1. Optimization in machine learning: least squares
2. Optimization in signal processing: imaging denoising
3. Optimization in energy systems

Linear Regression

Suppose we have the data for the square footage and price of houses sold in Pittsburgh within a certain time window.

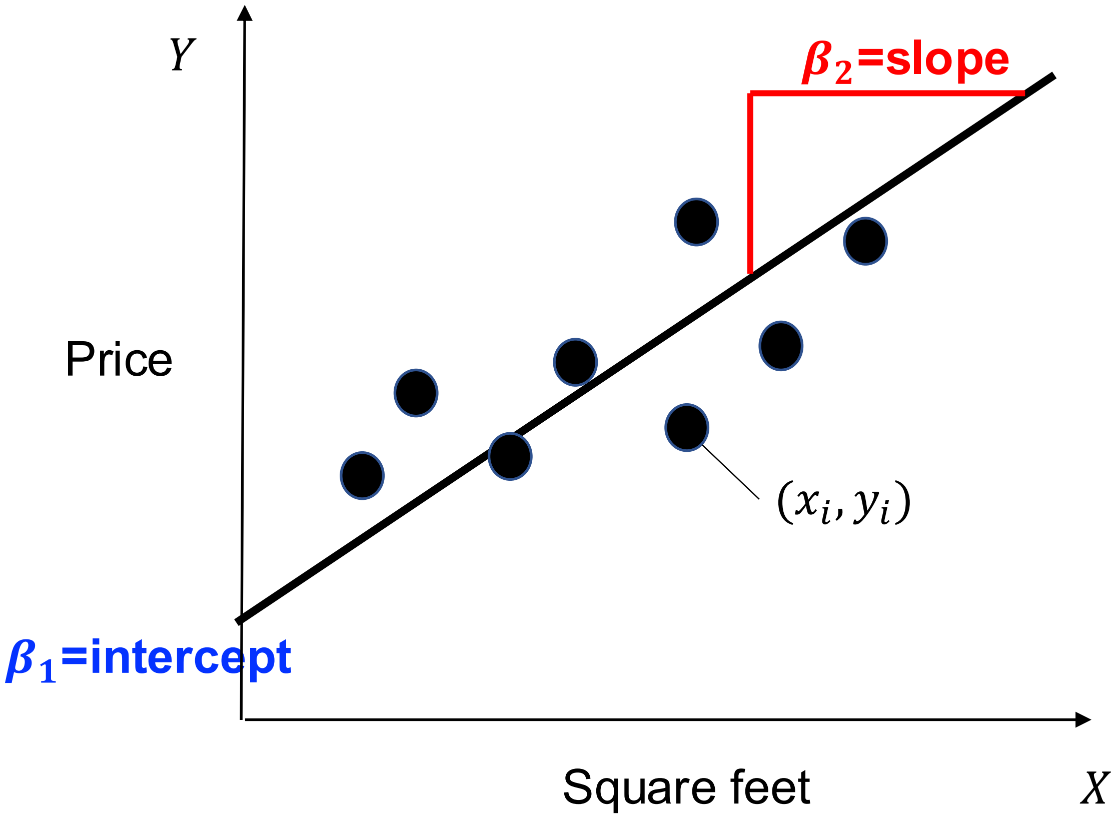


Linear Regression

Linear regression: find a line $y = \beta_1 + \beta_2 x$ that best describes the data

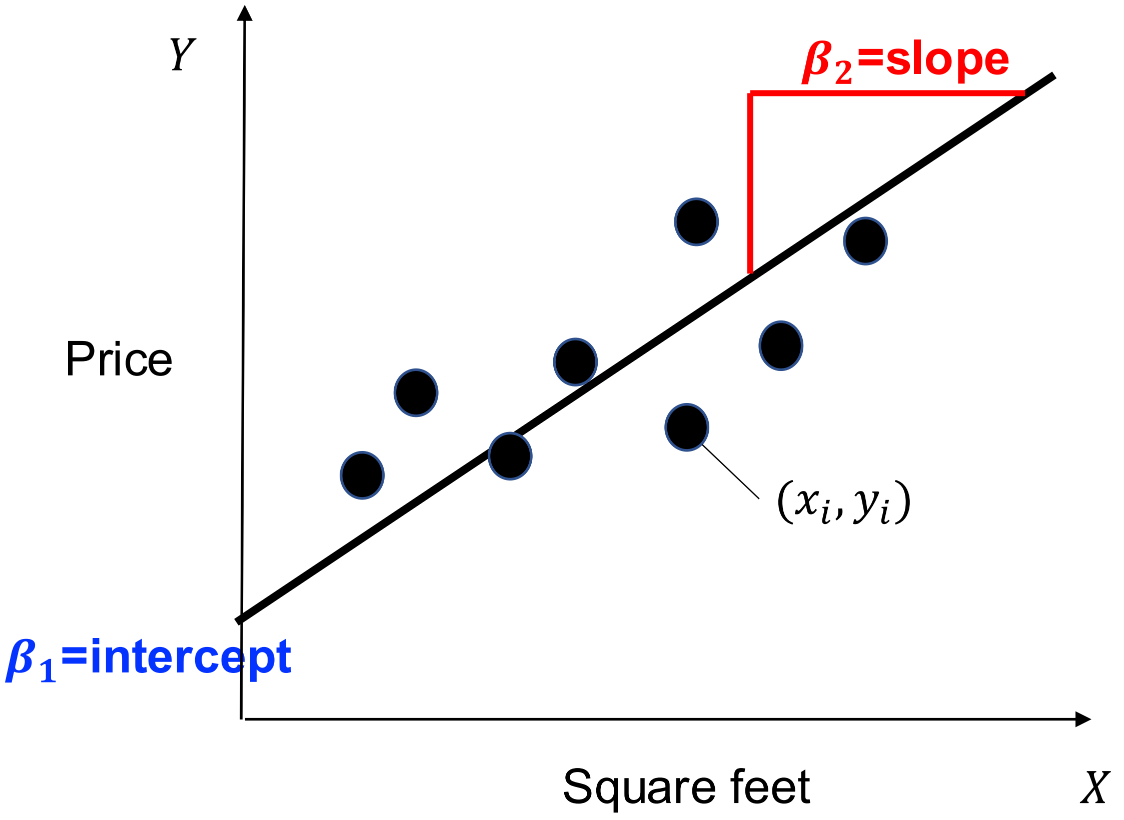
Decision Variable: $\beta_1, \beta_2 \in \mathbb{R}$

