Convex Optimization Problems

Prof. Daniel P. Palomar

The Hong Kong University of Science and Technology (HKUST)

MAFS6010R - Portfolio Optimization with R
MSc in Financial Mathematics
Fall 2019-20, HKUST, Hong Kong

Outline

1. Optimization Problems
2. Convex Optimization
3. Quasi-Convex Optimization
4. Classes of Convex Problems: LP, QP, SOCP, SDP
5. Multicriterion Optimization (Pareto Optimality)
6. Solvers

Outline

1. Optimization Problems
2. Convex Optimization
3. Quasi-Convex Optimization
4. Classes of Convex Problems: LP, QP, SOCP, SDP
5. Multicriterion Optimization (Pareto Optimality)
6. Solvers

Optimization Problem in Standard Form

General optimization problem in standard form:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \ldots, m \\ & h_i(x) = 0 & i = 1, \ldots, p \end{array}$$

where

$x = (x_1, \ldots, x_n)$ is the optimization variable

$f_0 : \mathbf{R}^n \longrightarrow \mathbf{R}$ is the objective function

$f_i : \mathbf{R}^n \longrightarrow \mathbf{R}, \quad i = 1, \ldots, m$ are inequality constraint functions

$h_i : \mathbf{R}^n \longrightarrow \mathbf{R}, \quad i = 1, \ldots, p$ are equality constraint functions.

- **Goal**: find an optimal solution $x^\star$ that minimizes f_0 while satisfying all the constraints.

Optimization Problem in Standard Form

- **Feasibility**:
  - a point $x \in \mathrm{dom}\, f_0$ is feasible if it satisfies all the constraints and infeasible otherwise
  - a problem is feasible if it has at least one feasible point and infeasible otherwise.
- **Optimal value**:

$$p^{\star} = \inf \left\{ f_0\left(x\right) \mid f_i\left(x\right) \leq 0,\ i = 1, \ldots, m,\ h_i\left(x\right) = 0,\ i = 1, \ldots, p \right\}$$

$$p^{\star} = \infty \text{ if problem infeasible (no } x \text{ satisfies the constraints)}$$
$$p^{\star} = -\infty \text{ if problem unbounded below.}$$

- **Optimal solution**: $x^{\star}$ such that $f\left(x^{\star}\right) = p^{\star}$ (and $x^{\star}$ feasible).

Global and Local Optimality

- A feasible x is **optimal** if $f_0(x) = p^\star$; X_{opt} is the set of optimal points.
- A feasible x is **locally optimal** if is optimal within a ball

Examples:

- $f_0(x) = 1/x$, $\text{dom}\, f_0 = \mathbf{R}_{++}$: $p^\star = 0$, no optimal point
- $f_0(x) = x^3 - 3x$: $p^\star = -\infty$, local optimum at $x = 1$.

Implicit Constraints

- The standard form optimization problem has an explicit constraint:

$$x \in \mathcal{D} = \bigcap_{i=0}^{m} \operatorname{dom} f_i \ \cap \ \bigcap_{i=1}^{p} \operatorname{dom} f_i$$

- $\mathcal{D}$ is the domain of the problem
- The constraints $f_i(x) \leq 0, \ h_i(x) = 0$ are the explicit constraints
- A problem is unconstrained if it has no explicit constraints
- Example:

$$\underset{x}{\text{minimize}} \quad \log\left(b - a^T x\right)$$

is an unconstrained problem with implicit constraint $b > a^T x$.

Feasibility Problem

- Sometimes, we don't really want to minimize any objective, just to find a feasible point:

$$\begin{array}{lll} \underset{x}{\text{find}} & x & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \ldots, m \\ & h_i(x) = 0 & i = 1, \ldots, p \end{array}$$

- This feasibility problem can be considered as a special case of a general problem:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & 0 & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \ldots, m \\ & h_i(x) = 0 & i = 1, \ldots, p \end{array}$$

where $p^\star = 0$ if constraints are feasible and $p^\star = \infty$ otherwise.

