Newton Directions and Symmetrization in Primal-Dual Interior-Point Methods for Semidefinite Programming

Henry Wolkowicz
University of Waterloo

Two Central Themes:

SDP relaxations arise through the dual of the Lagrangian dual; provides geometry explanations and approximations of optimum

Primal-dual interior-point methods for SDP do not require symmetrization when finding the Newton direction.

OUTLINE

- Introduction:
  What is semidefinite programming, SDP, and why use it. (quick)

- Similarities with LP:
  duality, complementarity conditions

- Differences with LP:
  duality gap, strict complementarity, symmetrization

- SDP relaxation:
  from dual of Lagrangian dual

- Search directions:
  different symmetrizations, Gauss-Newton, least squares

INTRODUCTION

Semidefinite Programming
looks just like
Linear Programming

$$
\textbf{(PSDP)} \quad \begin{array}{rccc} p^* = & \max & \operatorname{trace} CX & (\langle C, X \rangle) \\ & \text{s.t.} & \mathcal{A}X = b & \text{(linear)} \\ & & X \succeq 0, \quad (X \in \mathcal{P}) & \text{(nonneg)} \end{array}
$$

$\preceq$ denotes the Löwner partial order

$A \preceq B$ if $B - A \succeq 0$

$\mathcal{S}_n$ denotes $n \times n$ symmetric matrices

$$\mathcal{A} : \mathcal{S}_n \rightarrow \Re^m$$

$$(\mathcal{A}X)_i = \text{trace}\,(A_i X), \;\text{ for given } A_i \in \mathcal{S}_n$$

$\mathcal{P}$ - cone of positive semidefinite matrices

replaces

$\Re^n_+$ - nonnegative orthant

Why use SEMIDEFINITE PROGRAMMING?

Quadratic approximations are better than linear approximations.

Quadratic approximations are too hard to solve!

But, we can solve relaxations of quadratic approximations efficiently using semidefinite programming.

APPLICATIONS

Finding bounds and good feasible solutions for **hard combinatorial problems** such as: max-cut; graph partitioning; quadratic assignment problem; max-clique.

Unconstrained and constrained **NLP** techniques, e.g.: quasi-Newton updates that preserve positive definiteness; Trust region algorithms for large scale minimization; Extended SQP techniques for constrained minimization.

Partial Hermitian **matrix completion problems**.

Engineering problems such as: Ricatti equations; min-max eigenvalue problems; matrix norm minimization; eigenvalue localization.

SIMILARITIES WITH LP

DUALITY

payoff function, player Y to player X (Lagrangian)

$$L(X,y) := \operatorname{trace}(CX) + y^t(b - \mathcal{A}X)$$

Optimal (worst case) strategy for player X:

$$p^* = \max_{X \succeq 0} \min_{y} L(X,y)$$

Using the *hidden constraint* $b - \mathcal{A}X = 0$, recovers primal problem.

$$\begin{aligned} L(X,y) &= \text{trace}\,(CX) + y^t(b - \mathcal{A}X) \\ &= b^t y + \text{trace}\,(C - \mathcal{A}^* y)\, X \end{aligned}$$

adjoint operator, $\quad \mathcal{A}^* y = \sum_i y_i A_i$

$$\langle \mathcal{A}^* y, X \rangle = \langle y, \mathcal{A}X \rangle\,, \quad \forall X, y$$

$$p^* = \max_{X \succeq 0} \min_y L(X, y) \leq d^* := \min_y \max_{X \succeq 0} L(X, y)$$

The dual is obtained from the optimal strategy of the competing player, Y.