Objective: $\sum_{i=1}^{n}[y_i - (\beta_1 + \beta_2 x_i)]^2$



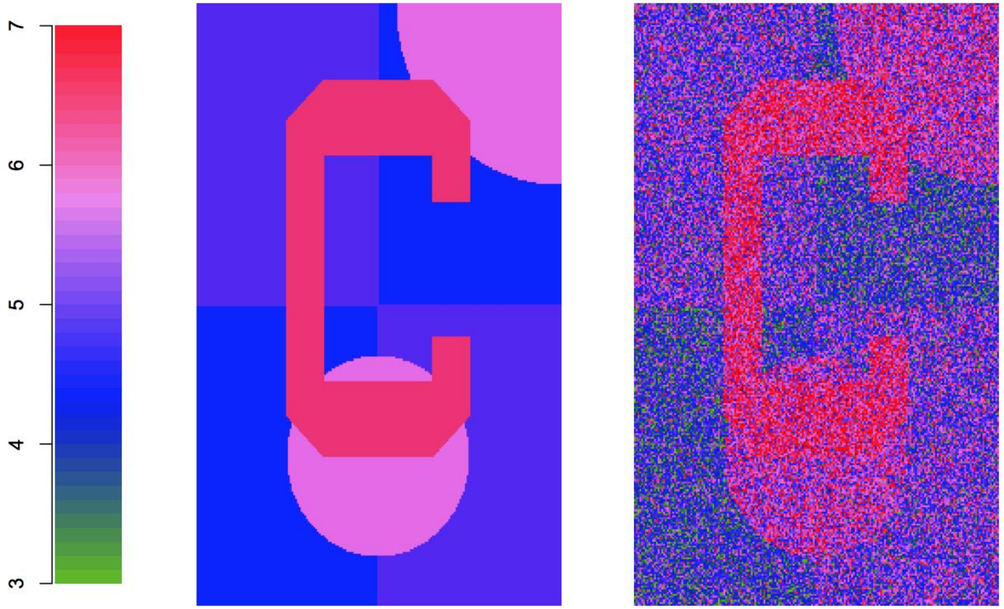
Linear Regression

$$\min_{\beta_1,\beta_2} \sum_{i=1}^{n} (y_i - \beta_1 - \beta_2 x_i)^2$$



Pretty much all ML models are trained by solving an optimization problem
e.g. logistics regression, SVM, neural networks, etc.

Denoising

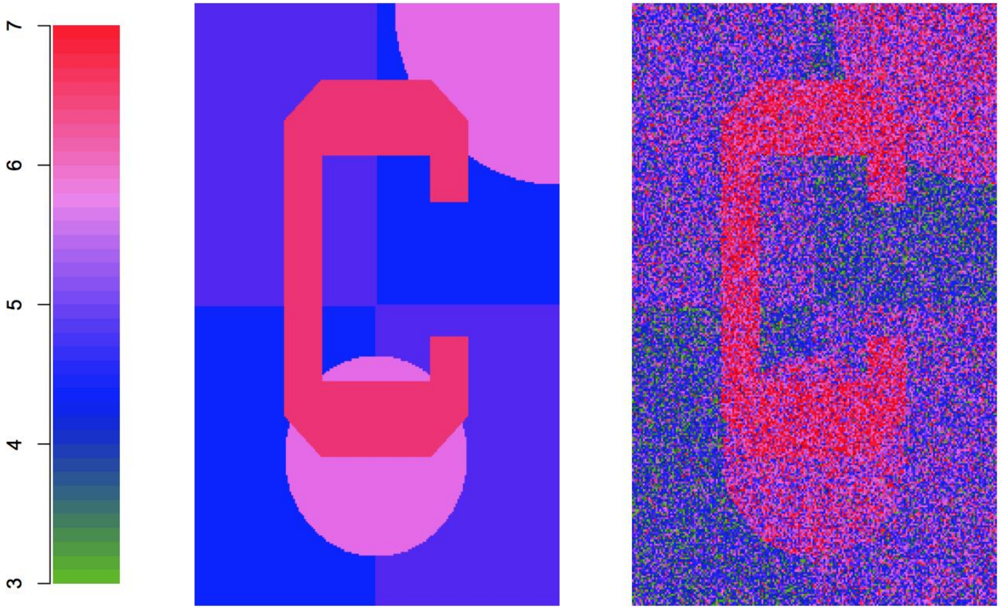


True image

Data (Represented by y_i, $i = 1, \ldots, n$, n pixels)

Goal: how to recover the true image from the noisy data?

Denoising

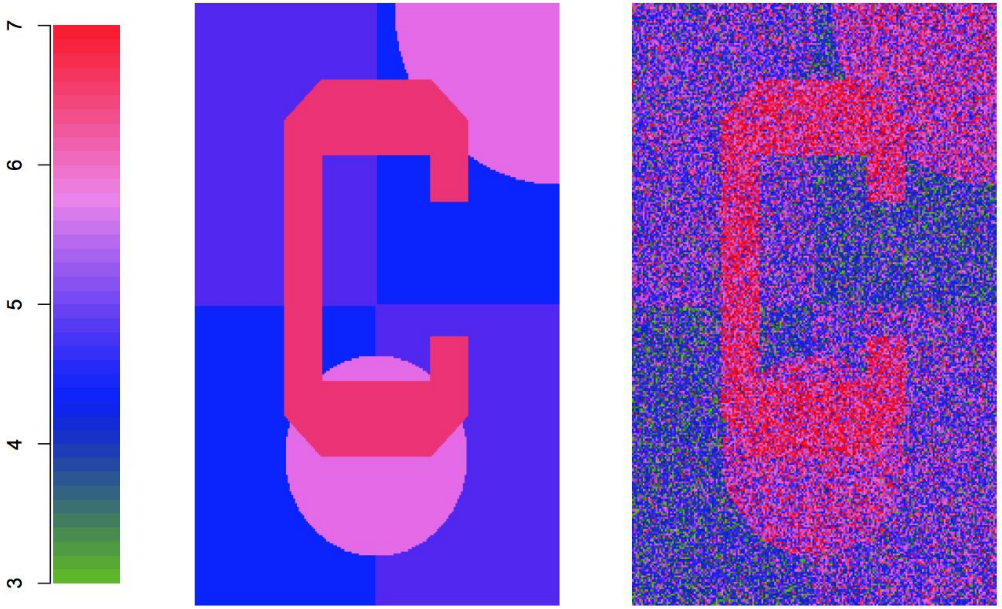


True image Data (Represented by y_i, $i = 1, \ldots, n$, n pixels)