Outline

1. Optimization Problems
2. Convex Optimization
3. Quasi-Convex Optimization
4. Classes of Convex Problems: LP, QP, SOCP, SDP
5. Multicriterion Optimization (Pareto Optimality)
6. Solvers

Convex Optimization Problem

Convex optimization problem in standard form:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \ldots, m \\ & Ax = b & \end{array}$$

where $f_0, f_1, \ldots, f_m$ are convex and equality constraints are affine.

- **Local and global optima**: any locally optimal point of a convex problem is globally optimal.
- Most problems are not convex when formulated.
- Reformulating a problem in convex form is an art, there is no systematic way.

Example

- The following problem is nonconvex (why not?):

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x_1^2 + x_2^2 \\ \text{subject to} & x_1 / \left(1 + x_2^2\right) \leq 0 \\ & \left(x_1 + x_2\right)^2 = 0 \end{array}$$

- The objective is convex.
- The equality constraint function is not affine; however, we can rewrite it as $x_1 = -x_2$ which is then a linear equality constraint.
- The inequality constraint function is not convex; however, we can rewrite it as $x_1 \leq 0$ which again is linear.
- We can rewrite it as

$$\begin{array}{ll} \underset{x}{\text{minimize}} & x_1^2 + x_2^2 \\ \text{subject to} & x_1 \leq 0 \\ & x_1 = -x_2 \end{array}$$

Equivalent Reformulations

- **Eliminating/introducing equality constraints**:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0 \qquad i = 1, \ldots, m \\ & Ax = b \end{array}$$

is equivalent to

$$\begin{array}{ll} \underset{z}{\text{minimize}} & f_0(Fz + x_0) \\ \text{subject to} & f_i(Fz + x_0) \leq 0 \qquad i = 1, \ldots, m \end{array}$$

where F and x_0 are such that $Ax = b \Longleftrightarrow x = Fz + x_0$ for some z.

Equivalent Reformulations

- **Introducing slack variables for linear inequalities**:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & a_i^T x \leq b_i \qquad i = 1, \ldots, m \end{array}$$

is equivalent to

$$\begin{array}{ll} \underset{x,s}{\text{minimize}} & f_0(x) \\ \text{subject to} & a_i^T x + s_i = b_i \qquad i = 1, \ldots, m \\ & s_i \geq 0 \end{array}$$

Equivalent Reformulations

- **Epigraph form**: a standard form convex problem is equivalent to

$$\begin{array}{ll} \underset{x,t}{\text{minimize}} & t \\ \text{subject to} & f_0(x) - t \leq 0 \\ & f_i(x) \leq 0 \qquad i = 1, \ldots, m \\ & Ax = b \end{array}$$

- **Minimizing over some variables**:

$$\begin{array}{ll} \underset{x,y}{\text{minimize}} & f_0(x, y) \\ \text{subject to} & f_i(x) \leq 0 \qquad i = 1, \ldots, m \end{array}$$

is equivalent to

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \tilde{f}_0(x) \\ \text{subject to} & f_i(x) \leq 0 \qquad i = 1, \ldots, m \end{array}$$

where $\tilde{f}_0(x) = \inf_y f_0(x, y)$.

Outline

1 Optimization Problems

2 Convex Optimization

3 Quasi-Convex Optimization

4 Classes of Convex Problems: LP, QP, SOCP, SDP

5 Multicriterion Optimization (Pareto Optimality)

6 Solvers

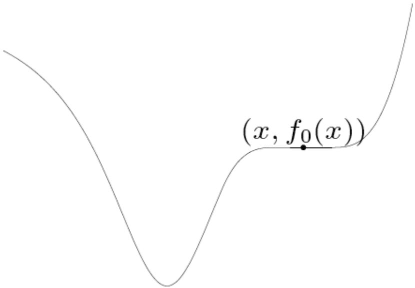
Quasiconvex Optimization

Quasi-convex optimization problem in standard form:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 0 & i = 1, \ldots, m \\ & Ax = b & \end{array}$$

where $f_0 : \mathbf{R}^n \longrightarrow \mathbf{R}$ is quasiconvex and $f_1, \ldots, f_m$ are convex.

