Tutorial on semidefinite programming (SDP)
Uniandes/Externado, Spring 2006

Last time:

- SDP: a generalization of linear programming (LP)
- Examples of SDP applications
- SDP duality, complementarity

Today:

- Second-order programming (SOCP)
- Examples of SOCP applications
- SOCP/LP/SDP conic programming
- Solvers: `SeDuMi, SDPT3`

Recall

SDP primal-dual pair

$$
(P)\quad \begin{array}{ll} \min & C \bullet X \\ \text{s.t.} & \mathcal{A}X = b \\ & X \succeq 0, \end{array} \qquad (D)\quad \begin{array}{ll} \max & b^{\mathsf{T}} y \\ \text{s.t.} & \mathcal{A}^* y + S = C \\ & S \succeq 0. \end{array}
$$

Here $\mathcal{A} \in L(\mathbf{S}^n, \mathbf{R}^m)$, $C \in \mathbf{S}^n$, $b \in \mathbf{R}^m$ are given.

Observe: LP $\subseteq$ SDP (why?)

Multivariate SDP

$$
\begin{array}{ll} \min & C_1 \bullet X_1 + \cdots + C_k \bullet X_r \\ \text{s.t.} & A_{11} \bullet X_1 + \cdots + A_{1r} \bullet X_r = b_1 \\ & \vdots \\ & A_{m1} \bullet X_1 + \cdots + A_{mr} \bullet X_r = b_m \\ & X_1, \ldots, X_r \succeq 0, \end{array}
$$

This is as general as a single-var SDP. (Why?)

Write above as

$$
\begin{array}{ll} \min & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}X = b \\ & X = (X_1, \ldots, X_r) \\ & X_1, \ldots, X_r \succeq 0, \end{array}
$$

For $b = (b_1, \ldots, b_m)$, $C = (C_1, \ldots, C_r)$ and $\mathcal{A} = \begin{bmatrix} A_{11} & \cdots & A_{1r} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mr} \end{bmatrix}$

Both LP and SDP are special cases of linear conic-programming

$$
\begin{array}{ll} \min & \langle c, x \rangle \\ \text{s.t.} & Ax = b \\ & x \in K \end{array}
$$

where E, Y Euclidean spaces, $A \in L(E, Y)$, $b \in Y$, $c \in E$, and $K \subseteq E$ is a closed, convex cone.

Second-order cone programming

Second-order cone (a.k.a. Lorentz cone):

$$\mathcal{Q}_n := \left\{ x = \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \in \mathbf{R}^n : x_0 \geq \|\bar{x}\| \right\}.$$

Write $x \succeq_{\mathcal{Q}_n} 0$ for $x \in \mathcal{Q}_n$.

Observe: $x = \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \in \mathcal{Q}_n$ iff $\begin{bmatrix} x_0 & \bar{x}^\mathsf{T} \\ \bar{x} & x_0 I \end{bmatrix} \succeq 0$.

Second-order cone programming (SOCP)

$$\begin{array}{ll} \min & c^\mathsf{T} x \\ \text{s.t.} & Ax = b \\ & x = (x_1, \ldots, x_r) \\ & x_i \succeq_{\mathcal{Q}_{n_i}} 0, \end{array}$$

Case $r = 1$ can be solved in closed-form. Interesting case is $r \geq 2$.

For convenience put $\mathcal{Q} := \mathcal{Q}_{n_1} \times \cdots \times \mathcal{Q}_{n_r}$. Write $x \succeq_{\mathcal{Q}} 0$ for $x \in \mathcal{Q}$.

Observe: LP $\subseteq$ SOCP $\subseteq$ SDP.

SOCP primal and dual forms

The dual of

$$\begin{array}{ll} \min & c^\mathsf{T} x \\ \text{s.t.} & Ax = b \\ & x \succeq_{\mathcal{Q}} 0, \end{array}$$

is

$$\begin{array}{ll} \max & b^\mathsf{T} y \\ \text{s.t.} & A^\mathsf{T} y \preceq_{\mathcal{Q}} c, \end{array}$$

which we will sometimes write as

$$\begin{array}{ll} \max & b^\mathsf{T} y \\ \text{s.t.} & A^\mathsf{T} y + s = c \\ & s \succeq_{\mathcal{Q}} 0. \end{array}$$

<u>Examples of SOCP</u>

Example 1 (norm minimization) Suppose $b_1, \ldots, b_r \in \mathbf{R}^d$ are given, and want to solve

$$\min_y \max_{i=1,\ldots,r} \|y - b_i\|$$

Can reformulate as

$$\begin{array}{ll} \min & t \\ & \|y - b_i\| \leq t \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \max & -t \\ & \begin{bmatrix} 0 \\ b_i \end{bmatrix} - \begin{bmatrix} -t \\ y \end{bmatrix} \succeq_{\mathcal{Q}_{d+1}} 0, \end{array}$$

which is a second-order program.