Decision variable: θ_i value of pixel i in the denoised image

How to select the objective?
In true image, adjacent pixels mostly the same; different only on the shape edges

Denoising

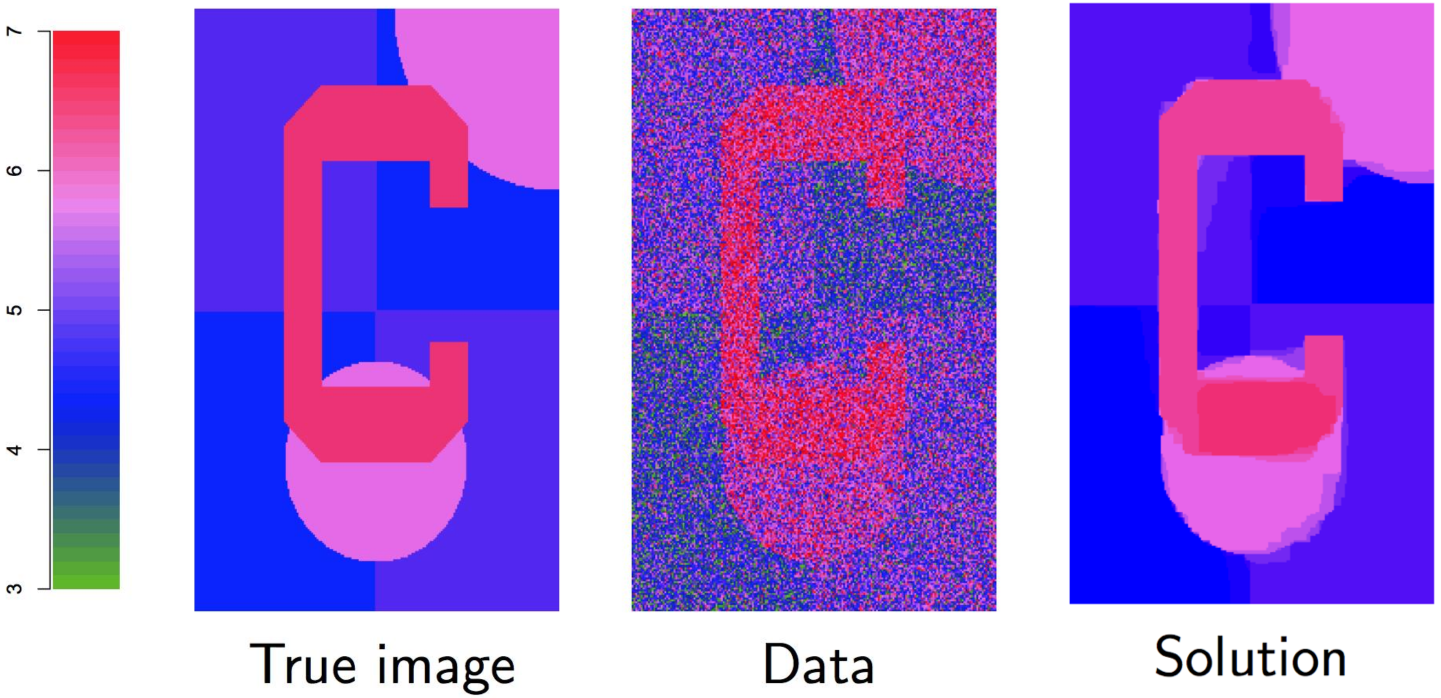


True image Data (Represented by y_i, $i = 1, \ldots, n$, n pixels)

Decision variable: θ_i value of pixel i in the denoised image

Objective: θ_i stays close to y_i, while we penalize changes in adjacent θ_i

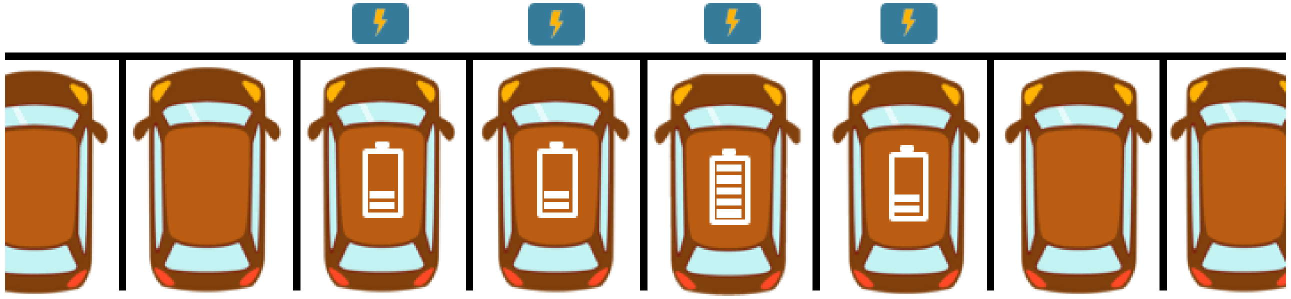
Denoising



$$\min_{\theta_1,\ldots,\theta_n} \frac{1}{2}\sum_{i=1}^{n}(y_i-\theta_i)^2 + \lambda \sum_{(i,j)\ \text{adjacent}} |\theta_i-\theta_j|$$