- Observe that it can have locally optimal points that are not (globally) optimal:



Quasiconvex Optimization

- **Convex representation** of sublevel sets of a quasiconvex function f_0: there exists a family of convex functions $\phi_t(x)$ for fixed t such that
$$f_0(x) \leq t \quad \Longleftrightarrow \quad \phi_t(x) \leq 0.$$
- **Example**:
$$f_0(x) = \frac{p(x)}{q(x)}$$
with p convex, q concave, and $p(x) \geq 0$, $q(x) > 0$ on $\mathrm{dom}\, f_0$. We can choose:
$$\phi_t(x) = p(x) - tq(x)$$
  - for $t \geq 0$, $\phi_t(x)$ is convex in x
  - $p(x)/q(x) \leq t$ if and only if $\phi_t(x) \leq 0$.

Quasiconvex Optimization

Solving a quasiconvex problem via convex feasibility problems: the idea is to solve the epigraph form of the problem with a sandwich technique in t:

- for fixed t the epigraph form of the original problem reduces to a feasibility convex problem

$$\phi_t(x) \leq 0, \quad f_i(x) \leq 0 \; \forall i, \quad Ax \leq b$$

- if t is too small, the feasibility problem will be infeasible
- if t is too large, the feasibility problem will be feasible
- start with upper and lower bounds on t (termed u and l, resp.) and use a sandwitch technique (bisection method): at each iteration use $t = (l + u)/2$ and update the bounds according to the feasibility/infeasibility of the problem.

Outline

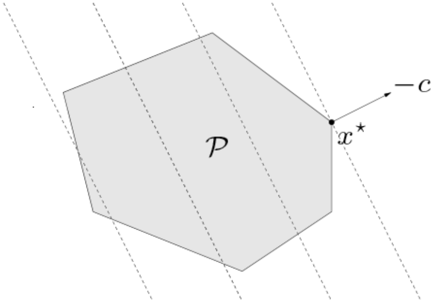
1. Optimization Problems
2. Convex Optimization
3. Quasi-Convex Optimization
4. Classes of Convex Problems: LP, QP, SOCP, SDP
5. Multicriterion Optimization (Pareto Optimality)
6. Solvers

Linear Programming (LP)

LP:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x + d \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

- Convex problem: affine objective and constraint functions.
- Feasible set is a polyhedron:



ℓ_1- and ℓ_∞- Norm Problems as LPs

ℓ_∞-norm minimization:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \|x\|_\infty \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

is equivalent to the LP

$$\begin{array}{ll} \underset{t,x}{\text{minimize}} & t \\ \text{subject to} & -t\mathbf{1} \leq x \leq t\mathbf{1} \\ & Gx \leq h \\ & Ax = b. \end{array}$$

ℓ_1- and ℓ_∞- Norm Problems as LPs

ℓ_1**-norm minimization**:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \|x\|_1 \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

is equivalent to the LP

$$\begin{array}{ll} \underset{t,x}{\text{minimize}} & \sum_i t_i \\ \text{subject to} & -t \leq x \leq t \\ & Gx \leq h \\ & Ax = b. \end{array}$$

Linear-Fractional Programming

Linear-fractional programming:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \left(c^T x + d\right) / \left(e^T x + f\right) \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

with $\mathrm{dom}\, f_0 = \left\{x \mid e^T x + f > 0\right\}$.

