Background

Historical Events

- Lyapunov (stability) 1890
- SDP (cone optimization/duality) 1960's
- Engineering applications 60's
- matrix completion problems 80's
- polynomial time algorithms Nesterov-Nemirovski 80's
- combinatorial appl., primal-dual interior-point (p-d i-p) algorithms (explosion of activity) 90's
- sparsity/special-structure/low-rank/large-scale/robust-opt. 00's

What are SDPs ?

Primal and Dual SDPs (look like LPs with matrix variables)

$$
\text{(PSDP)} \quad \begin{cases} p^* := & \max & \operatorname{trace} CX & (= \langle C, X \rangle) \\ & \text{s.t.} & \mathcal{A} X = b & (b_i = \langle A_i, X \rangle) \\ & & X \succeq 0. & \end{cases}
$$

$$
\text{(DSDP)} \quad \begin{cases} d^* := & \min & b^T y & (= \langle b, y \rangle) \\ & \text{s.t.} & \mathcal{A}^* y - Z = C & (\mathcal{A}^* y = \sum_{i=1}^m y_i A_i) \\ & & Z \succeq 0, & \end{cases}
$$

$\mathcal{S}^n$ space of $n \times n$ real symmetric matrices, $A_i, X, Z, C \in \mathcal{S}^n$

$\succ 0$ ($\succeq 0$) pos. (semi)definiteness; (Loewner partial order)

$\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ lin. transf.; $\mathcal{A}^*$ adjoint transf. (transpose)

Duality: Primal-Dual Pair PSDP, DSDP

PSDP

$$p^* := \max \{\operatorname{trace} CX : \mathcal{A}X = b, X \succeq 0\}$$

Weak Duality (using hidden constraints)

$$\begin{aligned}
p^* &= \max_{X \succeq 0} \min_y \operatorname{trace} CX + [y^T(b - \mathcal{A}X)] \quad \text{(PSDP)} \\
&\le \min_y \max_{X \succeq 0} y^T b + [\operatorname{trace} \overbrace{\overbrace{(C - \mathcal{A}^* y)}^{Z} X}^{\text{duality gap}}] \quad \text{(best LB)} \\
&= \min_{C - \mathcal{A}^* y \preceq 0} y^T b =: d^* \quad \text{(DSDP)}
\end{aligned}$$

$\mathcal{A}^*$ adjoint of $\mathcal{A}$

$$\langle \mathcal{A}(X), y \rangle = y^T \mathcal{A} X = \langle X, \mathcal{A}^* y \rangle = \operatorname{trace} X(\mathcal{A}^* y), \qquad \forall X, \forall y$$

Characterization of (p-d) Optimality

Characterization of Optimality for $Z, X \succeq 0$

$$(*)\begin{cases} \mathcal{A}^*y - Z - C = 0 & \text{dual feasibility} \\ b - \mathcal{A}(X) = 0 & \text{primal feasibility} \end{cases}$$

$$(**)\left\{\ ZX = 0 \quad \text{complementary slackness}\right.$$

$X, (y, Z)$ a primal-dual optimal pair; Z (dual) slack variable

Perturbed complementary slackness

For primal-dual interior-point (p-d i-p) methods, replace (**) with
(***) $ZX = \mu I, \quad Z, X \succ 0, \mu > 0$
solve (*) and (***): X_μ, y_μ, Z_μ on Central Path; $\mu \downarrow 0$

Difference with LP

$Z, X \in \mathcal{S}^n$ but ZX is not necessarily symmetric!

(unlike LP) Strong Duality Can Fail for SDP

Strong Duality for PSDP

e.g. [7]

- zero duality gap: $p^* = d^*$
- AND d^* is attained.
- (if both attained)

$$p^* = d^* \text{ iff } Z \circ X = 0 \text{ iff } \langle Z, X \rangle = 0 \text{ iff } ZX = 0$$

Regularization using Faces

ref. Borwein-W/80 [2, 1, 3], Ramana/97 [4],Ramana-Tuncel-W/97 [5], Tuncel-W/09 [6].

Faces of Cones

Face

A convex cone F is a face of K, denoted $F \trianglelefteq K$, if

$$x, y \in K \text{ and } x + y \in F \implies x, y \in F.$$

If $F \trianglelefteq K$ and $F \neq K$, write $F \triangleleft K$.