Can proceed similarly for

$$\min_y \sum_{i=1}^{r} \|y - b_i\|.$$

Example 2 (robust least-squares): Let $\mathcal{U} \subseteq \mathbf{R}^{d\times r}$, $q \in \mathbf{R}^d$, where $d > r$. Want to solve

$$\min_v \max_{P\in\mathcal{U}} \|Pv - q\|$$

Assume the uncertainty set $\mathcal{U}$ is ellipsoidal, e.g.,

$$\mathcal{U} = \{P : \|P - \bar{P}\| \leq \rho\}.$$

Thus for a given v we get

$$\max_{P\in\mathcal{U}} \|Pv - q\| = \|\bar{P}v - q\| + \rho\|v\|.$$

Hence the robust least-squares problem can be formulated as

$$\min_v(\|\bar{P}v - q\| + \rho\|v\|),$$

which in turn can be written as a second-order program.

Example 3 (robust LP): Can also apply the same to robust linear programming: if $a \in \mathcal{U} = \{a : \|a - \bar{a}\| \leq \rho\} \subseteq \mathbf{R}^n$ then

$$\max_{a\in\mathcal{U}} (a^\top y - b) \leq 0$$

iff

$$\bar{a}^\top y + \rho\|y\| - b \leq 0.$$

Thus if some LP constraints and/or objective are uncertain, can make them robust via SOCP.

For instance if $a_0 \in \mathcal{U}_0$, $a_1 \in \mathcal{U}_1$, $a_2 \in \mathcal{U}_2$, where $\mathcal{U}_i = \{a_i : \|a_i - \bar{a}_i\| \leq \rho_i\}$, $i = 0, 1, 2$ then the robust version of

$$\begin{array}{ll} \text{min} & a_0^\top y \\ \text{s.t.} & a_1^\top y \leq b_1 \\ & a_2^\top y \leq b_2 \end{array}$$

is

$$\begin{array}{ll} \text{min} & a_0^\top y + \rho_0\|y\| \\ \text{s.t.} & a_1^\top y + \rho_1\|y\| \leq b_1 \\ & a_2^\top y + \rho_2\|y\| \leq b_2, \end{array}$$

which can be rewritten as

$$\begin{array}{ll} \text{min} & a_0^\top y + \rho_0 t \\ \text{s.t.} & a_1^\top y + \rho_1 t \leq b_1 \\ & a_2^\top y + \rho_2 t \leq b_2 \\ & \|y\| \leq t. \end{array}$$

Example 4 (convex quadratic programming). Assume $Q = LL^\top \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, $t \in \mathbf{R}$. Then

$$x^\top Qx + q^\top x + \ell \leq 0$$

can be recast as

$$\left\| \begin{bmatrix} L^\top x \\ \frac{1+q^\top x+\ell}{2} \end{bmatrix} \right\| \leq \frac{1 - q^\top x - \ell}{2}.$$

Therefore a quadratic problem of the form

$$\begin{array}{ll} \text{min} & x^\top Q_0 x + q_0^\top x \\ \text{s.t.} & x^\top Q_i x + q_i^\top x + \ell_i \leq 0,\ i = 1, \ldots, r \end{array}$$

can be recast as an SOCP if each $Q_i \succeq 0$.

Hyperbolic inequalities

$$\|x\| \le st \Leftrightarrow \left\| \begin{bmatrix} 2x \\ s-t \end{bmatrix} \right\| \le s+t$$

Example 5.

$$\begin{array}{ll} \text{min} & \sum_{i=1}^{r} \frac{1}{a_i^\top x + b_i} \\ \text{s.t.} & a_i^\top x + b_i > 0,\ i = 1,\dots,r \end{array}$$

can be reformulated as

$$\begin{array}{ll} \text{min} & \sum_{i=1}^{r} t_i \\ \text{s.t.} & \left\| \begin{bmatrix} 2 \\ a_i^\top x + b_i - t_i \end{bmatrix} \right\| \le a_i^\top x + b_i + t_i,\ i = 1,\dots,r. \end{array}$$

As in SDP, need a bit more for strong duality.

