Lecture 23
Robust optimization

- overview
- example
- robust optimization
- classical approaches
- robust SDP and its dual
- what we'll do

Optimization and uncertainty

we've seen many applications of (convex) optimization

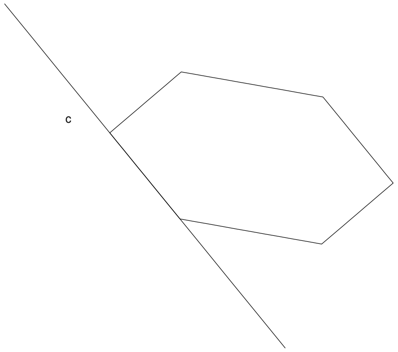
possible pitfalls of numerical optimization

- relies heavily on *precise* numerical knowledge
- hence, optimized solution highly tuned to this knowledge
  (this is a "sensitivity" problem)
- unability to treat "soft" constraints (AI)
- some (most?) practical problems involve **uncertain** data
  (see later where uncertain may come from)

what do we do with uncertain data?

Example: sensitivity of LP

minimize $c^T x$ subject to
$a_i^T x \geq b_i, \; i = 1, \ldots, L.$



in above situation

- solution not necessarily unique
- may be very sensitive to problem data
  (may jump from one vertex to another)

many other "pathologies" can occur (more often that you'd like to think!)

Where does uncertainty come from?

uncertainty due to

- *data collection:* measurement, pre-processing, statistics
- *soft constraints:* partial knowledge
- *model reduction:* approximate modelling of complex phenomena
- *implementation:* discretization, quantization
- *uncertainty about future:* exact value of data not yet known
  (especially true in dynamic problems)

in practice:

errors often large, nonlinear, highly structured (correlated)

Optimization with uncertain data: methods

- **soft:** constraint programming, AI, fuzzy logic
- **infinitesimal:** take into account *sensitivity* of data vs. uncertain parameters (neglect nonlinear, large perturbations)
- **scenarii:** design against a handful of scenarii ("what-if")
- **stochastic:** model uncertainties as random variables ("chance programming")
- **robust:** design against *worst-case* among infinite number of scenarii

a mix is possible (see later)

Robust optimization

Robustness refers to: a given system satisfies a given property **whatever** perturbations in a given set

- robustness **analysis:** evaluate effect of perturbation on given system
- robust **design:** decrease (or bound) this effect

- robustness depends heavily on perturbation set (structure)
- importance of robustness grows with performance requirements

  (remember, performant designs are very non robust!)

Robust optimization: main idea

- start with model of uncertainty for the data, including info on bounds and structure of perturbation
- design against worst-case
  - seek to satisfy the constraints whatever the admissible perturbation
  - optimize the worst-case objective among all robust solutions
- compute bounds on problem via convex programming

Example: robust LP

nominal LP: (data is (A, b, c))

$$\min \ c^T x \quad Ax \leq b$$

assume some data (say, (A, b)) is not exactly known—we only know it within a set:

$$(A, b) \in \mathcal{U}$$

where the *uncertainty set* is known

robust LP:

$$\min \ c^T x \quad \mathbf{A}x \leq \mathbf{b} \text{ for every } (\mathbf{A}, \mathbf{b}) \text{ in } \mathcal{U}$$

in general, NP-hard problem

note however: it is a convex problem!

Special case

Denote by a_i^T the i-th row of A

perturbation model: Assume a_i is only known within the ellipsoid

$$\mathbf{a}_i \in \mathcal{E}_i := \{a_i + E_i\delta_i, \ \ \|\delta_i\|_2 \leq 1\}$$

where a_i, E_i are given

robust LP is

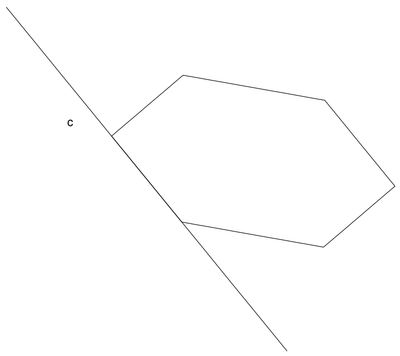
$$\min \ c^T x \ \ \mathbf{a}_i^T x \leq b_i \text{ for every } \mathbf{a}_i \in \mathcal{E}_i, \ \ i = 1, \ldots, m$$