- It is a quasiconvex optimization problem (solved by bisection).
- Interestingly, the following LP is equivalent:

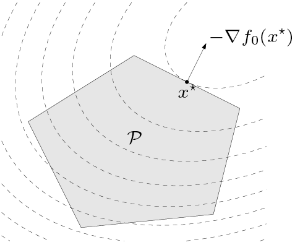
$$\begin{array}{ll} \underset{y,z}{\text{minimize}} & c^T y + dz \\ \text{subject to} & Gy \leq hz \\ & Ay = bz \\ & e^T y + fz = 1 \\ & z \geq 0 \end{array}$$

Quadratic Programming (QP)

Quadratic programming:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & (1/2)\, x^T P x + q^T x + r \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array}$$

- Convex problem (assuming $P \in \mathbf{S}^n \succeq 0$): convex quadratic objective and affine constraint functions.
- Minimization of a convex quadratic function over a polyhedron:



Quadratically Constrained QP (QCQP)

Quadratically constrained QP:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & (1/2)\, x^T P_0 x + q_0^T x + r_0 & \\ \text{subject to} & (1/2)\, x^T P_i x + q_i^T x + r_i \leq 0 & i = 1, \ldots, m \\ & Ax = b & \end{array}$$

- Convex problem (assuming $P_i \in \mathbf{S}^n \succeq 0$): convex quadratic objective and constraint functions.

Second-Order Cone Programming (SOCP)

Second-order cone programming:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & f^T x & \\ \text{subject to} & \|A_i x + b_i\| \leq c_i^T x + d_i & i = 1, \ldots, m \\ & Fx = g & \end{array}$$

- Convex problem: linear objective and second-order cone constraints
- For A_i row vector, it reduces to an LP.
- For $c_i = 0$, it reduces to a QCQP.
- More general than QCQP and LP.

Semidefinite Programming (SDP)

SDP:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & x_1 F_1 + x_2 F_2 + \cdots + x_n F_n \preceq G \\ & Ax = b \end{array}$$

- Inequality constraint is called linear matrix inequality (LMI).
- Convex problem: linear objective and linear matrix inequality (LMI) constraints.
- Observe that multiple LMI constraints can always be written as a single one.

Semidefinite Programming (SDP)

- **LP and equivalent SDP**:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & Ax \leq b \end{array}$$

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & \text{diag}\,(Ax - b) \preceq 0 \end{array}$$

- **SOCP and equivalent SDP**:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f^T x \\ \text{subject to} & \|A_i x + b_i\| \leq c_i^T x + d_i, \quad i = 1, \ldots, m \end{array}$$

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f^T x \\ \text{subject to} & \begin{bmatrix} \left(c_i^T x + d_i\right) I & A_i x + b_i \\ \left(A_i x + b_i\right)^T & c_i^T x + d_i \end{bmatrix} \succeq 0, \quad i = 1, \ldots, m \end{array}$$

Semidefinite Programming (SDP)

- **Eigenvalue minimization**:

$$\underset{x}{\text{minimize}} \quad \lambda_{\max}(A(x))$$

where $A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n$, is equivalent to SDP

$$\begin{array}{ll} \underset{x}{\text{minimize}} & t \\ \text{subject to} & A(x) \preceq tI \end{array}$$

- It follows from

$$\lambda_{\max}(A(x)) \leq t \iff A(x) \preceq tI$$

Outline

1. Optimization Problems
2. Convex Optimization
3. Quasi-Convex Optimization
4. Classes of Convex Problems: LP, QP, SOCP, SDP
5. Multicriterion Optimization (Pareto Optimality)
6. Solvers

Scalarization for Multicriterion Problems

- Consider a multicriterion optimization with q different objectives:

$$f_0(x) = (F_1(x), \ldots, F_q(x)).$$

To find Pareto optimal points, minimize the positive weighted sum:

$$\lambda^T f_0(x) = \lambda_1 F_1(x) + \cdots + \lambda_q F_q(x).$$

- Example: regularized least-squares:

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|_2^2 + \gamma \|x\|_2^2 \; .$$

References on Convex Problems

Chapter 4 of

- Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge University Press, 2004.

https://web.stanford.edu/~boyd/cvxbook/

Outline

1. Optimization Problems
2. Convex Optimization
3. Quasi-Convex Optimization
4. Classes of Convex Problems: LP, QP, SOCP, SDP
5. Multicriterion Optimization (Pareto Optimality)
6. Solvers

Solvers

- A solver is an engine for solving a particular type of mathematical problem, such as a convex program.
- Every programming language (e.g., Matlab, Octave, R, Python, C, C++) has a long list of available solvers to choose from.
- Solvers typically handle only a certain class of problems, such as LPs, QPs, SOCPs, SDPs, or GPs.
- They also require that problems be expressed in a standard form.
- Most problems do not immediately present themselves in a standard form, so they must be transformed into standard form.