The *hidden constraint* $C - \mathcal{A}^* y \preceq 0$ yields the dual

$$\textbf{(DSDP)} \quad \begin{array}{rcc} d^* = & \min & b^t y \\ & \text{s.t.} & \mathcal{A}^* y \succeq C \end{array}$$

for the primal

$$\textbf{(PSDP)} \quad \begin{array}{rcc} p^* = & \max & \operatorname{trace} CX \\ & \text{s.t.} & \mathcal{A}X = b \\ & & X \succeq 0 \end{array}$$

Characterization of optimality for the dual pair X, y (slack $Z \succeq 0$)

$$\mathcal{A}^* y - Z = C \qquad \text{dual feasibility}$$

$$\mathcal{A}X = b \qquad \text{primal feasibility}$$

$$ZX = 0 \qquad \text{complementary slackness}$$

$$ZX = \mu I \qquad \text{perturbed}$$

Forms the basis for:

primal simplex method
dual simplex method
interior point methods

Other dual pairs:

$$
\textbf{(P)} \quad \begin{array}{rll} p^* = & \max & b^t y \\ & \text{s.t.} & \mathcal{A}^* y \preceq C \end{array}
$$

$$
\textbf{(D)} \quad \begin{array}{rll} d^* = & \min & \operatorname{trace} CX \\ & \text{s.t.} & \mathcal{A}X = b \\ & & X \succeq 0 \end{array}
$$

However, there are several subtle
DIFFERENCES with LP.

1. Duality gaps can exist for SDP in the absence of strictly feasible solutions (Slater's constraint qualification, CQ).

2. Strict complementarity can fail at the optimum.

$\Rightarrow$ 3. The perturbed optimality conditions involve nondiagonal matrices and map between different spaces
(overdetermined -does it require symmetrization?).

1. Duality gaps:

<u>Example</u>

If the primal is

$$
\textbf{(P)} \qquad \begin{array}{rll}
p^* = & \max & x_2 \\
& \text{s. t.} & \begin{bmatrix} x_2 & 0 & 0 \\ 0 & x_1 & x_2 \\ 0 & x_2 & 0 \end{bmatrix} \preceq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
\end{array}
$$

then the dual is

$$
\textbf{(D)} \qquad \begin{array}{rlc}
d^* = & \min & \operatorname{trace} U_{11} \\
& \text{s. t.} & U_{22} = 0 \\
& & U_{11} + 2U_{23} = 1 \\
& & U \succeq 0.
\end{array}
$$

Then $p^* = 0 < d^* = 1$.

2. Strict complementarity:

<u>Example</u>

If the primal is

$$
\textbf{(P)} \quad
\begin{array}{rll}
p^* = & \max & x_1 \\
& \text{s. t.} & \begin{bmatrix} x_1 & x_3 & x_2 \\ x_3 & x_2 & 0 \\ x_2 & 0 & x_3 \end{bmatrix} \preceq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{array}
$$

then the dual is

$$
\textbf{(D)} \quad
\begin{array}{rll}
d^* = & \min & \operatorname{trace} U_{33} \\
& \text{s. t.} & U_{11} = 1 \\
& & U_{22} + 2U_{13} = 0 \\
& & U_{33} + 2U_{12} = 0 \\
& & U \succeq 0.
\end{array}
$$

Then the (unique) optimum pair is:
$U = \operatorname{Diag}(1, 0, 0)$
and $x = [0, 0, 0]$ with slack $Z = \operatorname{Diag}(0, 0, 1)$

Direct SDP Relaxation

Max-Cut Problem

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

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

$$q(x) := x^t Q x,$$

where Q is an $n \times n$ symmetric matrix. An equivalent problem is the homogeneous (± 1)-*quadratic programming problem*

$$\mu^* := \max \ q(x), \quad x \in \{\pm 1\}^n.$$

Replace $x \in \{\pm 1\}^n$ constraints $x_i^2 = 1$.

Note that for

$$X = x x^t,$$

$$X \succeq 0, \ \operatorname{diag}(X) = e, \ q(x) = \operatorname{trace} XQ.$$

Relax the rank-1 condition on X to get SDP.

How does SDP arise from general quadratic approximations?