θ_i stays close to y_i penalize changes in adjacent pixels

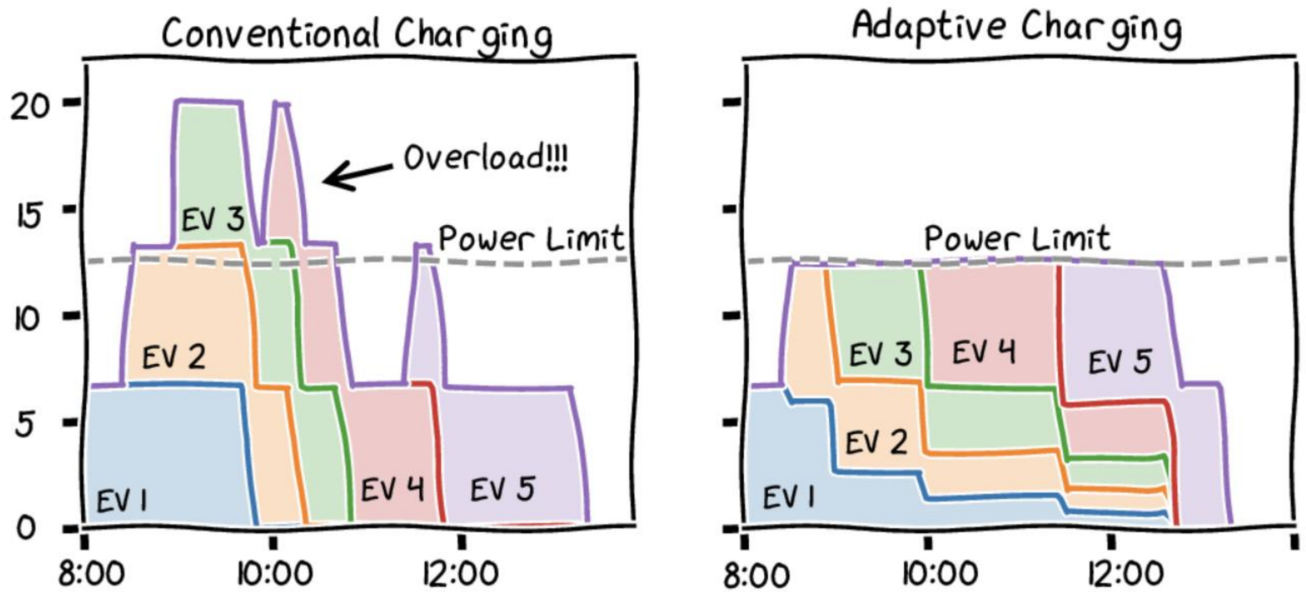
Energy Systems



Adaptive Charging Breaks Down Barriers to Electric Vehicle Adoption

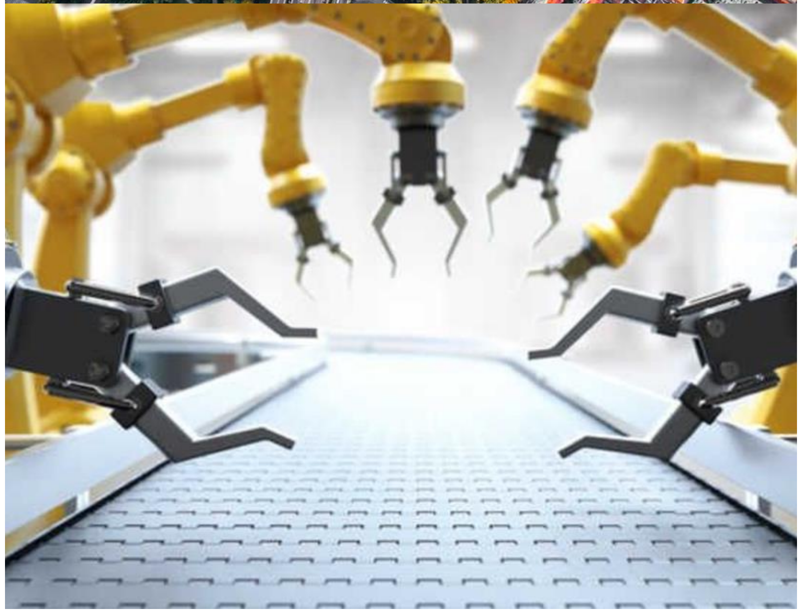
Zachary Lee | June 4, 2019

Energy Systems



Adaptive charging is able to stay below power limits while still meeting everyone's energy needs.

Optimization in real world



What you will learn from this class?

Engineering Problems ⇒ **Optimization Problems** ⇒ **Solutions**

1. Definition and classification of optimization problems

2. Formulate engineering problems as optimization problems

3. Find the solution of optimization problems

Definition and classification of optimization problems

Basic concepts
- Convex functions/sets
- Optimality condition
- Duality, KKT conditions

Types/properties
- Unconstrained vs. constrained
- Smooth vs. nonsmooth
- Strongly vs. “weakly” convex

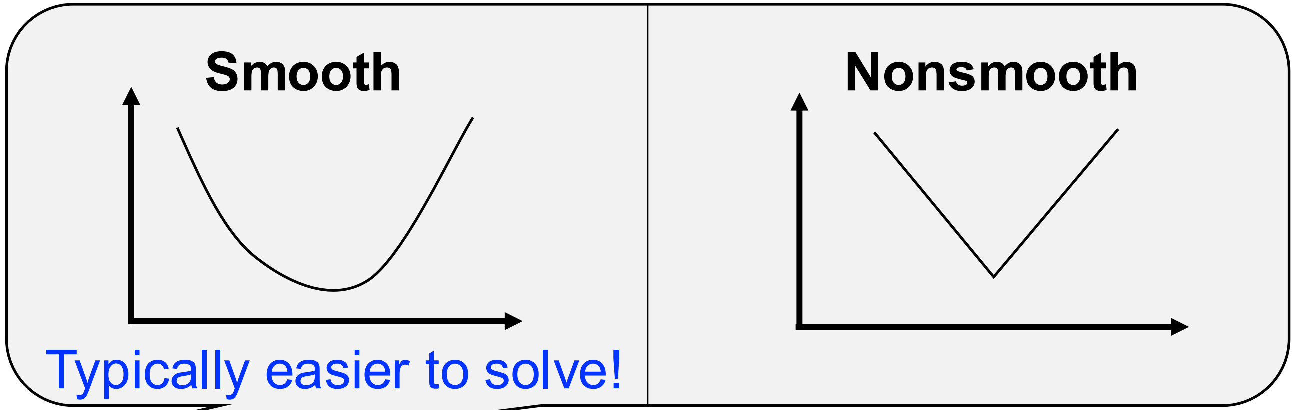
Canonical forms
- Linear Programming
- Quadratic Programming
- SOCP
- SDP

Unconstrained	**Constrained**
$\min_x f(x)$	$\min_x f(x)$
	s.t. $\quad x \in C$
Typically easier to solve!	

Definition and classification of optimization problems

Basic concepts
- Convex functions/sets
- Optimality condition
- Duality, KKT conditions

Types/properties
- Unconstrained vs. constrained
- Smooth vs. nonsmooth
- Strongly vs. “weakly” convex



Canonical forms
- Linear Programming
- Quadratic Programming
- SOCP
- SDP

Formulate engineering problems as optimization problems

- There are usually multiple optimization formulations for engineering applications
  - Some may be easier to solve, some harder
  - May need simplifications to the problem s.t. the resulting optimization problem is easy to solve
  - You will develop a capability of formulating an appropriate optimization problem from engineering applications (often times there is no single answer)!