Conjugate Face

If $F \trianglelefteq K$, the conjugate face (or complementary face) of F is

$$F^c := F^\perp \cap K^* \trianglelefteq K^*,$$

where $K^* = \{\phi : \langle \phi, k \rangle \geq 0, \forall k \in K\}$ (dual/polar cone)
If $x \in \text{relint}(F)$, then $F^c = \{x\}^\perp \cap K^*$.

Faces of SDP Cone

Face $F \unlhd \mathcal{S}^n_+$ Characterized by $X \in \operatorname{relint} F$

$$X = UDU^T \in \operatorname{relint} F \unlhd \mathcal{S}^n_+, U^TU = I_t, D \in \mathcal{S}^t_{++}$$

iff

$$F = US^t_+U^T$$

Conjugate Face of $F \unlhd \mathcal{S}^n_+$

the conjugate face (or complementary face) of F is

$$F^c := F^\perp \cap \mathcal{S}^n_+ = VS^{n-t}_+V^T, \quad V^TU = 0, V^TV = I_{n-t}$$

Minimal Face (Minimal Cone)

Feasible set of DSDP

Let $\mathcal{F}_D := \{y : Z = \mathcal{A}^* y - C \succeq 0\}$

Minimal Face

Assume $\mathcal{F}_D$ is nonempty, the minimal face (or minimal cone) of DSDP is

$$f_D := \bigcap \{F \trianglelefteq K : \mathcal{A}^*(\mathcal{F}_D) - C \subset F\}$$

i.e., the minimal face that contains all the feasible slacks.

DSDP for Example from Ramana, 1995

DSDP (Max instead of Min)

$$0 = d^* = \max_y \left\{ y_2 \; : \; \begin{pmatrix} y_2 & 0 & 0 \\ 0 & y_1 & y_2 \\ 0 & y_2 & 0 \end{pmatrix} \preceq \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right\}$$

$$y^* = \begin{pmatrix} y_1^* & 0 \end{pmatrix}^T, \quad y_1^* \leq 0, \quad Z^* = C - \mathcal{A}^* y^* = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -y_1^* & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Constraint Qualification (CQ) Fails

Slater's CQ (strict feasibility) fails for dual

PSDP for Example from Ramana, 1995

Primal Program, PSDP (Min instead of Max)

$$1 = p^* = \min_{X \succeq 0} \quad \{X_{11} : \quad \operatorname{trace} A_1 X = X_{22} = 0, \\ \operatorname{trace} A_2 X = X_{11} + 2X_{23} = 1\}$$

$$X^* = \begin{pmatrix} 1 & 0 & X_{13} \\ 0 & 0 & 0 \\ X_{13} & 0 & X_{33} \end{pmatrix}, \quad X_{33} \geq (X_{13}^2)$$

Slater's CQ for (primal) dual & complementarity fails

duality gap $= p^* - d^* = 1 - 0 = \underline{1 > 0}$,

$\operatorname{trace} X^* Z^* = \operatorname{trace} \begin{pmatrix} 1 & 0 & X_{13} \\ 0 & 0 & 0 \\ X_{13} & 0 & X_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -y_1^* & 0 \\ 0 & 0 & 0 \end{pmatrix} = \underline{1 > 0}$

Minimal Face for Ramana Example

Feasible Set/Minimal Face

$\mathcal{F}_D = \{y \in \mathbb{R}^2 : y_1 \leq 0,\ y_2 = 0\}$

$$\begin{array}{rcl} f_D & = & \bigcap\{F \trianglelefteq \mathcal{S}^3_+ : C - \mathcal{A}^*(\mathcal{F}_D) \subset F\} \\ & = & \begin{pmatrix} \mathcal{S}^2_+ & 0 \\ 0 & 0 \end{pmatrix} \\ & \lhd & \mathcal{S}^3_+ \end{array}$$

Slater CQ and Minimal Face

If DSDP is feasible, then

$C - \mathcal{A}^* y \not\succ_K 0, \forall y$ (Slater's CQ fails for DSDP) $\iff f_D \lhd K$

Regularization of DSDP

Borwein-W (1981)

If d^* is finite, then DSDP is equivalent to regularized DSDP

$$d^*_{RD} = \max_y \{\langle b, y\rangle \ : \ \mathcal{A}^* y \preceq_{f_D} C\}. \qquad \text{(RD)}$$