Thm (strong duality). Assume (P) and (D) are strongly feasible. Then both (P) and (D) have optimal solutions. Furthermore, x and (y,s) are optimal sols to (P) and (D) respectively iff

$$b^\top y = c^\top x \ \Leftrightarrow\ x^\top s = 0.$$

SOCP Duality

Consider the SDP primal-dual pair.

$$\text{(P)}\quad \begin{array}{ll} \text{min} & c^\top x \\ \text{s.t.} & Ax = b \\ & x \succeq_{\mathcal{Q}} 0, \end{array} \qquad \text{(D)}\quad \begin{array}{ll} \text{max} & b^\top y \\ \text{s.t.} & A^\top y + s = c \\ & s \succeq_{\mathcal{Q}} 0. \end{array}$$

Prop (weak duality). If x is (P)-feas, and (y,s) is (D)-feasible then $b^\top y \le c^\top x$.

Something like eigenvalues/eigenvectors for SOCP

For simplicity assume $\mathcal{Q} = \mathcal{Q}_n$ (only one second-order cone).

For $x \in \mathbf{R}^n$ define the following "eigenvalues"

$$\lambda_1(x) := x_0 + \|\bar{x}\|,\ \lambda_2(x) := x_0 - \|\bar{x}\|,$$

and the following "spectral decomposition":

$$x = \lambda_1(x) v_1 + \lambda_2(x) v_2$$

for the orthogonal vectors

$$v_1 = \frac{1}{2}\begin{bmatrix} 1 \\ \bar{x}/\|\bar{x}\| \end{bmatrix},\ v_2 = \frac{1}{2}\begin{bmatrix} 1 \\ -\bar{x}/\|\bar{x}\| \end{bmatrix}.$$

SOCP Complementarity

Prop (complementarity). Let $x, s \succeq_{\mathcal{Q}} 0$. Then

$$x^\top s = 0 \Leftrightarrow x^\top s = 0 \text{ and } x_0\bar{s} + s_0\bar{x} = 0$$

The latter in turn holds iff x, s satisfy

$$x = \lambda_1 v_1 + \lambda_2 v_2, \; s = \omega_1 v_1 + \omega_2 v_2$$

where v_1, v_2 are orthogonal vectors, each of the form $\frac{1}{2}\begin{bmatrix} 1 \\ \bar{v}/\|\bar{v}\| \end{bmatrix}$ and $\lambda_i \omega_i = 0, \; i = 1, 2$.

For $x, s \in \mathbf{R}^n$ define

$$x \circ s = \begin{bmatrix} x^\top s \\ x_0\bar{s} + s_0\bar{x} \end{bmatrix}.$$

Hence under the strong feasibility assumptions can recast (P) and (D) as

$$\begin{aligned} A^\top y + s &= c \\ Ax &= b \\ x \circ s &= 0 \\ x, s &\succeq_{\mathcal{Q}} 0. \end{aligned}$$

What can be formulated via SDP/SOCP?

A set $S \subseteq \mathbf{R}^d$ is SDP-representable if

$$S = \{x : \exists u \text{ s.t. } Ax + Bu + C \succeq 0\}$$

for some appropriate mappings A, B and matrix C.

Similarly, a function $g : \text{dom}(g) \to \mathbf{R}$ is SDP-representable if the set

$$\text{epi}(g) := \{(t, x) : t \geq g(x)\}$$

is SDP-rep.

Likewise, $S \subseteq \mathbf{R}^d$ is SOCP-rep iff

$$S = \{x : \exists u \text{ s.t. } Ax + Bu + c \succeq_{\mathcal{Q}} 0\}$$

for some appropriate mappings A, B and vector c

Observe:

If S is SDP-rep then $\min_{x \in S} c^\top x$ is an SDP.
If $f(x)$ is SDP-rep then $\min_x f(x)$ is an SDP.

Likewise for SOCP-rep.