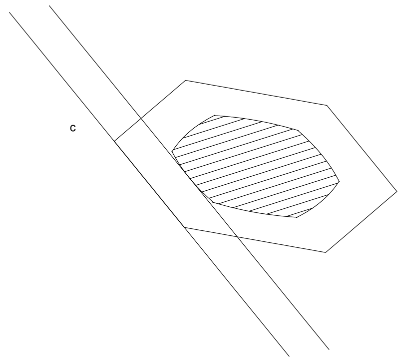
here, robust LP is *equivalent* to a *tractable* convex optimization problem (a SOCP!)

$$\min \ c^T x \ \ a_i^T x + \|E_i x\|_2 \leq b_i, \ \ i = 1, \ldots, m$$

for more complicated perturbation models, problem is not tractable, but we can approximate it

Picture





nominal linear program

- solution not unique
- hence, very sensitive
- low complexity

robust LP
(additive perturbations)

- solution unique
- hence, continuous
- almost same complexity as LP (not true in general)

introduction of uncertainty model *regularizes* the problem

Robust optimization framework

nominal optimization problem:

minimize $f_0(x)$ subject to $f_i(x) \leq 0,\ i = 1, \ldots, m$

assume "data" in problem (*i.e.*, functions f_0, f_i) is not well-known

- assume f_i's are perturbed by an unknown-but-bounded parameter $\delta \in \mathcal{D}$
- denote by $f_i(x, \delta)$ the perturbed value of function f_i

robust counterpart:

$$\text{minimize } \max_{\delta \in \mathcal{D}} f_0(x) \qquad \text{subject to}$$
$$f_i(x, \delta) \leq 0,\ i = 1, \ldots, m, \quad \text{for every } \delta \in \mathcal{D}$$

- new problem has infinite number of constraints
- convex if nominal problem is
- in general, much higher complexity than nominal
- seek approximate it (compute upper bounds)

Known objective is universal

WLOG we may assume objective fcn f_0 is known (*i.e.*, independent of δ)

proof: introduce new variable $t \in \mathbf{R}$, rewrite as:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & f_0(x, \delta) \leq t, \quad f_i(x, \delta) \leq 0 \text{ for every } \delta \in \mathcal{D} \end{array}$$

i.e., new problem has linear objective in variable $z = (x, t)$ and one more robustness constraint

Robust SDP

nominal SDP:

minimize $c^T x$ subject to $F_0 + x_1 F_1 + \cdots + x_m F_m \succeq 0$

- each coefficient matrix F_i symmetric
- $x \in \mathbf{R}^m$ is decision variable
- $F \succeq 0$ means every eigenvalue of symmetric F is non negative
- c is objective vector

robust SDP:

$$\text{minimize } c^T x \text{ subject to } \mathbf{F}_0 + x_1 \mathbf{F}_1 + \cdots + x_m \mathbf{F}_m \succeq 0 \quad \text{for every } (\mathbf{F}_0, \ldots, \mathbf{F}_m) \in \mathcal{U}$$

where

- WLOG objective c is perfectly known
- $\mathcal{U}$ is a set of matrices describing *partial knowledge* about the coefficient matrices

A duality view

Recall alternative (dual) form of SDP:

$$\min c^T x \; Ax = b, \quad x \preceq 0$$

where x represents a matrix variable

Assume the "A" matrix is uncertain:

$$\mathbf{A} \in \mathcal{U},$$

where $\mathcal{U}$ is a given set of matrices

what is robust counterpart?

Because of equality constraints, usually meaningless to ask the constraints to be always satisfied!

Robust counterpart

Dual of previous problem is

$$\max b^T y \quad A^T \star y \preceq c$$

(here $A^T \star y$ is dual map of linear map Ax)

Robust counterpart of dual is

$$\max b^T y \ \mathbf{A}^T \star y \preceq c \text{ for every } \mathbf{A} \in \mathcal{U}$$

We can form a dual of above problem (it's a convex problem!)

$$\min c^T x \ \mathbf{A}x = b \text{ for some } \mathbf{A} \in \mathcal{U}$$

In this form, robust SDP is a (quite general) non convex problem

under technical conditions, no primal-dual gap

What we'll do

– consider a robust SDP

– assume data depends algebraically on unknown-but-bounded parameters

  (bounds can be Euclidean, maximum norm, sign, matrix norm, or a mix)

– model uncertainty using *linear-fractional representations* (LFR) of algebraic functions

– using LFRs, represent robust SDP as a quadratically constrained program (or rank-constrained SDP)

– using Lagrange (or rank) relaxation, get SDP approximation to original problem

– extgend robust SDP to more general robust optimization problems

next: describe semi-algebraic matrix set $\mathcal{U}$ in a matrix-based way