Let

$$q_i(y) = \frac{1}{2} y^t Q_i y + y^t b_i + c_i, \;\; y \in \Re^n$$

$$(\mathbf{QQP}) \quad \begin{array}{rll} q^* = & \min & q_0(y) \\ & \text{s.t.} & q_i(y) \leq 0 \\ & & i = 1, \ldots m \end{array}$$

Lagrangian:

$$L(y, x) \;\; = \;\; q_0(y) + \textstyle\sum_{i=1}^m x_i q_i(y)$$

or equivalently

$$\begin{array}{rcl} L(y, x) & = & \frac{1}{2} y^t (Q_0 + \sum_{i=1}^m x_i Q_i) y \\ & & + y^t (b_0 + \sum_{i=1}^m x_i b_i) \\ & & + (c_0 + \sum_{i=1}^m x_i c_i) \end{array}$$

Weak duality follows from “hidden constraint”

$$d^* = \max_{x \geq 0} \min_y L(y, x) \leq q^* = \min_y \max_{x \geq 0} L(y, x).$$

Now homogenize; multiply linear term by new variable y_0

$$y_0 y^t (b_0 + \sum_{i=1}^m x_i b_i), \quad y_0^2 = 1.$$

$$
\begin{aligned}
d^* &= \\
&= \max_{x\geq 0} \min{}_y \quad L(y,x) \\
&= \max_{x\geq 0} \min_{y_0^2=1} \quad \tfrac{1}{2}y^t(Q_0 + \textstyle\sum_{i=1}^m x_i Q_i)y \qquad + t y_0^2 \\
&\qquad\qquad + y_0 y^t(b_0 + \textstyle\sum_{i=1}^m x_i b_i) \\
&\qquad\qquad + (c_0 + \textstyle\sum_{i=1}^m x_i c_i) \qquad - t \\
&= \max_{x\geq 0, t} \min_{y} \quad \tfrac{1}{2}y^t(Q_0 + \textstyle\sum_{i=1}^m x_i Q_i)y \qquad + t y_0^2 \\
&\qquad\qquad + y_0 y^t(b_0 + \textstyle\sum_{i=1}^m x_i b_i) \\
&\qquad\qquad + (c_0 + \textstyle\sum_{i=1}^m x_i c_i) \qquad - t
\end{aligned}
$$

The hidden semidefinite constraint yields the semidefinite program.

We get

$$B := \begin{pmatrix} 0 & b_0^t \\ b_0 & Q_0 \end{pmatrix},$$

$A : \Re^{m+1} \to \mathcal{S}_{n+1}$

$$A \begin{pmatrix} t \\ x \end{pmatrix} := - \begin{bmatrix} t & \sum_{i=1}^m x_i b_i^t \\ \sum_{i=1}^m x_i b_i & \sum_{i=1}^m x_i Q_i \end{bmatrix}$$

and the SDP constraint

$$B - A \begin{pmatrix} t \\ x \end{pmatrix} \succeq 0.$$

NOTE There is NO hidden constraint on the Q_i if all q_i are convex. There is no need for SDP in the convex case. Better algorithms exist, e.g. proximal methods, using quadratic cones, ...

The dual program is equivalent to the SDP (with $c_0 = 0$)

$$\textbf{(D)} \qquad \begin{array}{rll} d^* = & \sup & -t + \sum_{i=1}^m x_i c_i \\ & \text{s.t.} & A \begin{pmatrix} t \\ x \end{pmatrix} \preceq B \\ & & x \in \Re^m, t \in \Re \end{array}$$

As in linear programming, the dual of the dual is obtained from the optimal strategy of the competing player:

$$\textbf{(DD)} \qquad \begin{array}{rll} d^* = & \inf & \text{trace}\, BU \\ & \text{s.t.} & A^* U = \begin{pmatrix} -1 \\ c \end{pmatrix} \\ & & U \succeq 0. \end{array}$$

In some sense, the Lagrangian relaxation is the **best tractable relaxation**.

