IOE 611/ Math 663
Lecture 1: Introduction

Salar Fattahi

August 30, 2022

IOE 611/Math 663: Nonlinear Programming

- **Lectures:** Tue and Thu, 12:00–1:30 PM, 1005 Dow
- **Instructors:**
  - Salar Fattahi
    OH: Tuesdays 4:00-5:00PM, Wednesdays 9:00-10:00AM
  - Geyu Liang
    OH: Mondays 1:30:00-2:30PM, Fridays 12:30-1:30PM
  - OH will be on Zoom.
- **Main Textbook:** "Convex Optimization" by Stephen Boyd and Lieven Vandenberghe
  `https://web.stanford.edu/~boyd/cvxbook/`
  See syllabus for a complete list of references.
- **Required background:** Working knowledge of linear algebra and analysis (review Appendix A of the book). Also, a level of mathematical maturity to program in Matlab. Exposure to numerical computing, optimization, and its application fields is helpful.

Please read syllabus for the complete course policies.

Course Logistics

- Lectures:
  - Slides posted prior to lectures
  - In class, most mathematical proofs will be handwritten.
  - The annotated slides will be also posted.
- Problem sets every two weeks or so, include some computer modeling and programming (35% total)
- Two take-home exams (30% each)
- Active participation (5% total)
- Written solutions must be typed, LaTeX highly recommended

Covid policies

- If you feel sick, do not come in; the lectures will be streamed live, and will be recorded.
- No eating or drinking in class!

Mathematical optimization

x can be scalar, vector, matrix or even a tensor.

(mathematical) optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- $x = (x_1, \ldots, x_n)$: optimization variables (or decision variables)
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$: constraint functions

optimal solution $x^\star$ has smallest value of f_0 among all vectors that are *feasible, i.e.*, satisfy the constraints

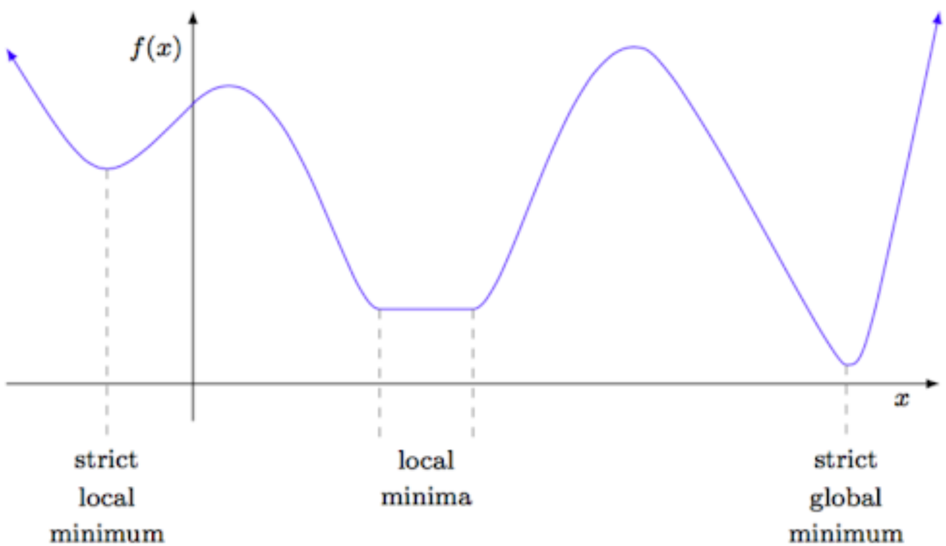
Constraint function's can be form, Laws of Physics, Budget etc.

Local/Global solutions

Best possible solution within a neighbourhood.

strict local min if $f(x) < f(y)$

- A point x is a **local minimum** of $f(x)$ if there exists $\delta > 0$ such that $f(x) \leq f(y)$ for every y such that $\|x - y\| \leq \delta$. (any norm)
- A point x is a **global minimum** of $f(x)$ if $f(x) \leq f(y)$ for every y.



Examples

portfolio optimization: looking for a best way to invest in a set of n assets

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return
- objective: risk

$x_i :=$ amount you invest in asset i

Constraint := $x_1 + x_2 + \cdots + x_n \le B$

Objective = maximize the Profit

= minimize the (−Profit)

Examples

portfolio optimization: looking for a best way to invest in a set of n assets

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return
- objective: risk

data fitting: from a family of potential models, find a model that best fits observed data and prior information

- variables: model parameters
- constraints: prior information, parameter limits
- objective: measure of misfit or estimation error

Solving optimization problems

solution method

- A *solution method* for a class of problems is an algorithm that computes the solution to a given *accuracy*, given an instance of a problem from the class.