Solver Example: Matlab's `linprog`

- A program for solving LPs:

```
x = linprog( c, A, b, A_eq, B_eq, l, u )
```

- Problems must be expressed in the following standard form:

$$
\begin{array}{ll}
\underset{\mathbf{x}}{\text{minimize}} & \mathbf{c}^T\mathbf{x} \\
\text{subject to} & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
 & \mathbf{A}_{\text{eq}}\mathbf{x} = \mathbf{b}_{\text{eq}} \\
 & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}
\end{array}
$$

Conversion to Standard Form: Common Tricks

- Representing free variables as the difference of nonnegative variables:

$$x \text{ free} \quad \Longrightarrow \quad x_+ - x_-, \; x_+ \geq 0, \; x_- \geq 0$$

- Elimininating inequality constraints using slack variables:

$$\mathbf{a}^T\mathbf{x} \leq b \quad \Longrightarrow \quad \mathbf{a}^T\mathbf{x} + s = b, \; s \geq 0$$

- Splitting equality constraints into inequalities:

$$\mathbf{a}^T\mathbf{x} = b \quad \Longrightarrow \quad \mathbf{a}^T\mathbf{x} \leq b, \; \mathbf{a}^T\mathbf{x} \geq b$$

Solver Example: SeDuMi

- A program for solving LPs, SOCPs, SDPs, and related problems:

```
x = sedumi( A, b, c, K )
```

- Solves problems of the form:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathcal{K} \triangleq \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_L \end{array}$$

  where each set $\mathcal{K}_i \subseteq \mathbf{R}^{n_i}$, $i = 1, 2, \ldots, L$ is chosen from a very short list of cones.

- The Matlab variable `K` gives the number, types, and dimensions of the cones $\mathcal{K}_i$.

Solver Example: SeDuMi

Cones supported by SeDuMi:

- free variables: $\mathbf{R}^{n_i}$
- a nonnegative orthant: $\mathbf{R}_+^{n_i}$ (for linear inequalities)
- a real or complex second-order cone:

$$\mathbf{Q}^n \triangleq \{(\mathbf{x}, y) \in \mathbf{R}^n \times \mathbf{R} \mid \|\mathbf{x}\|_2 \leq y\}$$
$$\mathbf{Q}_c^n \triangleq \{(\mathbf{x}, y) \in \mathbf{C}^n \times \mathbf{R} \mid \|\mathbf{x}\|_2 \leq y\}$$

- a real or complex semidefinite cone:

$$\mathbf{S}_+^n \triangleq \left\{\mathbf{X} \in \mathbf{R}^{n \times n} \mid \mathbf{X} = \mathbf{X}^T,\ \mathbf{X} \succeq \mathbf{0}\right\}$$
$$\mathbf{H}_+^n \triangleq \left\{\mathbf{X} \in \mathbf{C}^{n \times n} \mid \mathbf{X} = \mathbf{X}^H,\ \mathbf{X} \succeq \mathbf{0}\right\}$$

The cones must be arranged in this order, i.e., the free variables first, then the nonnegative orthants, then the second-order cones, then the semidefinite cones.

