CONVEX OPTIMIZATION

Lecture 1: Introduction.

+) An optimization problem in general has the form:

$$\begin{array}{lll} \min & f_0(x) & \\ \text{s.t.} & f_i(x) \leq b_i & i = 1, \dots, m \end{array}$$

- Vector $x = (x_1, x_2 \dots x_n) \in \mathbb{R}^n$ is the optimization variable
- function $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective function.
- functions $f_i : \mathbb{R}^n \to \mathbb{R}$ are constraint functions

  $i = 1, \dots, m$ $\quad m$: # of constraints.

- Optimal value or solution is a vector x^* that has the smallest objective function among all vectors that satisfy the constraints.

$$f_0(x^*) \leq f_0(z) \quad \forall z : f_i(z) \leq b_i, \ i = 1 \dots m.$$

+) A general optimization problem is difficult to solve (to find the optimal value).

There are certain classes of problems that can be solved efficiently and reliably. Convex optimization problems make a (large) such class.

+) Convex optimization problem: all constraints and objective functions are convex. (more later).

- Includes several well-known classes of optimization problems as special cases: least-squares, linear programming (among others).

+) <u>Least-squares problems</u>:

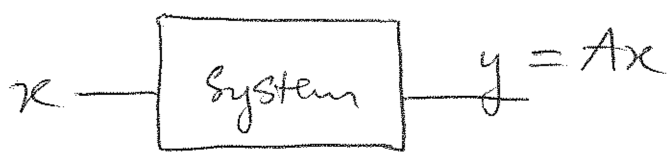
$$\min \ \|Ax - b\|_2^2 \qquad A \in \mathbb{R}^{k \times n} \ (k \geq n), \quad x \in \mathbb{R}^n$$

- unconstrained optimization
- objective function

$$f_0 = \|Ax - b\|_2^2 = \sum_{i=1}^{k} (a_i^T x - b_i)^2$$

$$\underset{\text{tall matrix}}{A} = \begin{array}{cc} k & \begin{bmatrix} \text{---} \ a_1^T \ \text{---} \\ \text{---} \ a_2^T \ \text{---} \\ \vdots \\ \text{---} \ a_{k-1}^T \ \text{---} \\ \text{---} \ a_k^T \ \text{---} \end{bmatrix} \\ & n \end{array} \qquad y = Ax = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{k-1} \\ y_k \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{k-1} \\ b_k \end{bmatrix}$$

- <u>Examples</u>:



- estimation in communication, control:
  - system equation $y = Ax$
  - you observe k output instances that form vector b
  - you want to estimate the input variable x.

(Note: If we add noise and want to minimize the average square error in this estimation, we will get the MMSE estimator).

- data fitting:
  - you are given a large amount of data
    $(u_1, b_1), (u_2, b_2), \ldots, (u_k, b_k)$
  - you want to find a function f that matches this data as closely as possible

$$\min \sum_{i=1}^{k} (f(u_i) - b_i)^2$$

- Say you want to do polynomial fitting

$$f(u) = x_1 + x_2 u + \dots + x_n u^{n-1}$$

Then for each vector $x = (x_1, x_2 \dots x_n)$ we can compute the error vector

$$e = (f(u_1) - b_1, f(u_2) - b_2), \dots, f(u_k) - b_k)$$

We can formulate an optimization problem that minimizes the norm of this error vector

$$\min \|e\|_2^2 = \|Ax - b\|_2^2$$

where $A \in \mathbb{R}^{k \times n}$, $A_{ij} = u_i^{j-1}$

$$A = \begin{bmatrix} 1 & u_1 & u_1^2 & \dots & u_1^{n-1} \\ 1 & u_2 & u_2^2 & \dots & u_2^{n-1} \\ & & \ddots & & \\ 1 & u_k & u_k^2 & \dots & u_k^{n-1} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{k-1} \\ b_k \end{bmatrix}$$

- The good news: least-squares problems can be solved analytically, in closed-form!

$$x^* = (A^T A)^{-1} A^T b$$

- Numerical computation of this solution (numerical solvers) can be carried out in approximately $n^2 k$ unit time.

- The main computational cost is in the matrix inversion.

- Can exploit structure of A (such as sparsity) to reduce the computation time

- Desktop computers can do of orders $n = 10K$'s, $k = 100K$'s in minutes. Larger problems (millions of variables) are challenging.

+) Linear programming:

$$\min \; c^T x$$
$$\text{s.t. } a_i^T x \le b_i \qquad i = 1, \dots, m.$$

$c, a_1, \dots, a_m \in \mathbb{R}^n$ (vectors)
$b_1, \dots, b_m \in \mathbb{R}$ (scalars) > problem parameters

- No simple analytical solution as in least-squares
- Well-developed methods for solving them numerically.
- Complexity is of the order $n^2 m$ (for $m \ge n$) but with a constant factor less well-characterized than least-squares.