(Is there a better one?)

geometrical description of SDP relaxations: approximate feasible set F by valid (not necessarily convex) inequalities:

$$F \subset \left\{y : \sum_{i=1}^{m} x_i q_i(y) \leq 0\right\}$$

where $x \geq 0$ and

$$Q_0 + \sum_{i=1}^{m} x_i Q_i \succeq 0.$$

Recover optimal approximation from optimal dual $x*$

$$\bar{y} \in \arg\min L(y, x^*)$$

Obtained from column 1 of optimal SDP matrix Y^*. Then shift to guarantee complementary slackness with valid optimal inequality.

The Trust Region Subproblem:

Let

$$q(x) := x^t A x - 2a^t x,$$

$$(TRS) \qquad \begin{array}{rll} \mu^* := & \min & q(x) \\ & \text{s.t.} & x^t x = s^2 \quad (\leq s^2). \end{array}$$

where

$A = A^t$, not necessarily semidefinite
$a \in \Re^n$, $s > 0$.
(application: quadratic model for unconstrained minimization)

Homogenization of TRS

$$
\begin{aligned}
\mu^* &= \min_{||x||=s,\ y_0^2=1} x^tAx - 2y_0a^tx \\
&= \max_t \min_{||x||=s} x^tAx - 2y_0a^tx + ty_0^2 - t \\
&= \max_t \min_{||x||=s,\ y_0^2=1} x^tAx - 2y_0a^tx + ty_0^2 - t \\
&= \max_t \min_{||x||^2+y_0^2=s^2+1} x^tAx - 2y_0a^tx + ty_0^2 - t
\end{aligned}
$$

$$
= \max_t (s^2+1)\lambda_1(D(t)) - t
$$

$$
D(t) = \begin{pmatrix} t & -a^t \\ -a & A \end{pmatrix}.
$$

unconstrained dual problem to TRS

$$D(t) = \begin{pmatrix} t & -a^t \\ -a & A \end{pmatrix}$$

$y = \begin{pmatrix} y_0 \\ x \end{pmatrix}$ normalized eigenvector for $\lambda_{\min} D(t)$

$$k(t) = (s^2 + 1)\lambda_{\min}(D(t)) - t,$$

$$*** \qquad \max_t k(t)$$

Note

$$k'(t) = (s^2 + 1)y_0^2 - 1 = 0$$

is feasibility for x

SDP Primal-Dual Pair

$$\max_t k(t) = (s^2 + 1)\lambda_{\min}(D(t)) - t,$$

add the variable λ

$$\begin{array}{ll} \max & (s^2+1)\lambda - t \\ \text{s.t.} & D(t) \succeq \lambda I \end{array} \quad (DSDP)$$

Lagrangian dual of this dual is:

$$\begin{array}{ll} \min & \operatorname{trace} D(0)X \\ \text{s.t.} & \operatorname{trace} X = s^2 + 1 \\ & X_{11} = 1 \\ & X \succeq 0 \end{array} \quad (PSDP)$$

primal-dual interior point method:

approx. solve perturbed optimality conditions using Newton's method:

$$\begin{gathered}
\operatorname{trace} X = s^2 + 1 \\
X_{11} = 1 \\
D(t) - \lambda I - Z = 0 \\
\mu Z^{-1} - X = 0 \\
X \succ 0, Z \succ 0
\end{gathered}$$

QUADRATIC ASSIGNMENT PROBLEM
QAP

$$\mu^* := \min_{X \in \Pi} \quad \operatorname{trace} AXBX^t - 2CX^t$$

A, B and C are real $n \times n$ matrices
Π is the set of permutaion matrices.

Rewrite as

$$(QAP_E) \qquad \begin{array}{rll} \mu^* := & \min & \operatorname{trace} AXBX^t - 2CX^t \\ & \text{s.t.} & XX^t = I, \left(X^tX = I\right) \\ & & \left(Xe = X^te = e\right) \\ & & X_{ij}^2 - X_{ij} = 0, \quad \forall i, j. \end{array}$$

ignore $Xe = X^te = e$ for now

Find the semidefinite relaxation by taking the dual of the Lagrangian dual.