$\Rightarrow$ Most important and most difficult part of solving any optimization problem is to come up with a solution method or algorithm.

$\Rightarrow$ for a class of problem's, Given accuracy

$\Downarrow$

Ideally there should be as Large as possible: (Ex: quadratic function's ax^2+bx+c)

Solving optimization problems

solution method

- A *solution method* for a class of problems is an algorithm that computes the solution to a given *accuracy*, given an instance of a problem from the class.

general optimization problem

- very difficult to solve
- methods involve some compromise, *e.g.*, very long computation time, or not always finding the solution

Least-squares

$\|x\|_2^2 = x^T x$

$$\text{minimize} \quad \|Ax - b\|_2^2$$

$A \in \mathbf{R}^{k\times n}$, $k \geq n$, $\mathbf{rank}(A) = n$

$$\|Ax-b\|_2^2 = (Ax-b)^T(Ax-b) = (x^TA^T - b^T)(Ax-b)$$

$$= x^TA^TAx - x^TA^Tb - b^TAx + b^Tb$$

Quadratic function of $x \Rightarrow \nabla f(x) = 0$

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} = 0 \Rightarrow \quad 2A^TAx - 2A^Tb = 0$$

$$\Rightarrow \boxed{x^* = (A^TA)^{-1}A^Tb}$$

Global solution for all A, b

Least-squares

$$\text{minimize} \quad \|Ax - b\|_2^2$$

$A \in \mathbf{R}^{k\times n}$, $k \geq n$, $\mathbf{rank}(A) = n$

solving least-squares problems

- analytical solution: $x^\star = (A^T A)^{-1} A^T b$

Least-squares

$$\text{minimize} \quad \|Ax - b\|_2^2$$

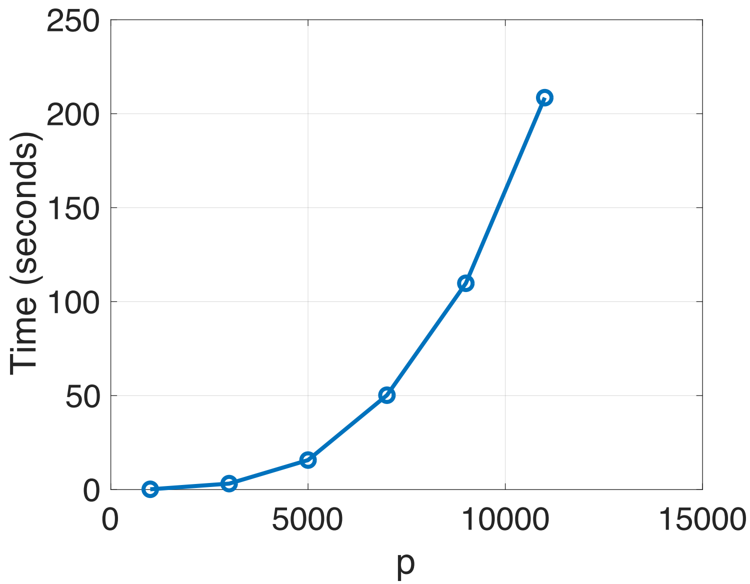
$A \in \mathbf{R}^{k \times n}$, $k \geq n$, $\mathbf{rank}(A) = n$

solving least-squares problems

- analytical solution: $x^\star = (A^T A)^{-1} A^T b$

or

Iterative method's Ex: Gradient descent
Stocastic " "



Linear programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m \end{array}$$

in Linear Programming there are two main Assumption:

① Objective should be Linear $(f_0(x))$

② constrain function's should be Linear $(f_i(x))$

Linear programming

$$
\begin{array}{ll}
\text{minimize} & c^T x \\
\text{subject to} & a_i^T x \leq b_i, \quad i = 1, \dots, m
\end{array}
$$

solving linear programs (No closed form sol^n)

- no analytical formula for solution No choice but to use iterative method's
- reliable and efficient algorithms and software
- computation time *in practice* proportional to $n^2 m$ if $m \geq n$; less with structure
- a "mature technology"

Linear Program is a very mature class of optimization problem's, we know exactly how to solve them, The algorithm's we have are very reliable, very efficient, and we have different solvere's.