- We will cover examples in
  - Energy/power networks
  - Signal processing
  - Control/robotics
  - Machine learning

Find the solution of optimization problems

- **Gradient-based methods**
  - Gradient descent (GD)
  - Sub-GD, proximal GD
  - Stochastic GD
  - Projected GD
  - Accelerated GD
- **Newton methods**
- **Duality-based methods**
  - Dual gradient method
  - Primal-dual method

Simplest, but only limited to unconstrained, smooth problems

Can handle nonsmooth objectives

Designed for large scale ML applications

Can handle simple constraints

Very fast in theory, but impractical for high-dimensional problems

Can handle general constraints

What you will learn from this class?

1. Definition and classification of optimization problems
2. Formulate engineering problems as optimization problems
3. Find the solution of optimization problems

Highly integrated!

Tentative Schedule

Week 1-3	• Convex functions, convex sets, convex optimization • Types, properties and canonical forms of convex optimization
Week 4	• Formulate engineering problems as optimization problems
Week 5-7	• Gradient based methods: GD, SGD, Subgradient, Proximal GD • Test 1 at week 7
Week 8-11	• Acceleration • Newton Method • Duality, KKT Conditions • Dual-based methods for solving constrained optimization
Week 12	• Test 2 at Week 12
Week 13-14	• Special topics and project

Basics: what is convex optimization and why we care?

Convex Optimization:
Optimization problems where both the objective and constraint are *convex*.

$$\min_x f(x)$$

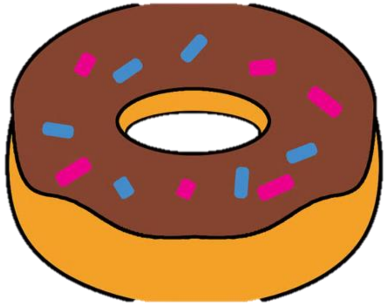
$$\text{s.t. } x \in C$$

What is convexity?

Definition in Merriam-Webster

convex adjective

1 **a** : curved or rounded outward like the exterior of a sphere or circle



Is this convex?

What is convexity?

Convex set: set $C \subseteq \mathbb{R}^n$ such that

$$x, y \in C \Rightarrow \underline{tx + (1-t)y} \in C, \text{ for all } t \in [0,1]$$

When $t = 0$, this becomes y

What is convexity?

Convex set: set $C \subseteq \mathbb{R}^n$ such that

$$x, y \in C \Rightarrow \underline{tx + (1-t)y} \in C, \text{ for all } t \in [0,1]$$

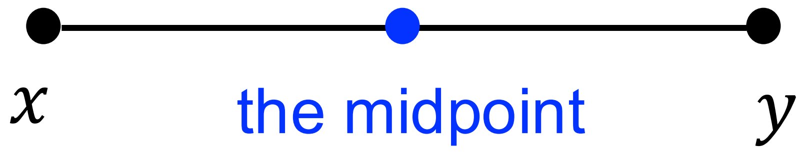
When $t = 1$, this becomes x

What is convexity?

Convex set: set $C \subseteq \mathbb{R}^n$ such that

$$x, y \in C \Rightarrow tx + (1-t)y \in C, \text{ for all } t \in [0,1]$$

Say $t = \frac{1}{2}$, this becomes $\frac{1}{2}x + \frac{1}{2}y$



What is convexity?

Convex set: set $C \subseteq \mathbb{R}^n$ such that

$$x, y \in C \Rightarrow \underline{tx + (1-t)y} \in C, \text{ for all } t \in [0,1]$$

When t ranges between $[0,1]$

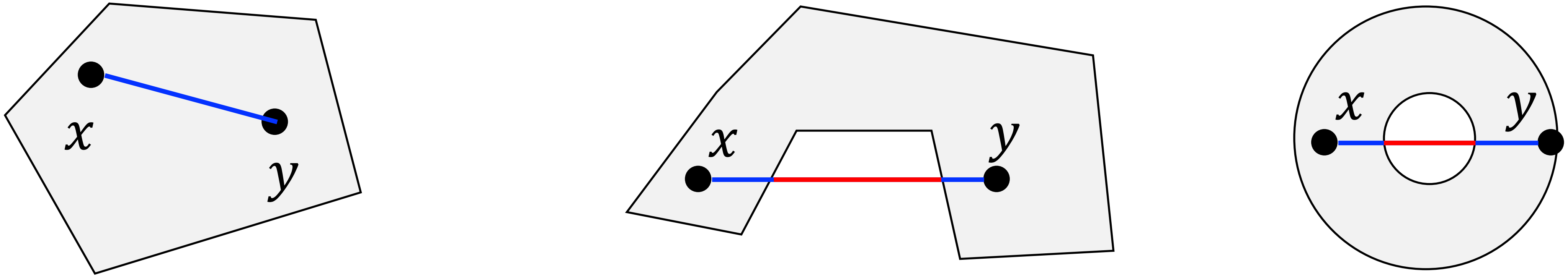
this becomes the entire line segment

What is convexity?

Convex set: set $C \subseteq \mathbb{R}^n$ such that

$$x, y \in C \Rightarrow tx + (1-t)y \in C, \text{ for all } t \in [0,1]$$

For any two points in the set, the entire line segment in between also lies in the set

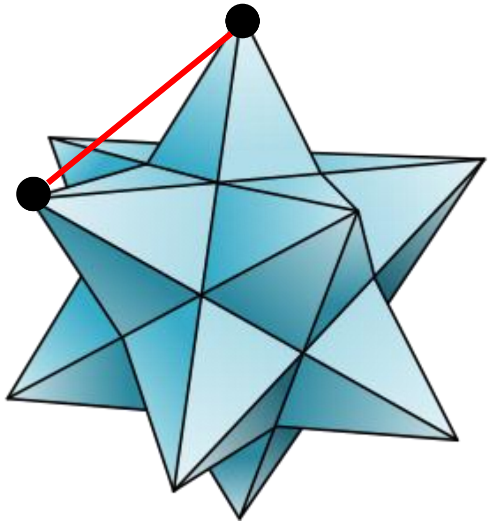


What is convexity?

