Lecture 0: Introduction and overview

- organizational matters
- prerequisites
- some general features of convex optimization
- example
- duality example
- what we will/won't do

Course organization

- Shuguang (Rober) Cui, sgcui@ucdavis.edu, 3131 Kemper Hall
- course material (lecture notes, homework, project etc):

  `via Canvas and email`

- Office hour: Wed. 12:30∼1:30pm
- reference/textbooks
  1. S. Boyd and L. Vandenberghe, **Convex Optimization**.
     `http://www.stanford.edu/~boyd/cvxbook/`
  2. D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, **Convex Analysis and Optimization** Athena Scientific
- lecture slides on convex optimization based mainly on the slides of Luo, Boyd, and Vandenberghe

Prerequisites

- good knowledge of linear algebra
- elementary probability
- exposure to electrical engineering (communications, signal processing, information theory, networking,. . .)
- elementary analysis (norms, limits, . . .)
- knowledge of Matlab, or willingness to learn

not required but helps

- exposure to optimization
- numerical linear algebra

Course topics

- **Tentative Course Outline**:

  1. Introduction
  2. Convex sets and functions
  3. Convex optimization problems
  4. KKT conditions and duality
  5. Numerical solutions
  6. Application examples

- **Course Requirement and Grading:**

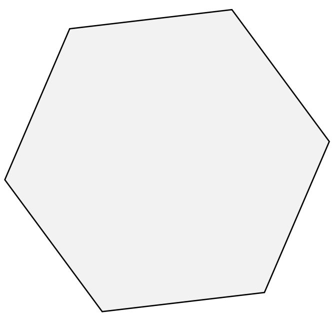
  Homework 20%, midterm 30%, class participation 10%, and project 40%.

Convex set

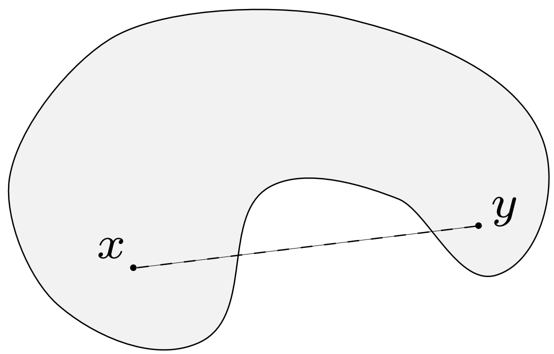
$C \subseteq \mathbf{R}^n$ is convex if

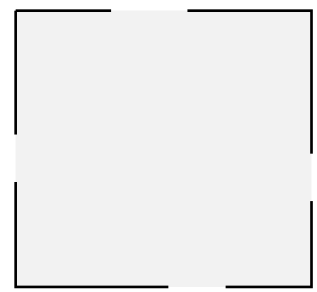
$$x, y \in C, \quad \theta \in [0,1] \implies \theta x + (1-\theta) y \in C$$

convex



not convex



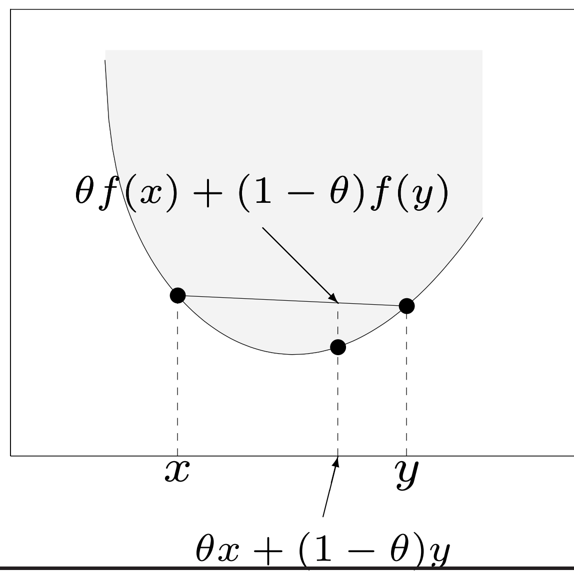


(more later!)