We first add the (0,1)-constraints to the objective function using Lagrange multipliers W_{ij}

$$\mu_{\mathcal{O}} = \min_{XX^t=X^tX=I} \max_{W} \operatorname{trace} AXBX^t - 2CX^t + \sum_{ij} W_{ij}(X_{ij}^2 - X_{ij}).$$

We now homogenize the objective function by multiplying by a constrained scalar x_0

$$\mu_{\mathcal{O}} \geq \mu_R = \max_{W} \min_{\substack{XX^t=X^tX=I \\ x_0^2=1}} \operatorname{trace}\left[AXBX^t + W(X \circ X)^t - x_0(2C+W)X^t\right].$$

Introducing a Lagrange multiplier w_0 for the constraint on x_0 and Lagrange multipliers S_b for $XX^t = I$ and S_o for $X^tX = I$ we get

$$
\begin{array}{ll}
\multicolumn{2}{l}{\mu_{\mathcal{O}} \geq \mu_R :=} \\
\max\limits_{W} \min\limits_{X,\ x_0} & \operatorname{trace}\Big[AXBX^t + W(X \circ X)^t + w_0 x_0^2 \\
& +S_b XX^t + S_o X^t X\Big] \\
& -\operatorname{trace} x_0(2C + W)X^t \\
& -w_0 - \operatorname{trace} S_b - \operatorname{trace} S_o.
\end{array}
$$

We have grouped the quadratic, linear, and constant terms together. We now define $x := \operatorname{vec} X$, $y^t := (x_0, x^t)$ and $w^t := (w_0, \operatorname{vec} W^t)$ and get

$$
\begin{array}{ll}
\multicolumn{2}{l}{\mu_R =} \\
\max\limits_{W} \min\limits_{y} & y^t \Big[L_Q + Arrow(w) + \mathrm{B}^0\mathrm{Diag}\,(S_b) + \\
& \mathrm{O}^0\mathrm{Diag}\,(S_o)\Big]\, y \\
& -w_0 - \operatorname{trace} S_b - \operatorname{trace} S_o
\end{array}
$$

We used the $(n^2+1)\times(n^2+1)$ matrix

$$L_Q := \begin{bmatrix} 0 & -\mathrm{vec}\,(C)^t \\ -\mathrm{vec}\,(C) & B\otimes A \end{bmatrix},$$

and the (interesting) linear operators

$$\mathrm{Arrow}\,(w) := \begin{bmatrix} w_0 & -\frac{1}{2}w^t_{1:n^2} \\ -\frac{1}{2}w_{1:n^2} & \mathrm{Diag}\,\left(w_{1:n^2}\right) \end{bmatrix},$$

$$\mathrm{B}^0\mathrm{Diag}\,(S) := \begin{bmatrix} 0 & 0 \\ 0 & I\otimes S_b \end{bmatrix}$$

and

$$\mathrm{O}^0\mathrm{Diag}\,(S) := \begin{bmatrix} 0 & 0 \\ 0 & S_o\otimes I \end{bmatrix}.$$

The hidden semidefinite constraint yields the equivalend SDP:

$$
(D_{\mathcal{O}}) \quad \begin{array}{ll} \max & -w_0 - \operatorname{trace} S_b - \operatorname{trace} S_o \\ \text{s.t.} & L_Q + \operatorname{Arrow}(w) + \\ & \mathrm{B}^0\operatorname{Diag}(S_b) + \mathrm{O}^0\operatorname{Diag}(S_o) \succeq 0, \end{array}
$$

The dual of this dual yields the semidefinite relaxation.