Lagrangian Dual DRD Satisfies Strong Duality:

$$d^* = d^*_{RD} = d^*_{DRD} = \min_X \{\langle C, X\rangle \ : \ \mathcal{A}X = b, \ X \succeq_{f_D^*} 0\} \quad \text{(DRD)}$$

and d^*_{DRP} is attained

Implementation Problems with Regularization; but, Many Applications

Difficulties

Borwein and W. also gave an algorithm to compute f_D.
But Difficulties:

1. The algorithm requires the solution of several (homogeneous) cone programs (constraints are: $\mathcal{A}x = 0, \langle c, x \rangle = 0, 0 \neq x \succeq_K 0$)
2. If Slater's CQ fails for PSDP then it also fails for each of these cone programs.

Application to Combinatorial Problems

Slater CQ fails for many applications to combinatorial problems.
But, f_D can be found explicitly.

Further Differences with LP

Strict Complementarity can Fail

$\boxed{Z + X \succ 0}$ Theorem of Goldman and Tucker for LP can fail, though conditions hold generically; ref. Shapiro/99, Pataki-Tuncel/98, Alizadeh-Haeberly-Overton/98)

Polynomial Time Complexity/Algorithms

SDP are convex programs; can be approximately solved in polynomial time by interior point algorithms (ref. Nesterov-Nemirovski/88)

Strong Relaxations of Computationally Hard Problems

Modelling Computationally Hard Problems

- Many computationally hard problems can be modelled as quadratically constrained quadratic programs, (QQP) (rather than LPs).
- QQPs are themselves computationally hard.
- But, Lagrangian relaxation can be solved efficiently using SDP.

Applications

statistics, engineering, matrix completions, approximation theory, nonlinear programming, Euclidean distance matrix completion, (EDM); sensor network localiz. (SNL) combinatorial optimization: max-cut; graph partitioning; quadratic assignment problem; graph colouring; max-clique.

SDP Webpage

Software

List available at:
www-user.tu-chemnitz.de/ helmberg/sdp_software.html

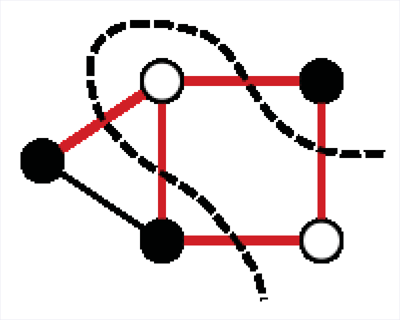
- SDPLIB SDPLIB is a collection of semidefinite programming test problems. (in SDPA sparse format)
- CVX, Disciplined Convex Programming
- Solvers: CSDP (exploits BLAS); SeDuMi1.1 (dependable, popular); SDPT3(including quadratic/sensor localization); SDPA (including parallel); GloptiPoly-3 (moments; optimization; and SDP); PENNON (nonlinear SDP); SBmethod(first order method/large scale);

SDP Relaxation of Max-Cut Problem, (MC)

Max-Cut Problem

undirected, complete, graph $\mathcal{G} = (V, E)$, $|V| = n$, with edge weights w_{ij}; divide nodes into two sets to maximize the sum of weights of cut edges.

A Maximum Cut



Quadratic-Quadratic (QQP) Model for MC

Quadratic Model of MC with Integer Constraints

$$\max\ \tfrac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j), \quad x \in \{\pm 1\}^n.$$

Equate $x_i = 1$ with $i \in \mathcal{I}$; and -1 otherwise.

QQP Model of MC

Let L be the Laplacian of $\mathcal{G}$, e.g. if weights are $0, 1$

$$L_{ij} = \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$$

Let $q(x) := (\frac{1}{4})x^T Lx$; equivalent QP problem

$$(4)\, p^* := \max \left\{ q(x) = x^T Lx : \ x \in \{\pm 1\}^n \right\}$$

SDP Relaxation; use commutativity trace $AB =$ trace BA

Direct Relaxation

$$(4) \quad p^* := \max \left\{ x^T L x : \ x \in \{\pm 1\}^n \right\}$$

Replace $x \in \{\pm 1\}^n$ with $x_i^2 = 1$. Note that

$$x^T L x = \operatorname{trace} x^T L x = \operatorname{trace} L x x^T = \operatorname{trace} L X, \text{with } \overbrace{X = x x^T}^{\text{rank-1}}$$

also: $\quad X \succeq 0, \quad \operatorname{diag}(X) = e, \quad q(x) = \operatorname{trace} L X$