Convex set: set $C \subseteq \mathbb{R}^n$ such that

$$x, y \in C \Rightarrow tx + (1-t)y \in C, \text{ for all } t \in [0,1]$$

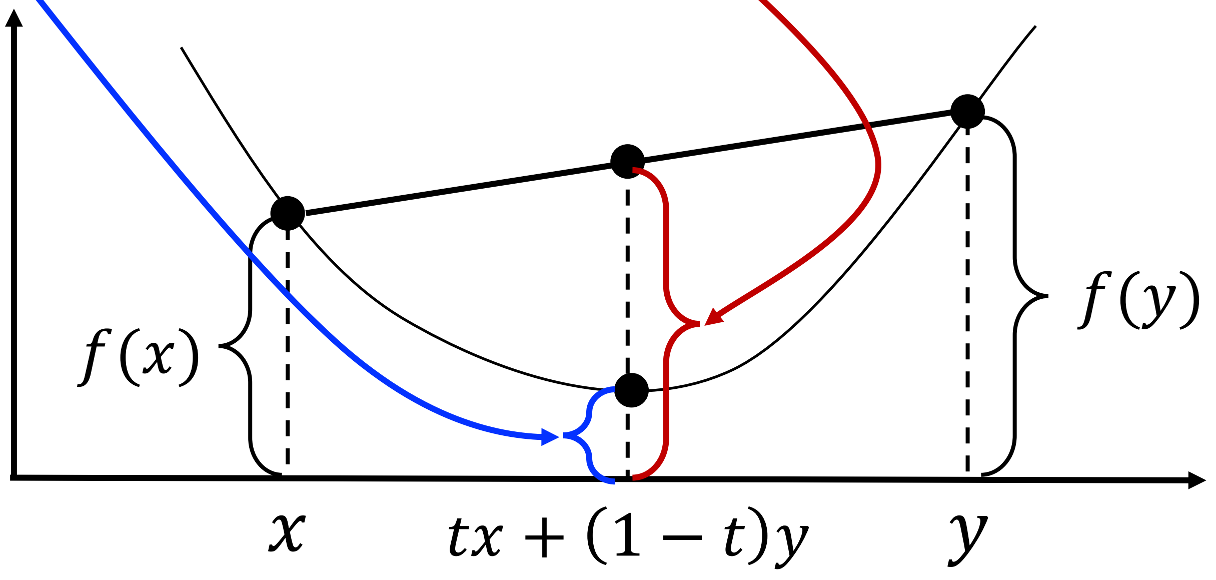
For any two points in the set, the entire line segment in between also lies in the set



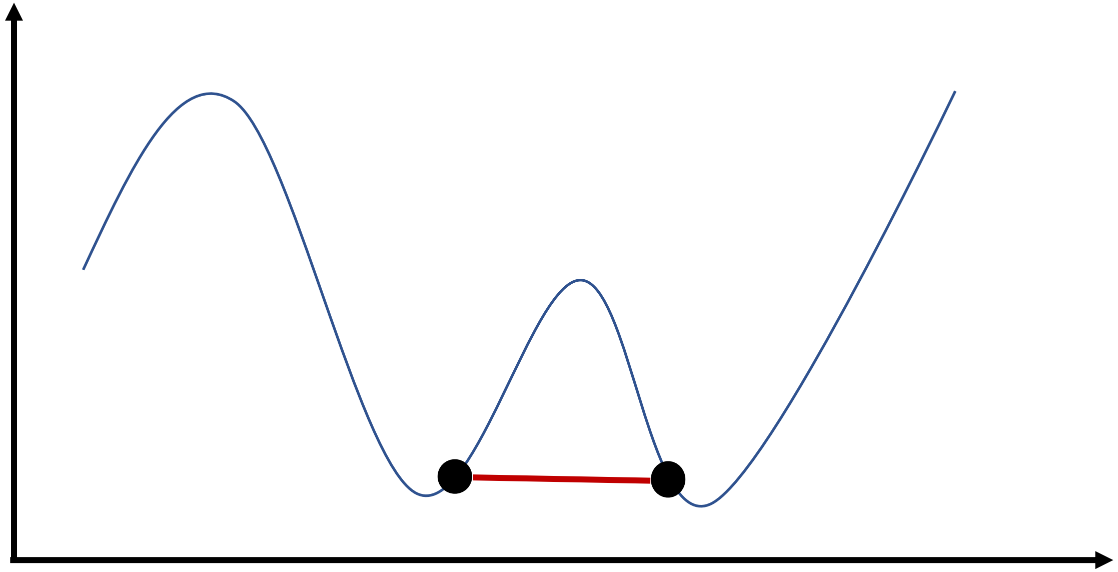
What is convexity?

Convex function: function $f\colon \mathbb{R}^n \to \mathbb{R}$ such that $\mathrm{dom}(f) \subseteq \mathbb{R}^n$ is convex, and

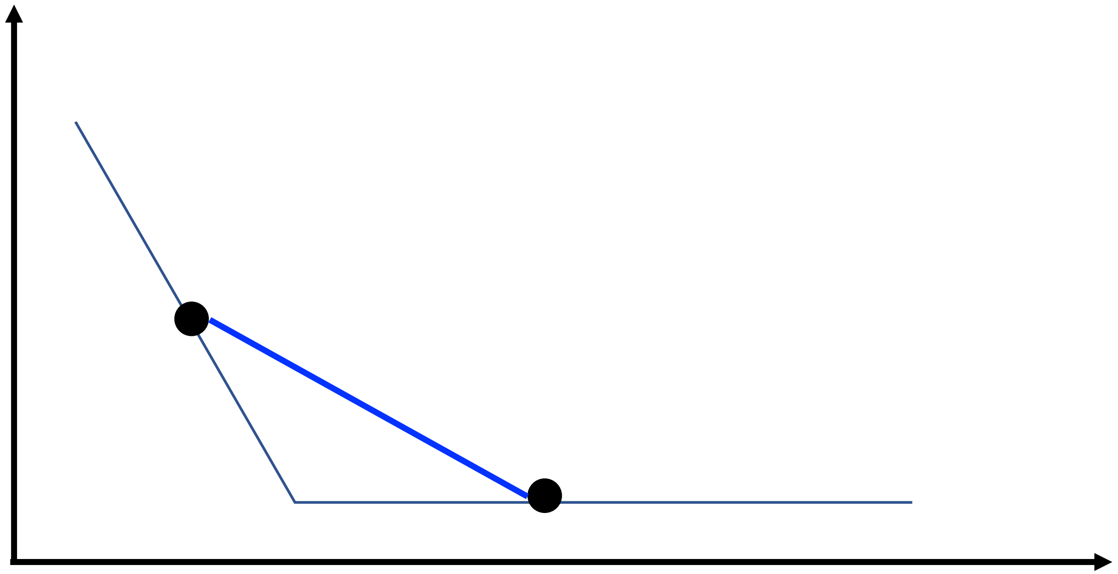
$$f(tx + (1-t)y) \le t f(x) + (1-t) f(y), \quad \text{for all } t \in [0,1]$$



Are these functions convex?