Example: Norm Approximation

Consider the norm approximation problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{A}\mathbf{x} - \mathbf{b}\|$$

- An optimal value $\mathbf{x}^{\star}$ minimizes the residuals

  $$r_k = \mathbf{a}_k^T \mathbf{x} - b_k, \;\; k = 1, 2, \ldots, m$$

  according to the measure defined by the norm $\|\cdot\|$
- Obviously, the value of $\mathbf{x}^{\star}$ depends significantly upon the choice of that norm...
- ... and so does the process of conversion to standard form

Example: Euclidean or ℓ_2-Norm

Norm approximation problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 = \left(\sum_{k=1}^{m} \left(\mathbf{a}_k^T \mathbf{x} - b_k\right)^2\right)^{1/2}$$

- No need to use any solver here: this is a least squares (LS) problem, with an analytic solution:

$$\mathbf{x}^\star = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

- In Matlab or Octave, a single command computes the solution:

```
>> x = A \ b
```

- Similarly, in R:

```
>> x = solve(A, b)
```

Example: Chebyshev or ℓ_∞-Norm

Norm approximation problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_\infty = \underset{\mathbf{x}}{\text{minimize}} \max_{1\leq k\leq m} \left|\mathbf{a}_k^T\mathbf{x}-b_k\right|$$

- This can be expressed as a linear program:

$$\begin{array}{ll} \underset{\mathbf{x},t}{\text{minimize}} & t \\ \text{subject to} & -t\mathbf{1} \leq \mathbf{A}\mathbf{x}-\mathbf{b} \leq t\mathbf{1} \end{array}$$

  or, equivalently,

$$\begin{array}{ll} \underset{\mathbf{x},t}{\text{minimize}} & \begin{bmatrix} \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} \mathbf{A} & -\mathbf{1} \\ -\mathbf{A} & -\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ t \end{bmatrix} \leq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \end{array}$$

Example: Chebyshev or ℓ_∞-Norm

- Recall the final formulation:

$$\begin{array}{ll} \underset{\mathbf{x},t}{\text{minimize}} & \left[\begin{array}{cc} \mathbf{0}^T & 1 \end{array}\right] \left[\begin{array}{c} \mathbf{x} \\ t \end{array}\right] \\ \text{subject to} & \left[\begin{array}{cc} \mathbf{A} & -\mathbf{1} \\ -\mathbf{A} & -\mathbf{1} \end{array}\right] \left[\begin{array}{c} \mathbf{x} \\ t \end{array}\right] \leq \left[\begin{array}{c} \mathbf{b} \\ -\mathbf{b} \end{array}\right] \end{array}$$

- Matlab's `linprog` call:

```
>> xt = linprog( [zeros(n,1); 1], ...
                 [A,-ones(m,1); -A,-ones(m,1)], ...
                 [b; -b] )
>> x = xt(1:n)
```

Example: Manhattan or ℓ_1-Norm

Norm approximation problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_1 = \underset{\mathbf{x}}{\text{minimize}} \sum_{k=1}^{m} \left|\mathbf{a}_k^T\mathbf{x}-b_k\right|$$

- This can be expressed as a linear program:

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{t}}{\text{minimize}} & \mathbf{1}^T\mathbf{t} \\ \text{subject to} & -\mathbf{t} \leq \mathbf{A}\mathbf{x}-\mathbf{b} \leq \mathbf{t} \end{array}$$

or, equivalently,

$$\begin{array}{ll} \underset{\mathbf{x},t}{\text{minimize}} & \begin{bmatrix} \mathbf{0}^T & \mathbf{1}^T \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{t} \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} \mathbf{A} & -\mathbf{I} \\ -\mathbf{A} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{t} \end{bmatrix} \leq \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \end{array}$$

Example: Manhattan or ℓ_1-Norm

- Recall the final formulation:

$$\begin{array}{ll}\underset{\mathbf{x},t}{\text{minimize}} & \left[\begin{array}{cc}\mathbf{0}^T & \mathbf{1}^T\end{array}\right]\left[\begin{array}{c}\mathbf{x}\\ \mathbf{t}\end{array}\right]\\ \text{subject to} & \left[\begin{array}{cc}\mathbf{A} & -\mathbf{I}\\ -\mathbf{A} & -\mathbf{I}\end{array}\right]\left[\begin{array}{c}\mathbf{x}\\ \mathbf{t}\end{array}\right]\leq\left[\begin{array}{c}\mathbf{b}\\ -\mathbf{b}\end{array}\right]\end{array}$$