Relax the hard rank-1 condition on X; get SDP relaxation

The SDP Relaxation of MC

$$\boxed{p^* := \max \{\operatorname{trace} L X : \operatorname{diag}(X) = e, X \succeq 0\}}$$

Duality for SDP Relaxation of MC

Primal-Dual Programs

$$\text{(PSDP)} \qquad \begin{array}{rll} d^* = p^* := & \max & \operatorname{trace} LX \\ & \text{s.t.} & \operatorname{diag}(X) = e \\ & & X \succeq 0, X \in \mathcal{S}^n, \end{array}$$

diag: vector from diagonal; Diag: diagonal matrix from vector; e vector of ones;

$$\text{(DSDP)} \qquad \begin{array}{rll} p^* = d^* := & \min & e^T y \\ & \text{s.t.} & \operatorname{Diag}(y) - Z = L \\ & & Z \succeq 0, Z \in \mathcal{S}^n, \end{array}$$

Slater Points

$\hat{X} = I \succ 0$; $\quad \hat{Z} = L - \operatorname{Diag}(\hat{y}) \succ 0$ for $\hat{y} << 0$

Modern Optimality Framework

(Perturbed) Overdetermined Optimality Conditions, $X, Z \succ 0$

$$F_\mu(X,y,Z) = \begin{cases} R_d := \mathrm{Diag}(y) - Z - L & = 0 \quad \text{dual feas.} \\ R_p := \mathrm{diag}(X) - e & = 0 \quad \text{primal feas.} \\ ZX & = 0 \quad \text{compl. slack.} \\ R_c := ZX - \mu I & = 0 \quad \text{pert. C.S.} \end{cases}$$

ZX NOT nec. symmetric

Linearization/(LSS-Gauss)-Newton Direction

$$F'_\mu(X,y,Z)\begin{pmatrix}\Delta X \\ \Delta y \\ \Delta Z\end{pmatrix} = \begin{bmatrix}\mathrm{Diag}(\Delta y) - \Delta Z \\ \mathrm{diag}(\Delta X) \\ Z\Delta X + \Delta Z X\end{bmatrix} = -F_\mu(X,y,Z)$$

Simple/Efficient Algorithm

Block Eliminations; Block Backsolves

- $\leftarrow$ solve for $\Delta Z = \text{Diag}(\Delta y) + R_d$
- substitute $Z\Delta X + (\text{Diag}(\Delta y) + R_d)X$
- $\leftarrow$ solve for $\Delta X = Z^{-1}(-\text{Diag}(\Delta y)X - R_d X - R_c)$
- substitute and solve for Δy
  $\text{diag}\left[Z^{-1}(-\text{Diag}(\Delta y)X - R_d X - R_c)\right] = -R_b$
  equivalently $\boxed{\text{diag}\left[Z^{-1}\text{Diag}(\Delta y)X\right] = (\mu\,\text{diag}(Z^{-1}) - e)}$
  $-\text{diag}\left(Z^{-1}R_d X\right) = 0$, since $R_d = 0$ easy to obtain.

Cheat/Symmetrize ΔX in Backsolve; AHO Search Direction

$\leftarrow$ backsolve for $\Delta Z, \Delta X$; $\Delta X \leftarrow \frac{1}{2}(\Delta X + \Delta X^T)$

Initialization: $X, Z \succ 0$

```
function [phi, X, y] = psd_ip( L);
% solves: max trace(LX) s.t. X psd, diag(X) = b;  b = ones(n,1)/4
%         min b'y       s.t. Diag(y) - L psd, y unconstrained,
%**input:  L ... symmetric matrix
%**output: phi ... optimal value of primal, phi =trace(LX)
%           X   ... optimal primal matrix
%           y   ... optimal dual vector
% call:   [phi, X, y] = psd_ip( L);
%%%%%%%%%%%%%%%%%%%
         %%Initialization
digits = 6;                     % 6 significant digits of phi
[n, n1] = size( L);             % problem size
b = ones( n,1 ) / 4;            % any b>0 works just as well
X = diag( b);                   % initial primal matrix is pos. def.
y = sum( abs( L))' * 1.1;       % initial y is chosen so that
Z = diag( y) - L;               % initial dual slack Z is pos. def.
phi = b'*y;                     % initial dual costs
psi = L(:)' * X( :);            % and initial primal costs
mu = Z( :)' * X( :)/( 2*n);     % initial complementarity
iter=0;                         % iteration count
```