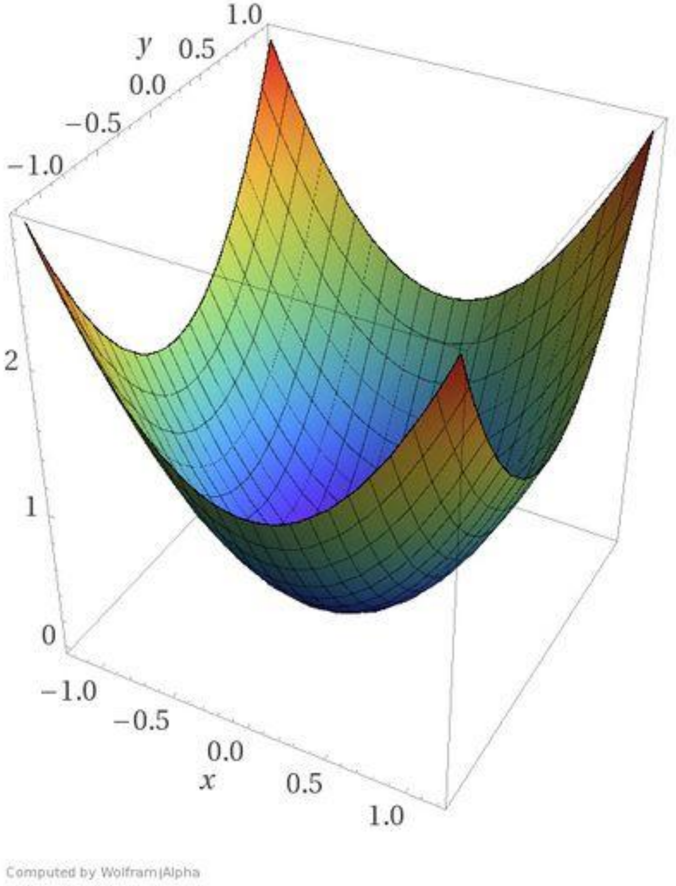


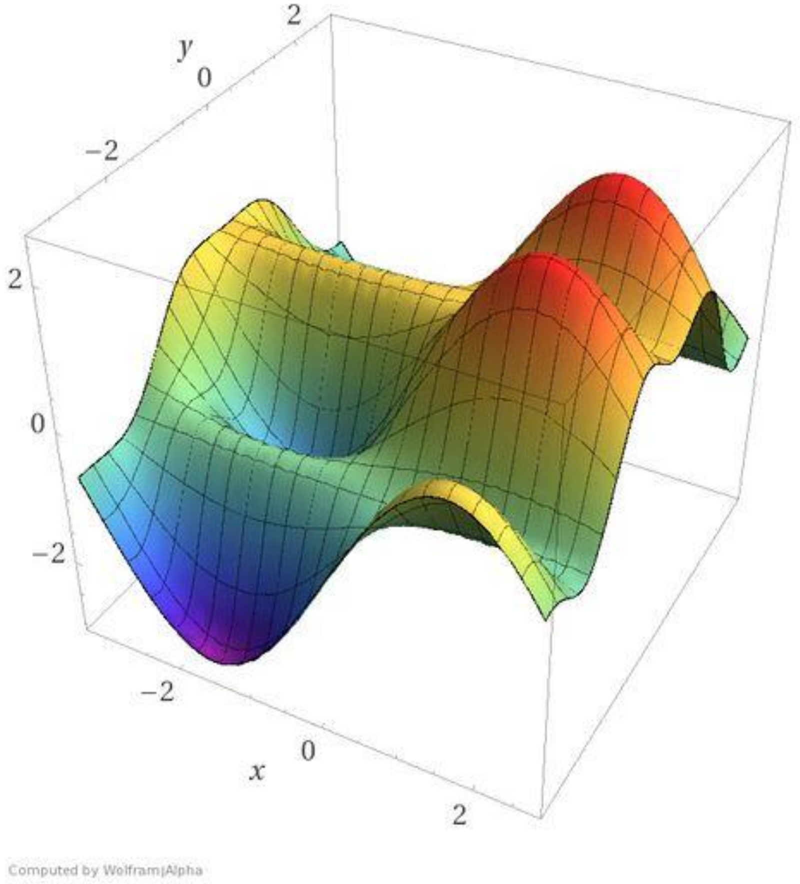
Nonconvex as red line lies below curve



Convex

Are these functions convex?





Convex Optimization

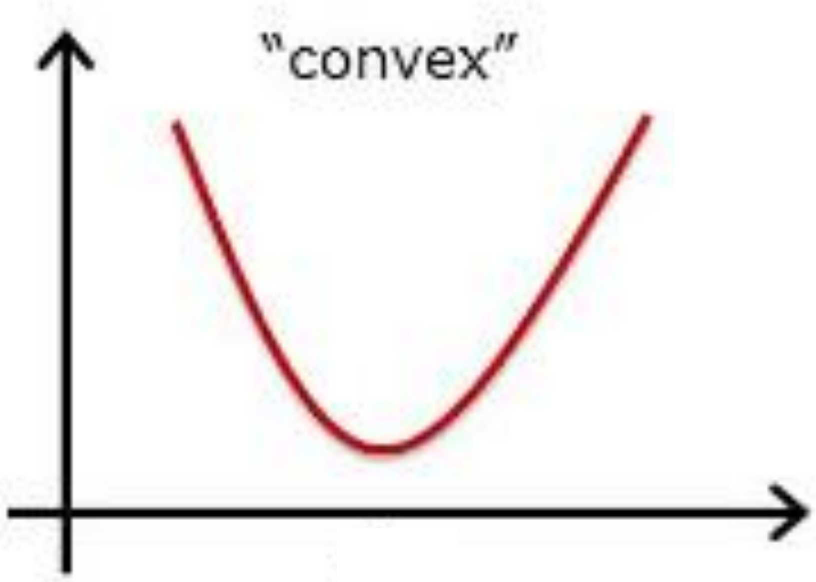
Optimization problems where both the objective and constraint are convex.