Convex function

$f : \mathbf{R}^n \rightarrow \mathbf{R}$ is convex if

$$x, y \in \mathbf{R}^n, \quad \theta \in [0,1] \quad \Longrightarrow \quad f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y)$$



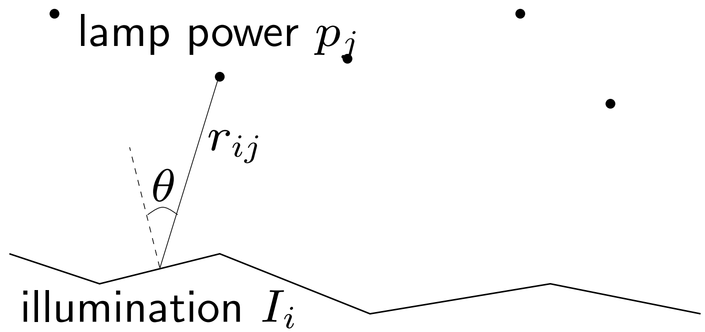
Convex optimization problem

minimize $f(x)$ subject to $x \in C$, with f convex, C convex

- can be solved numerically with great efficiency
- have extensive, useful theory
- occur often in engineering problems
- often go unrecognized
- **tractable** in theory and practice: there exist algorithms such that
  - computation time small, grows gracefully with problem size
  - global solutions attained
  - non heuristic stopping criteria; provable lower bounds
  - handle nondifferentiable as well as smooth problems
- **duality theory**:
  - necessary and sufficient conditions for global optimality
  - certificates that **prove** infeasibility or lower bounds on objective
  - sensitivity analysis w.r.t. changes in f, C

Example

m lamps illuminating n (small, flat) patches



$$I_i = \sum_{j=1}^{m} a_{ij} p_j, \qquad a_{ij} = r_{ij}^{-2} \max\{\cos\theta_{ij}, 0\}$$

lamp power limits: $0 \le p_j \le p_{\mathsf{max}}$

problem (one possible formulation): variables p_j

$$\text{minimize} \quad \max_{i=1,\ldots,n} |\log I_i - \log I_{\mathsf{des}}|$$

how to solve?

1. uniform power: $p_i = p$, vary p; could try heuristic adjustment of powers
2. least squares: minimize $\sum_j (I_j - I_{\text{des}})^2$

   (closed form, widely available and reliable software, fast)
   what if $p_i \geq p_{\text{max}}$ or $p_i \leq 0$?
   could 'saturate' or add weights:

$$\text{minimize} \sum_j (I_j - I_{\text{des}})^2 + \sum_i w_i (p_i - p_{\text{max}}/2)^2$$

Of course these are approximate 'solutions'.

- Using the knowledge from this course, one can show that this problem can be formulated as a convex optimization problem, hence is readily solved
- exact solution obtained with effort $\approx$ modest factor times least squares effort

Variants: two additional constraints

1. no more than half total power is in any 10 lamps
2. no more than half of the lamps are on ($p_i > 0$)

does adding (1) or (2) complicate the problem?

- with (1), still easy to solve
- with (2), **extremely difficult** to solve

Moral:

- without the proper background ($i.e.$, this course) very easy problems can appear quite similar to very difficult problems
- (untrained) intuition doesn't always work

Application of duality in algorithms

1. feasibility problem: find $x \in C$

- convex optimization methods

$$\text{either find } x \in C\text{, or yield proof that } C = \emptyset$$

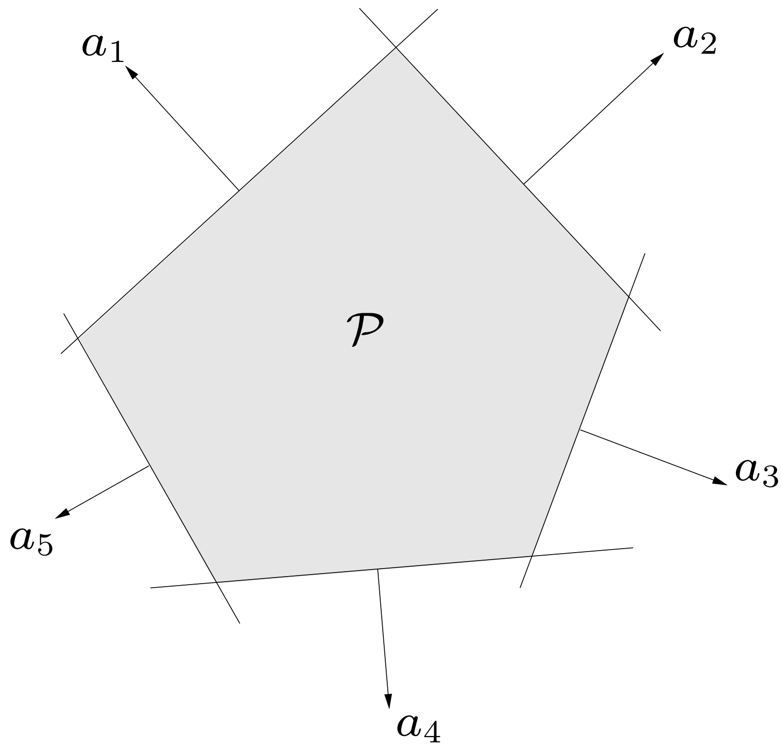
  c.f. conventional case: algorithms either

$$\text{find } x \in C\text{, or do not find } x \in C$$

- convex case: feasibility algs. return **yes** or **no**
  general case: feasibility algs. return **yes** or **maybe**

example:

$$\mathcal{P} = \left\{ x \mid a_k^T x \leq b_k, k = 1, \ldots, m \right\}$$



how could you know $\mathcal{P} = \emptyset$?