Linear programming

developed Btw 1940 ~ 1980
=> WW II

widly used Algo => SIMPLEX
Developed by
Dantzig
(1948)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \le b_i, \quad i = 1, \ldots, m \end{array}$$

solving linear programs

in 1984 Karmarkar developed
Interior Point
to solve LP

- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time *in practice* proportional to $n^2 m$ if $m \ge n$; less with structure
- a "mature technology"

using linear programming

- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs (*e.g.*, problems involving ℓ_1- or ℓ_∞-norms, convex piecewise-linear functions)

Convex optimization problem

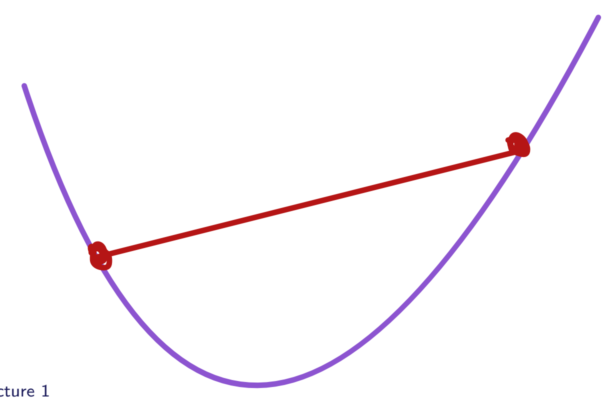
The Largest, Perhap's the most modern class of optimization problem's that are considered easy in set of convex optimization problem's (Main focus)

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- objective and constraint functions are convex: for $i = 0, 1, \ldots, m$,

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

  for any $x, y \in \mathbf{R}^n$ and any $\alpha \geq 0$, $\beta \geq 0$ such that $\alpha + \beta = 1$
- includes least-squares problems and linear programs as special cases



Convex optimization problem

solving convex optimization problems

- no analytical solution
- reliable and (sometimes) efficient algorithms

"In fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."
—Rockafellar 1993

Convex optimization problem

solving convex optimization problems

- no analytical solution
- reliable and (sometimes) efficient algorithms

"In fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."
—Rockafellar 1993

using convex optimization

- often difficult to recognize
- many tricks for transforming problems into convex form
- Convex relaxation techniques

Convex optimization is not easy identify or to recognize. infact we need to use lot of tricks to convert optimization problem to convex.

Nonconvex (nonlinear) optimization

traditional techniques for general nonconvex problems involve compromises

Even if it is non-convex, we can actually relax the problem into convex optimization problem.

Nonconvex (nonlinear) optimization

traditional techniques for general nonconvex problems involve compromises

local methods

- find a point that minimizes f_0 among feasible points near it, useful if goal is to find a good point
- fast, can handle large problems
- require initial guess
- provide no information about distance to (global) optimum

Nonconvex (nonlinear) optimization

traditional techniques for general nonconvex problems involve compromises

local methods

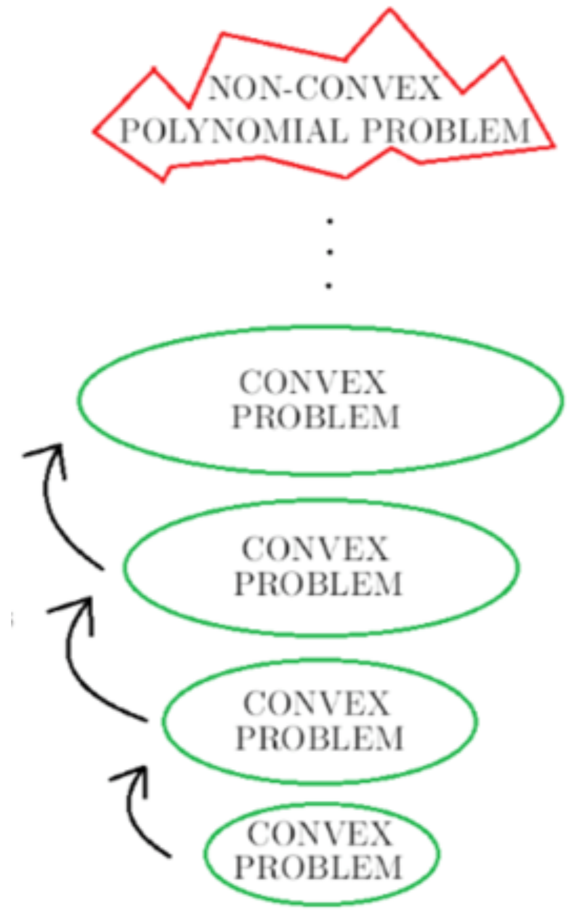
- find a point that minimizes f_0 among feasible points near it, useful if goal is to find a good point
- fast, can handle large problems
- require initial guess
- provide no information about distance to (global) optimum