$Y \succeq 0$ is $(n^2+1) \times (n^2+1)$
the dual matrix variable

$$
(SDP_{\mathcal{O}}) \quad \begin{array}{lll} \min & \operatorname{trace} L_Q Y & \\ \text{s.t.} & \mathrm{b}^0\operatorname{diag}(Y) = I & \mathrm{o}^0\operatorname{diag}(Y) = I \\ & \operatorname{arrow}(Y) = e_0 & Y \succeq 0 \end{array}
$$

adjoint operators are:

$$\mathrm{arrow}\,(Y) := \mathrm{diag}\,(Y) - (0, (Y_{0,1:n^2})^t.$$

$$\mathrm{b}^0\mathrm{diag}\,(Y) := \sum_{k=1}^{n} Y_{(k-1)n+1:kn,(k-1)n+1:kn}$$

$$[\mathrm{o}^0\mathrm{diag}\,(Y)]_{ij} := \mathrm{trace}\, Y_{(i-1)n+1:in,(j-1)n+1:jn}$$

Direct Approach to SDP Relaxation

Let
$X \in \Pi_n$ be a permutation matrix
$x = \text{vec}\,(X)$, $c = \text{vec}\,(C)$.

$$
\begin{aligned}
q(X) &= \text{trace}\, AXBX^t - 2CX^t \\
&= x^t(B \otimes A)x - 2c^t x \\
&= \text{trace}\, xx^t(B \otimes A) - 2c^t x \\
&= \text{trace}\, L_Q Y_X,
\end{aligned}
$$

where L_Q is as above and

$$
Y_X := \begin{bmatrix} 1 & x^t \\ x & xx^t \end{bmatrix}.
$$

INTERIOR-POINT METHODS
for
LP and SDP

started with Karmarkar 1984

primal-dual interior-point methods:

1. potential reduction

2. path following

3. infeasible-interior-point

4. predictor-corrector

(see e.g. recent book by Steve Wright, or interior-point archives)

Connection between Karmarkar's algorithm and log-barrier problems (Stanford gang of 4: Gill, Murray, M. Wright, Saunders)

$$
\textbf{(D)} \qquad
\begin{array}{rll}
d^* = & \min & b^t y \\
 & \text{s.t.} & \mathcal{A}^* y - Z = C \\
 & & Z \succeq 0
\end{array}
$$

log-barrier problem for dual SDP, with fixed $\mu > 0$

$$
\begin{array}{ll}
\min & b^t y - \mu \log \det Z \\
\text{s.t.} & \mathcal{A}^* y - Z = C
\end{array}
$$

and $\mu \downarrow 0$

Lagrangian for log-barrier

$$
\min b^t y + \operatorname{trace} X \left(C - \mathcal{A}^* y + Z \right) - \mu \log \det Z
$$

Optimality conditions for the
dual log-barrier
solve $F_\mu(X, y, Z) = 0$, with $X, Z \succ 0$

$$\frac{\partial}{\partial X}: \quad C - \mathcal{A}^* y + Z = 0 \qquad \text{dual feasibility}$$

$$\frac{\partial}{\partial y}: \quad b - \mathcal{A}X = 0 \qquad \text{primal feasibility}$$

$$\frac{\partial}{\partial Z}: \quad X - \mu Z^{-1} = 0 \qquad \text{pert. compl. slack.}$$

or, to avoid ill-conditioning, use:

$ZX - \mu I = 0$ perturbed complementary slackness

Note:
the nonlinear system $F_\mu(X, y, Z) = 0$ is overdetermined if (more linear) $ZX - \mu I = 0$ is used.
(this is **not** the case for LP)

CENTRAL PATH

For each $\mu > 0$, there exists a unique (X_μ, y_μ, Z_μ), with $X_\mu, Z_\mu \succ 0$, that solves the nonlinear system $F_\mu(X, y, Z) = 0$.

$$\mathcal{C} = \{(X_\mu, y_\mu, Z_\mu) : \mu > 0\}$$

well defined,
converges to optimum as $\mu \downarrow 0$,
long steps possible in neighbourhoods near $\mathcal{C}$

p-d i-p methods followed:
Megiddo 1987, Kojima, Mizuono, and Yoshise 1987, many others ....