Here is how:

suppose $\lambda_i \geq 0$, $\sum \lambda_i a_i = 0$, $\sum \lambda_i b_i < 0$

then $a_i^T x \leq b_i$, $i = 1, \ldots, m$, implies

$$0 \leq \sum_i \lambda_i (b_i - a_i^T x) = \sum_i \lambda_i b_i < 0$$

$\rightarrow$ **Contradiction!**

we conclude:

$$\exists \lambda_i \geq 0, \ \sum \lambda_i a_i = 0, \ \sum \lambda_i b_i < 0 \Longrightarrow \mathcal{P} = \emptyset$$

we say λ_i's are a *certificate* or *proof* of infeasibility

fact (convexity): if $a_i^T x \leq b_i$ is infeasible, then there exists a certificate proving it!

certificate useful several ways:

- know for sure that $\mathcal{P} = \emptyset$
- can conclude that some 'relaxed' specs are also infeasible:

$$a_i^T x \leq b_i + \epsilon$$

  is infeasible for $\epsilon < (-\sum_i \lambda_i b_i)/(\sum_i \lambda_i)$

$\texttt{feas}(a_1, \ldots, a_m, b_1, \ldots, b_m)$ returns either

- feasible point x
- certificate of infeasibility λ

2. stopping criterion

convex optimization algorithms provide at iteration k

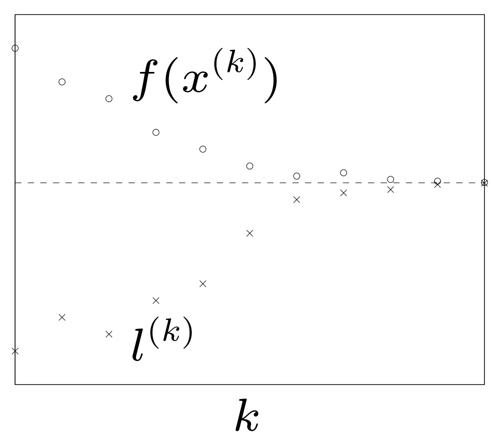
$$x^{(k)} \in C, \text{ a suboptimal point,}$$

with $f(x^{(k)}) \to f^\star = \inf_{x \in C} f(x)$ as $k \to \infty$

and a provable lower bound on optimal value, *i.e.*,

$$l^{(k)} \text{ s.t. } l^{(k)} \leq f^\star$$

with $l^{(k)} \to f^\star$ as $k \to \infty$



at iteration k we **know**

$$f^\star \in \left[l^{(k)}, f(x^{(k)}) \right]$$

hence stopping criterion

$$\textbf{until } f(x^{(k)}) - l^{(k)} \leq \epsilon$$

guarantees on exit

$$\text{absolute error } = \left| f(x^{(k)}) - f^\star \right| \leq \epsilon$$

similarly, stopping criterion

$$\textbf{until } \left(l^{(k)} > 0 \;\; \& \;\; \frac{f(x^{(k)}) - l^{(k)}}{l^{(k)}} \leq \epsilon \right)$$

guarantees (for $f^\star > 0$) on exit

$$\text{relative error } = \frac{f(x^{(k)}) - f^\star}{f^\star} \leq \epsilon$$

what we will cover

- recognizing & exploiting convexity in engineering context
- ideas of convex optimization
- a few algorithms extremal on the run time/code time tradeoff curve

what we won't do

- details of convex **analysis**
- details of optimization **theory** (regularity conditions, constraint qualifications, . . .)
- encyclopedia of algorithms

What fraction of 'real' problems are convex?

- by no means all
- many more than are recognized

analogy: linear programs

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

- no "closed form" solution
- very large LPs solved very quickly in practice
- extensive, useful theory

how many problems are LPs?

- 1940s: "the real world is nonlinear, hence LP silly"
- a few examples known (in planning)
- **many** examples found **after** LP became widely known. . .

Why convex optimization?

convex optimization

- dividing line between 'easy' and 'hard' optimization problems
- no local min; always global optimal solution;
- no such headaches as stepsize selection, initialization, etc
- handles* some problems very well; highly efficient algorithms exist
- can say a lot about it

other *wildly* used methods: simulated annealing, genetic algorithms, neural networks, ...

- handle† many problems
- slow convergence (they are too general to be efficient)
- can say very little about it

* means a lot — global solutions, always works, worst case computation time, etc.

† means much less — local solutions (sometimes), no complexity theory, etc.