global methods

- find the (global) solution
- worst-case complexity grows exponentially with problem size

these algorithms are often based on solving convex subproblems

Convex relaxation



Course goals and topics

goals

1. recognize/formulate problems as convex optimization problems
2. develop code for problems of moderate size
3. characterize optimal solution, give limits of performance, etc.

topics

1. Convex optimization
2. Numerical algorithms
3. Nonconvex optimization

Convex Sets

The First step to understand convex optimization

ie to understand convex sets.

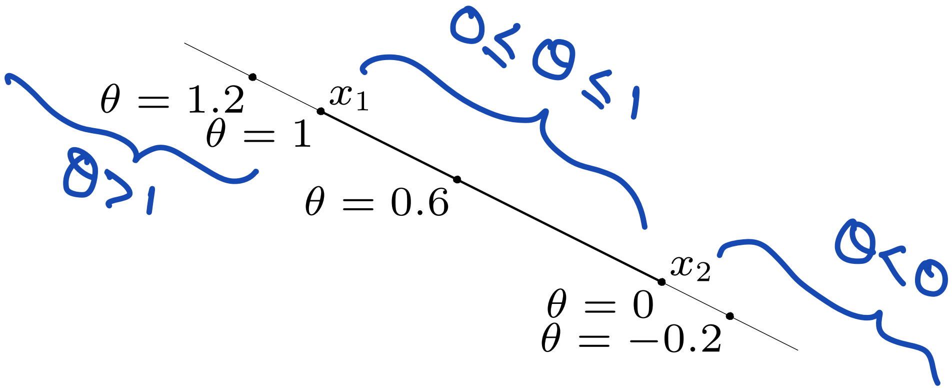
Outline

- Affine and convex sets, and convex cones
- Some examples
- Polyhedron
- Positive semidefinite cone

Affine set

affine combination of x_1 and x_2: any point x of the form

$$x = \theta x_1 + (1-\theta)x_2 \qquad \text{where } \theta \in \mathbf{R}$$



line through x_1, x_2: all affine combinations of x_1 and x_2

x can be a vector $\in \mathbb{R}^n$ or a matrix $\in \mathbb{R}^{n\times m}$

But θ Need's to be scalar.

$x = \theta x_1 + (1-\theta)x_2 \Rightarrow x = x_2 + \theta(x_1 - x_2)$

$\Rightarrow$ Parametric Line

Suppose $x_1 \neq x_2$ two points $\in \mathbb{R}^n$, The points of the form $x = \theta x_1 + (1-\theta)x_2$ $\theta \in \mathbb{R}$ form the line passing through x_1 and x_2

$$x(\theta=0) = x_2, \quad x(\theta=1) = x_1 \in \mathbb{R}^n$$

$$\Rightarrow \quad x = x_2 + \theta(x_1 - x_2)$$

$\Rightarrow$ $x_2 :=$ Base point

$x_1 - x_2 :=$ direction, and the direction is scaled by the parameter θ.

$\Rightarrow$ As θ increases from 0 to 1, the point x moves from x_2 to x_1; for $\theta > 1$, the point x lies on the line beyond x_1.

* Affine combination of two points in the line passing through those two points.

The definition can be extended to k points

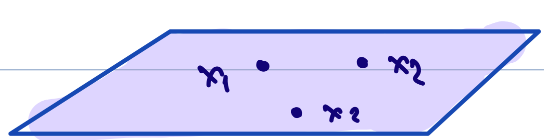
Let $x_1, x_2, \ldots, x_k$

The affine combination of these k points

say $x = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k$

where $\theta_1 + \theta_2 + \theta_3 + \cdots + \theta_k = 1$

Ex:



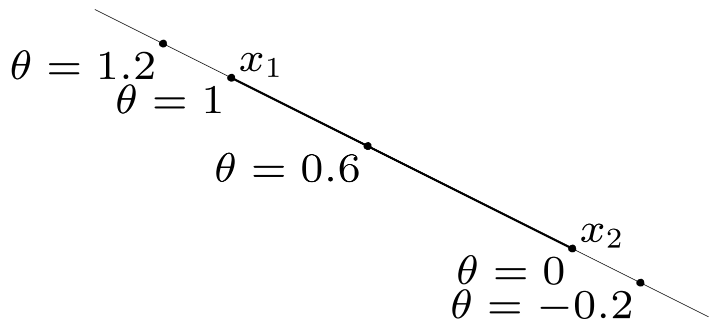
plane passing through x_1, x_2, x_3

$x \in \mathbb{R}^2$

Affine set

affine combination of x_1 and x_2: any point x of the form

$$x = \theta x_1 + (1 - \theta) x_2 \qquad \text{where } \theta \in \mathbf{R}$$



line through x_1, x_2: all affine combinations of x_1 and x_2

affine set: contains the line through any two distinct points in the set (i.e., a set that's closed under affine combinations)

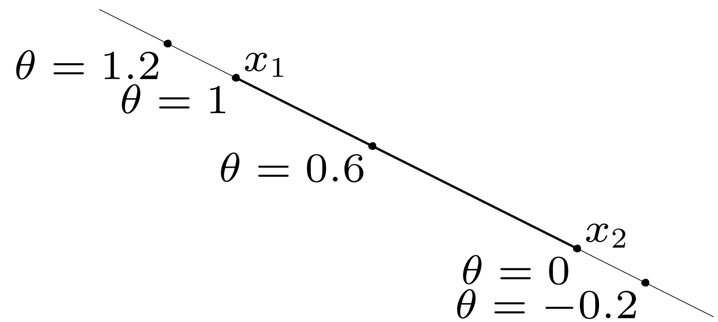
affine combination.

A set is Affine if it contain's the line through any two distinct points in set.

Affine set

affine combination of x_1 and x_2: any point x of the form

$$x = \theta x_1 + (1-\theta)x_2 \qquad \text{where } \theta \in \mathbf{R}$$



line through x_1, x_2: all affine combinations of x_1 and x_2

affine set: contains the line through any two distinct points in the set (i.e., a set that's closed under affine combinations)

example: solution set of linear equations $\{x \mid Ax = b\}$ $= C$

Proof:- Pick any two points $x_1, x_2 \in C$, the line passing through $x_1, x_2 \in C$

$\theta A x_1 + (1-\theta) A x_2 = b$

$x_1, x_2 \in C \Rightarrow$ $Ax_1 = b \quad \times \theta$

$Ax_2 = b \quad \times (1-\theta)$ $\Big\} + \Rightarrow$ $\Rightarrow A(\theta x_1 + (1-\theta)x_2) = b$

$\Rightarrow \theta x_1 + (1-\theta)x_2 \in C$

⟹ A set $C \subseteq \mathbb{R}^n$ is affine if the Line through any two distinct points in C Lies in C

⟹ $x_1, x_2 \in C$, $\theta \in \mathbb{R}$, then $\theta x_1 + (1-\theta) x_2 \in C$

⟹ In other word's C contain's the Affine Combination of any two Points in C, Provided co-effic's sum to 1.

⟹ Any Affine set is a solution set of Bunch of Linear inequalities.

⟹ in higher dim, Affine set's are subspaces like hyperplane's. it cannot be Bounded. because it should contain Line.

The set $\{x \mid Ax = b\} = C$ characterize every single Affine set.

Convex set's are more general than Affine set's.

Convex sets

convex combination of x_1 and x_2: any point x of the form

$$x = \theta x_1 + (1-\theta)x_2 \text{ where } \underbrace{0 \le \theta \le 1}_{\text{only difference.}}$$

line segment between x_1 and x_2: all points

$$x = \theta x_1 + (1-\theta)x_2$$

with $0 \le \theta \le 1$

Any point in the segment b/w x_1 and x_2. It's a segment.