- Examples: Chebyshev approximation problem (minimax)

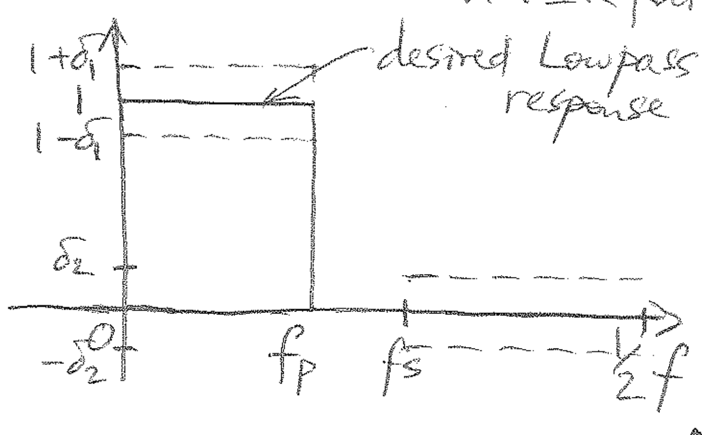
$$\text{minimize } \max_{i=1\dots k} |a_i^T x - b_i|$$

This problem can be reformulated into an LP as:

$$\min \; t$$
$$\text{s.t. } a_i^T x - t \le b_i \qquad i = 1, \dots, k$$
$$-a_i^T x - t \le -b_i$$

- Chebyshev approximation problem is used in FIR filter design.

- Given a desired filter frequency response $D(f)$



- We want to design an FIR filter with coefficients $x = (x_1, x_2 \dots x_n)$ to closely match the desired response, such that it minimizes the maximum difference in the frequency response.

FIR filter $x = (x_1, x_2 \dots x_n)$

$\rightarrow$ frequency response $\quad H(f) = \sum_{k=1}^{n} x_k e^{-j2\pi f(k-1)}$

Weighted error $\quad E(f) = \underset{\text{weight}}{W(f)} \left[\underset{\text{FIR response}}{H(f)} - \underset{\text{desired response}}{D(f)} \right]$

Then we want to design x such that

$$\min \max_{f \in \mathcal{F}} |E(f)| \qquad (\mathcal{F} \subset [0, \tfrac{1}{2}))$$

is a set of normalized frequencies.

This problem is a Chebyshev approximation which can be transformed into a linear programming problem. The difference here is that the set of constraints is continuous (must apply for all $f \in \mathcal{F}$) and therefore it is an LP with a finite number of variables in presence of an infinite number of constraints.

But we will stay in the domain of finite numbers of constraints for now.

+) Convex optimization:

$$\min \; f_0(x)$$
$$\text{s.t.} \quad f_i(x) \leq b_i, \quad i = 1, \dots, m$$

All objective and constraints are convex:

$$f_i(\theta x + \bar{\theta} y) \leq \theta f_i(x) + \bar{\theta} f_i(y), \quad i = 0 \dots m$$

$0 \leq \theta \leq 1$, $\bar{\theta} = 1 - \theta$

This includes least-squares and LP as special cases.

– About convex optimization:
- usually no analytical solutions (although for some there is!)
- but can understand and say a lot about optimality through duality and KKT conditions.
- has a unique solution or else the problem is infeasible
- can design algorithms to solve the problem efficiently and reliably.

– Using convex optimization:
- often difficult to recognize (if a problem is convex)
- many tricks to transform a problem into convex form.
- increasingly more problems are recognized as convex or can be transformed into convex problems
- also plays a role in nonlinear (non-convex) optimization by providing reliable bounds or good starting points.

– Compared to general nonlinear optimization:

+) local optimization:
- finds local optimal point
- fast, can handle large problems
- requires good initial guess
- no information about how close the solution to global optimum.

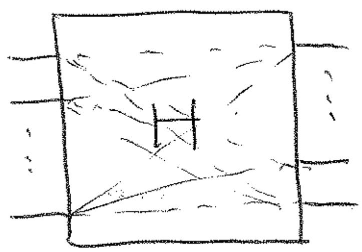
+) global optimization:
- finds the global solution.
- worst_case complexity grows exponentially with problem size.

+) Convex optimization:
- guarantees global solution (or certificate of infeasibility)
- Computation time small ($\simeq \max\{n^3, n^2 m, F\}$ where F is cost of evaluating f_i's first and second derivatives)
- non-heuristic stopping criterion, can guarantee a tolerance gap.
- handles non-differentiable functions as well.

- <u>Example:</u> MIMO capacity maximization

$$\max \ \log\det\left(HQH^{+} + \sigma^2 I\right)$$
$$\text{s.t.} \ \ \mathrm{tr}(Q) \le P.$$

$Q \in \mathbb{R}^{n\times n}$: optimization variable (transmit covariance
$H \in \mathbb{C}^{m\times n}$: channel parameters, given
$\sigma^2 \in \mathbb{R}$: noise power, a scalar constant.

$x(n)$  $y(m)$ $\quad y = Hx + n$

P: total transmit power
$Q = E[x^T x]$.

- This problem is convex, and in fact has an analytical solution.
- Can build a very simple algorithm to solve this problem and find the optimal solution.

+) This course: the goals are for you to be able to
- recognize / formulate problems as convex optimization
- characterize the optimal solution
- develop algorithms and write codes for moderate size problems.

We will cover:
- convex sets, functions, problems
- duality theory for analyzing / characterizing convex problems
- algorithms including (a bit of) performance + complexity