Some basic SOCP-rep functions/sets:

$$g(x) = \|x\|, \; g(x) = x^\top x, \; g(x) = a^\top x + b,$$

$$S = \left\{(s, t) \in \mathbf{R}^2 : st > 0, \, t > 0\right\}.$$

Some basic SDP-rep functions:

$$g(X) = \lambda_{\max}(X), \; g(X) = \sum \{k \text{ largest } \lambda_i(X)\}$$

Calculus of SDP-rep/SOCP-rep sets/functions

If S, T are SDP-rep (SOCP-rep) then so are

$S + T$, $S \cap T$, $S \times T$;
$A^{-1}(S)$ for A affine, $A(S)$ for A affine

If $f_1, \ldots, f_m$ and g are SDP-rep (SOCP-rep) then so are

$$\sum_{i=1}^{m} \alpha_i f_i \text{ for } \alpha \geq 0, \quad \max_i f_i, \quad g(f_1(x), \cdots, f_m(x))$$

Many more...

Sometimes it is useful to combine LP/SOCP/SDP:

Example (nearest matrix problems). Given $A \in \mathbf{S}^n$ find the nearest matrix to A in $\mathbf{S}^n_+$.

May be restricted to perturbing only certain entries. For example, maintain zeros in

$$A = \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 0.5 & 0.2 & 0.4 & 0 \\ 0 & 0.4 & 1 & 0.6 \\ 0 & 0 & 0.6 & 1.1 \end{bmatrix}$$

Can consider a more general conic program

$$\begin{array}{ll} \min & \langle c, x\rangle \\ & Ax = b \\ & x \in K, \end{array}$$

where $K = K_1 \times \cdots \times K_r$, and each K_i is one of

$$\mathbf{R}^n_+,\ \mathcal{Q}_n,\ \mathbf{S}^n_+,\ \mathbf{R}^n.$$

Dual of K: $K^* = K_1^* \times \cdots \times K_r^*$.

Conic programming dual

$$\begin{array}{ll} \max & \langle b, y\rangle \\ & A^* y + s = c \\ & s \in K^*. \end{array}$$

Duality/complementarity extend block-wise.

Solvers for LP/SOCP/SDP conic programming

When we mix LP/SOCP/SDP it is convenient to convert matrices into vectors

`vec`: $\mathbf{R}^{n\times n} \to \mathbf{R}^{n^2}$ is the mapping

$$X \mapsto \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1n} & X_{21} & X_{22} & \cdots & X_{nn} \end{bmatrix}^\mathsf{T}$$

`mat`: $\mathbf{R}^{n^2} \to \mathbf{R}^{n\times n}$ is the inverse mapping.

Related mapping `svec`: $\mathbf{S}^n \to \mathbf{R}^{n(n+1)/2}$

$$X \mapsto \begin{bmatrix} X_{11} & \sqrt{2}X_{12} & \cdots & \sqrt{2}X_{1n} & X_{22} & \sqrt{2}X_{23} & \cdots & \sqrt{2}X_{n-1,n} & X_{nn} \end{bmatrix}^\mathsf{T}.$$

Notice: For $X, S \in \mathbf{S}^n$

$$X \bullet S = \mathtt{vec}(X)^\mathsf{T}\mathtt{vec}(S) = \mathtt{svec}(X)^\mathsf{T}\mathtt{svec}(S).$$

SDP solvers

`SeDuMi`: Developed by late J. Sturm. Freely available from :
`http://sedumi.mcmaster.edu.ca`

Matlab-based: Some `.m` and `.mex` files.

Syntax

```
> [x,y,info] = sedumi(A,b,c,K) ;
```

This solves the pair

$$\begin{array}{llll} \min & \langle c, x\rangle & \max & \langle b, y\rangle \\ & Ax = b & & A^*y + s = c \\ & x \in K & & s \in K^*. \end{array}$$

Normal termination gives either an optimal solution, or a certificate (Farkas like) of infeasibility.

Can also use as

```
> [x,y,info] = sedumi(A,b,0,K) ;
```

for $Ax = b$, $x \in K$

```
> [x,y,info] = sedumi(A,0,c,K) ;
```

for $c - A^*y \in K^*$

In matlab environment `A` is an $m \times n$ matrix, `c,x` are n-vectors, and `b,y` are m-vectors.