Convex sets

convex combination of x_1 and x_2: any point x of the form

$$x = \theta x_1 + (1-\theta)x_2 \text{ where } 0 \leq \theta \leq 1$$

line segment between x_1 and x_2: all points

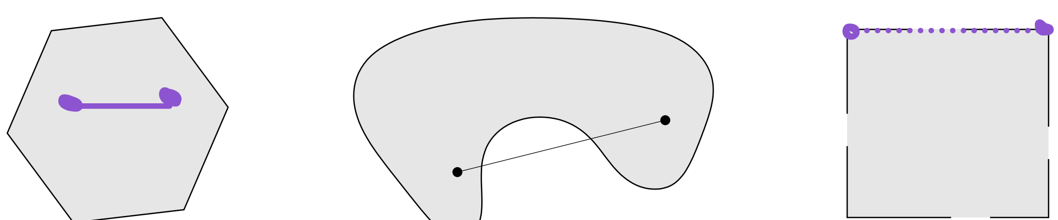
$$x = \theta x_1 + (1-\theta)x_2$$

with $0 \leq \theta \leq 1$

convex set: contains line segment between any two points in the set (i.e., closed under convex combinations)

$$x_1, x_2 \in C, \quad 0 \leq \theta \leq 1 \quad \Longrightarrow \quad \theta x_1 + (1-\theta)x_2 \in C$$

examples (one convex, two nonconvex sets)



Convex Not-convex Not - convex

Convex combination and convex hull

convex combination of $x_1, \ldots, x_k$: any point x of the form

$$x = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k$$

with $\theta_1 + \cdots + \theta_k = 1$, $\theta_i \geq 0$ (Non-negative co-efficients)

A set is convex $\Longleftrightarrow$ if contain's any convex combination of any finite number of its points.

Proof $\Longleftarrow$

if the set contain's any convex combination of any finite number of its points the claim is $\Longrightarrow$ The set is convex

By definition

$\Longrightarrow$ requires little bit of work (Homework)

Hint: use Induction.

Convex combination and convex hull

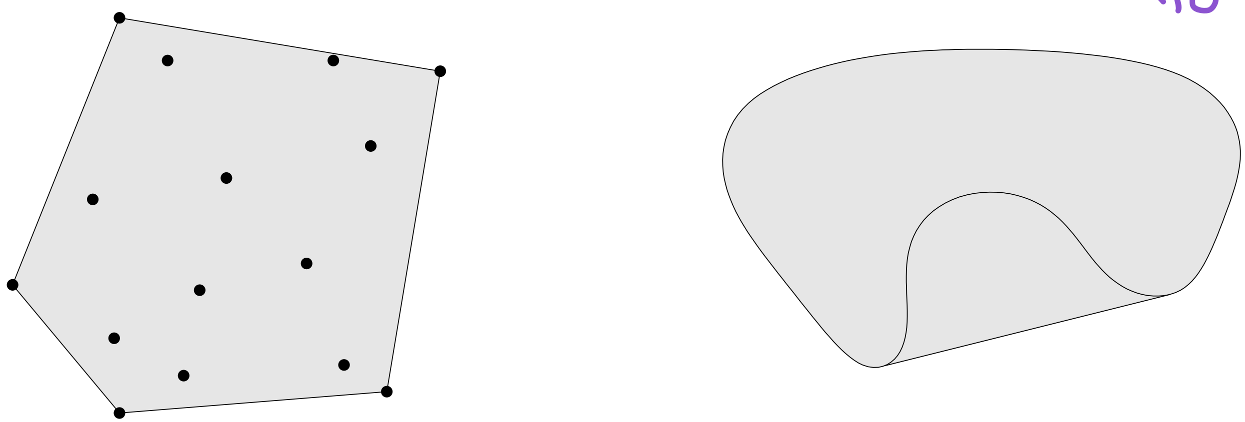
convex combination of $x_1, \ldots, x_k$: any point x of the form

$$x = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k$$

with $\theta_1 + \cdots + \theta_k = 1$, $\theta_i \geq 0$

convex hull $\mathbf{conv}\, S$: set of all convex combinations of points in $\underline{S}$

does not have to be convex.



① convex hull by definition is a convex set

Suppose we have a set S. does not have to be convex set, can be discrete set, can be 0 or 1 points.

$\Rightarrow$ The set of all convex combination's of the points in S is called convex hull.

② Convex hull of a set S is the smallest convex set that contain's S.

D = convex set

By contradiction suppose $\exists D : S \subseteq D$
$\operatorname{conv} S \not\subseteq D$

$\Rightarrow \exists x \in \operatorname{conv} S$ s.t $x \notin D$

$x = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k$

where $x_1, x_2, \cdots, x_k \in S$

$\theta_1 + \theta_2 + \cdots + \theta_k = 1$

$x_1, x_2, \cdots, x_k \in S$

$\Rightarrow x_1, x_2, \cdots, x_k \in D$ (Because $S \subseteq D$)

& D is a convex set

$\Rightarrow \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k \in D$

(contradiction)