Find Search Direction/Symmetrize *dX*

Solve: *dy*; backsolve: *dZ* , *dX*; symmetrize *dX*

```
disp(['      iter    alphap    alphad      gap     lower      upper']);

while phi-psi > max([1,abs(phi)]) * 10^(-digits)

      iter = iter + 1;          % start a new iteration
      Zi = inv( Z);             % inv(Z) is needed explicitly
      Zi = (Zi + Zi')/2;
      dy =  (Zi.*X) \ (mu * diag(Zi) - b);      % solve for dy
      dX = - Zi * diag( dy) * X + mu * Zi - X;  % back substitute for dX
      dX = ( dX + dX')/2;       % symmetrize
```

Line Search to Stay Interior; and Update

Backtrack to keep $X, Z \succ 0$; Update X, y, Z

```
% line search on primal
      alphap = 1;               % initial steplength
      [dummy,posdef] = chol( X + alphap * dX );  % test if pos.def
      while posdef > 0,
              alphap = alphap * .8;
              [dummy,posdef] = chol( X + alphap * dX );
              end;
      if alphap < 1, alphap = alphap * .95; end; % stay away from boundary
% line search on dual; dZ is handled implicitly: dZ = diag( dy);
      alphad = 1;
      [dummy,posdef] = chol( Z + alphad * diag(dy) );
      while posdef > 0;
              alphad = alphad * .8;
              [dummy,posdef] = chol( Z + alphad * diag(dy) );
              end;
      if alphad < 1, alphad = alphad * .95; end;
% update
      X = X + alphap * dX;
      y = y + alphad * dy;
      Z = Z + alphad * diag(dy);
      mu = X( :)' * Z( :) / (2*n);
      if alphap + alphad > 1.8, mu = mu/2; end; % speed up for long steps
      phi = b' * y; psi = L( :)' * X( :);
% display current iteration
        disp([ iter alphap alphad (phi-psi) psi  phi ]);

        end;            % end of main loop
```

Using a SDP Solver – Problem

Problem

```
 http://infohost.nmt.edu/~sdplib/FORMAT
min 10x1+20x2
st  X=F1x1+F2x2-F0, X >= 0
where
  F0=[1 0 0 0
      0 2 0 0
      0 0 3 0
      0 0 0 4]
  F1=[1 0 0 0
      0 1 0 0
      0 0 0 0
      0 0 0 0]
  F2=[0 0 0 0
      0 1 0 0
      0 0 5 2
      0 0 2 6]
```

Using a SDP Solver

SDP Solver and SDPA Format

```
In SDPA sparse format, this problem can be written as:
"A sample problem.
2 =mdim            "number of constraints m
2 =nblocks         "number of blocks in block diagonal structure
{2, 2}             " sizes of individual blocks
10.0 20.0          " objective function vector
0 1 1 1 1.0         " entries of constraint matrix
0 1 2 2 2.0         " <matno> <blkno> <i> <j> <entry>
0 2 1 1 3.0
0 2 2 2 4.0
1 1 1 1 1.0
1 1 2 2 1.0
2 1 2 2 1.0
2 2 1 1 5.0
2 2 1 2 2.0
2 2 2 2 6.0
```

Success of SDP Relaxation of MC

Goemans-Williamson .878 approx. algor. for MC

MC is one of Karp's NP-complete problems (APX-hard);
G-W '94 showed (with nonnegative weights on edges):

$$.87856(\mathrm{bnd}_{SDP}) \leq \mathrm{optvalue}_{MC} \leq \mathrm{bnd}_{SDP}$$

Extensions/Numerics

This result has been extended (e.g. Nesterov/97) to more general quadratic functions to obtain a $\frac{\pi}{2}$ guarantee
In practice, the strength of the bound is much tighter; large problems can be solved (many authors).