`K` is a structure that describes K, done through the fields `K.f, K.l, K.q, K.r, K.s`

```
(1) K.f is the number of FREE primal components.
    These are ALWAYS the first components in x.

(2) K.l is the number of NONNEGATIVE components.
    E.g. if K.f=2, K.l=8 then x(3:10) >=0.

(3) K.q lists the dimensions of LORENTZ (second-order) constraints.
    E.g. if K.l=10 and K.q = [3 7] then
    x(11) >= norm(x(12:13)), x(14) >= norm(x(15:20)).
    These components ALWAYS immediately follow the K.l nonneg ones.

(4) K.s lists the dimensions of POSITIVE SEMI-DEFINITE (PSD) const.
    E.g. if K.l=10, K.q = [3 7] and K.s = [4 3], then
    mat( x(21:36),4 ) is PSD, mat( x(37:45),3 ) is PSD.
    These components are ALWAYS the last entries in x.
```

Recall

Example (robust least-squares): Let $\mathcal{U} \subseteq \mathbf{R}^{d\times r}$, $q \in \mathbf{R}^d$, where $d > r$. The problem

$$\min_v \max_{P\in\mathcal{U}} \|Pv - q\|$$

can be formulated as

$$\min_v (\|\bar{P}v - q\| + \rho\|v\|),$$

i.e.,

$$\begin{array}{ll} \max\limits_{t_1,t_2,v} & -t_1 - \rho t_2 \\ & \begin{bmatrix} 0 \\ q \end{bmatrix} - \begin{bmatrix} -t_1 \\ \bar{P}v \end{bmatrix} \succeq_{\mathcal{Q}_{d+1}} 0 \\ & -\begin{bmatrix} -t_2 \\ v \end{bmatrix} \succeq_{\mathcal{Q}_{r+1}} 0. \end{array}$$

Example (robust least-squares):

```
% [At,b,c,K] = rls(P,q,rho)
% Creates dual standard form for robust least squares problem "Pu=q"
function [At,b,c,K] = rls(P,q,rho)
[d, r] = size(P);
% ---------- minimize t_1 + rho * t_2 ----------
b = -[1; rho; zeros(r,1)];
% ---------- (t_1, q - P v) in Qcone ----------
At = [-1, zeros(1,1+r); zeros(d,2), P];
c = [0;q];
K.q = [1+d];
% ---------- (t_2, v) in Qcone ----------
At = [At; 0, -1, zeros(1,r); zeros(r,2), eye(r)];
c = [c; 0;zeros(r,1)] ;
K.q = [K.q, 1+r] ;
```

Recall Lovász theta function for a graph $G = (N, E)$:

$$\begin{array}{rcll} \vartheta(G) & := & \max & ee^\top \bullet X \\ & & & I \bullet X = 1 \\ & & & X_{ij} = 0,\ ij \in E \\ & & & X \succeq 0. \end{array}$$

```
% function [A,b,c,K] = theta(G,n)
% Creates primal standard form for Lovasz theta function
% Assume G is a (2 by numEdges) array that lists the edges
% and n is the number of vertices
function [A,b,c,K] = theta(G,n)
numEdges = size(G,2) ;
A = zeros(numEdges+1,n^2);
% ------ add a constraint for each edge -------
for edge = 1:numEdges
    newconst = zeros(n) ;
    newconst(G(1,edge),G(2,edge)) = 1 ;
    newconst(G(2,edge),G(1,edge)) = 1 ;
    A(edge,:) = vec(newconst)' ;
end
I = eye(n) ;
A(numEdges+1,:) = vec(I)' ;
b = [zeros(numEdges,1); 1] ;
c = -ones(n^2,1);
% ------------ mat(x) in SDP cone -----------
K.s = n ;
```

`SDPT3`: Developed by M. Todd, K. Toh, and R. Tütüncü. Freely available from
`http://www.math.cmu.edu/~reha/sdpt3.html`

It is also matlab-based: `.m` and `.mex` files

Syntax is a bit different:

```
> [obj,X,y,S] = sqlp(blk,A,C,b) ;
```

`blk` describes the blocks (LP/SOCP/SDP) in K.

It works with `svec` instead of `vec`.

References for today's material

- F. Alizadeh and D. Goldfarb, "Second-order Cone Programming," Mathematical Programming 95 (2003) 3–51.
- A. Ben-Tal and A. Nemirovski, "Lectures on Modern Convex Optimization," MPS-SIAM Series on Optimization, 2001. Related material available from `http://www2.isye.gatech.edu/~nemirovs/`
- S. Boyd and L. Vanderberghe, "Convex Optimization," Cambridge Academic Press, 2004. Available from `http://www.stanford.edu/~boyd/cvxbook/`
- `SeDuMi` files and documentation available from `http://sedumi.mcmaster.ca/`