- Matlab's `linprog` call:

```
>> xt = linprog( [zeros(n,1); ones(n,1)], ...
                 [A,-eye(m,1); -A,-eye(m,1)], ...
                 [b; -b] )
>> x = xt(1:n)
```

Example: Constrained Euclidean or ℓ_2-Norm

Constrained norm approximation problem:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & \|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2 \\ \text{subject to} & \mathbf{C}\mathbf{x}=\mathbf{d} \\ & \mathbf{l}\leq\mathbf{x}\leq\mathbf{u} \end{array}$$

- This is not a least squares problem, but it is QP and an SOCP.
- This can be expressed as

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{y},t,\mathbf{s}_l,\mathbf{s}_u}{\text{minimize}} & t \\ \text{subject to} & \mathbf{A}\mathbf{x}-\mathbf{b}=\mathbf{y} \\ & \mathbf{C}\mathbf{x}=\mathbf{d} \\ & \mathbf{x}-\mathbf{s}_l=\mathbf{l} \\ & \mathbf{x}+\mathbf{s}_u=\mathbf{u} \\ & \mathbf{s}_l,\mathbf{s}_u\geq\mathbf{0} \\ & \|\mathbf{y}\|_2\leq t \end{array}$$

Example: Constrained Euclidean or ℓ_2-Norm

- Equivalently:

$$\begin{array}{ll}\underset{\mathbf{x},\mathbf{y},t,\mathbf{s}_l,\mathbf{s}_u}{\text{minimize}} & \begin{bmatrix} \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & \mathbf{0}^T & 1 \end{bmatrix} \bar{\mathbf{x}} \\ \text{subject to} & \begin{bmatrix} \mathbf{A} & & & -\mathbf{I} & \\ \mathbf{C} & & & & \\ \mathbf{I} & -\mathbf{I} & & & \\ \mathbf{I} & & \mathbf{I} & & \end{bmatrix} \bar{\mathbf{x}} \leq \begin{bmatrix} \mathbf{b} \\ \mathbf{d} \\ \mathbf{l} \\ \mathbf{u} \end{bmatrix} \\ & \bar{\mathbf{x}} \in \mathbf{R}^n \times \mathbf{R}_+^n \times \mathbf{R}_+^n \times \mathbf{Q}^m \end{array}$$

- SeDuMi call:

```
>> AA = [ A,    zeros(m,n), zeros(m,n),  -eye(m),    0;
          C,    zeros(p,n), zeros(p,n)   zeros(p,n), 0;
          eye(n), -eye(n),  zeros(n,n), zeros(n,n), 0;
          eye(n), zeros(n,n),  eye(n),   zeros(n,n), 0 ]
>> bb = [ b; d; l; u ]
>> cc = [ zeros(3*n+m,1); 1 ]
>> K.f = n; K.l = 2*n; K.q = m + 1;
>> xsyz = sedumi( AA, bb, cc, K )
>> x = xsyz(1:n)
```

Modeling Frameworks: cvx

- A modeling framework simplifies the use of a numerical technology by shielding the user from the underlying mathematical details.
- Examples: SDPSOL, YALMIP, cvx, etc.
- cvx is designed to support convex optimization or, more specifically, disciplined convex programming (available in Matlab, Python, R, Julia, Octave, etc.)
- People don't simply write optimization problems and hope that they are convex; instead they draw from a "mental library"of functions and sets with known convexity properties and combine them in ways that convex analysis guarantees will produce convex results.
- Disciplined convex programming formalizes this methodology.
- Links:
  - cvx: http://cvxr.com
  - cvx user guide: http://web.cvxr.com/cvx/doc/CVX.pdf
  - cvx for R (cvxr): https://github.com/cvxgrp/CVXR

Thanks

For more information visit:

https://www.danielppalomar.com