$$\min_{x} f(x)$$

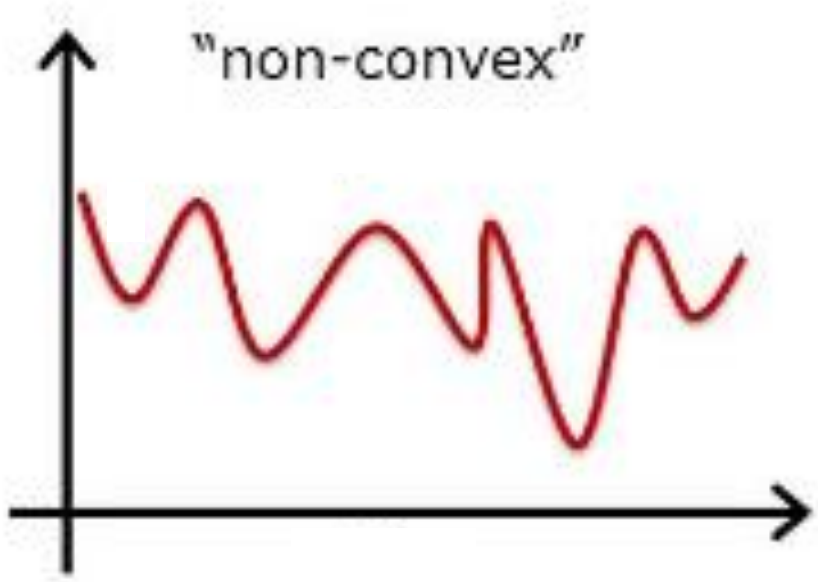
$$\text{s.t. } x \in C$$

Why we care convexity?

Convex functions are easier to optimize!



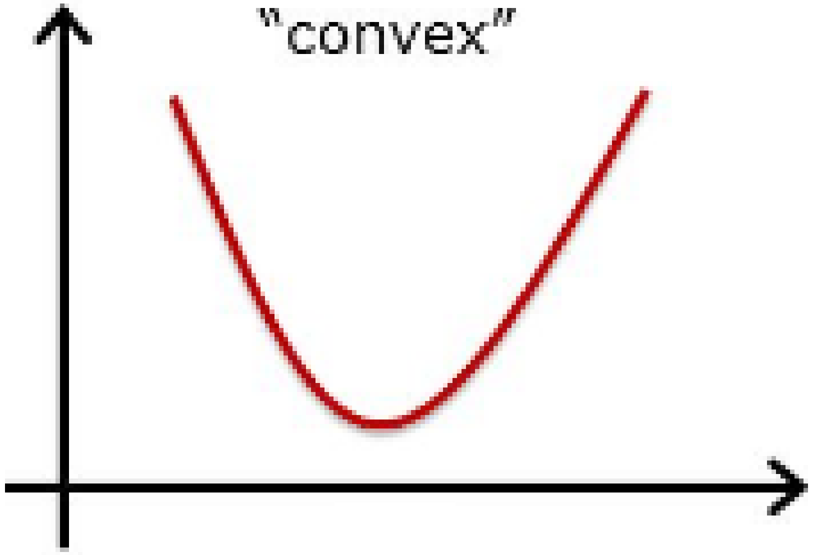
Single "Basin"



Multiple "Basins"

Why we care convexity?

Convex optimization already capture many real-world problems



Single "Basin"

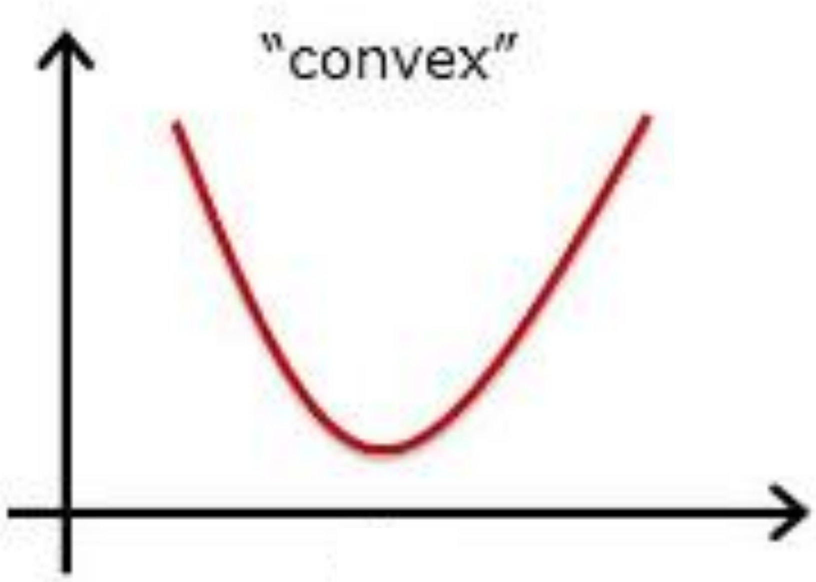
For example

- Common functions: linear, quadratic,…
- ML like linear/logistic regression, SVM…
- DC power flow, linear optimal control, …

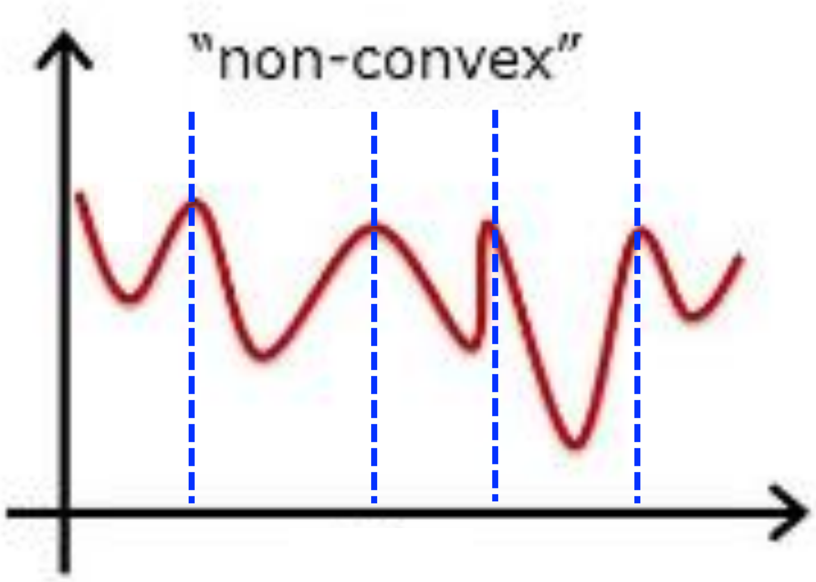
We will give many examples in the class

Why we care convexity?

Convexity serves as the basis to study non-convex optimization



Single "Basin"



Each individual region is convex

We will discuss non-convex optimization in the final lectures.

Summary

Today: A brief introduction to convex sets/functions/optimization

Next lecture: convex functions