In numerical analysis, how do we solve a nonlinear system of equations?

The standard approach is to solve the equivalent nonlinear least squares problem using "Gauss-Newton" method.

$$\min \frac{1}{2}||F_\mu(X, y, Z)||_2^2$$

subject to $X, Z > 0$.

In the LP case, when we differentiate we get a square system and Gauss-Newton reduces to Newton's method.

i.e. solving

$$\left(F'_\mu\right)^t F'_\mu(\Delta s) = -\left(F'_\mu\right)^t F_\mu$$

(where F'_μ is the Jacobian)
is equivalent to solving

$$F'_\mu(\Delta s) = -F_\mu$$

framework p-d i-p:
Given $(X^0, y^0, Z^0) \in \mathcal{F}^0$
for $k = 0, 1, 2 \ldots$
solve for the search direction

$$F_\mu'(X^k, y^k, Z^k) \begin{pmatrix} \Delta X^k \\ \Delta y^k \\ \Delta Z^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -X^k Z^k + \sigma_k \mu_k I \end{pmatrix}$$

where σ_k centering, $\mu_k = \operatorname{trace} X^k Z^k / n$

$$\begin{aligned}(X^{k+1}, y^{k+1}, Z^{k+1}) = \\ (X^k, y^k, Z^k) + \alpha_k(\Delta X^k, \Delta y^k, \Delta Z^k)\end{aligned}$$

so that $(X^{k+1}, Z^{k+1}) \succ 0$
end (for).

Complications in the SDP case:

Overdetermined system, i.e. should we symmetrize ZX, e.g. $ZX + XZ$, many others ...

line search

strict complementarity may fail

duality gaps

SYMMETRIZATION

from log-barrier we get:
$X - \mu Z^{-1} = 0$
with linearization:
$X + \Delta X - \mu \left(Z^{-1} - Z^{-1} \Delta Z Z^{-1} \right) = 0$

more linear version - multiply by Z:
$ZX - \mu I = 0$
with linearization:
$ZX - \mu I + Z \Delta X + \Delta Z X = 0$

symmetrize:
$ZX + ZX - 2\mu I = 0$
with linearization:
$ZX + ZX - 2\mu I + Z\Delta X + \Delta X Z + \Delta Z X + X \Delta Z = 0$

Others:
$Z^{1/2} X Z^{1/2} - \mu I = 0$, etc...
$H_P(M) := \frac{1}{2} \left(PMP^{-1} + (PMP^{-1})^t \right)$
$M = XZ$ and P any nonsingular matrix

$XZ + ZX$:
advantage: fast convergence and stability;
disadvantage: solve Lyapunov equation when finding Newton direction

Heuristic; symmetrize ΔX after it is calculated:
advantages: fast convergence and inexpensive Newton direction, especially for max-cut problem!
disadvantage: stability

Still many open questions, e.g. why symmetrize? Can apply Gauss-Newton method to solving overdetermined system of nonlinear equations.

“best direction”: Newton direction for least squares problem

best compromise: Gauss-Newton direction

other directions: are approximations of the Gauss-Newton direction

Shrinking the Overdetermined System

We can find redundant equations and solve a smaller system. We can remove the lower triangular part from the nonlinear equation and get the equivalent equation

$$\mathcal{T}\,(ZX - \mu I) = 0,$$

where the linear operator $\mathcal{T} : \mathcal{M}_n \to \mathcal{S}_n$ by ignoring the strictly lower triangular part of the matrix, i.e. the i,j components are

$$(\mathcal{T}(W))_{ij} = \begin{cases} W_{ij} & \text{if } i \le j \\ W_{ji} & \text{if } i > j \end{cases}$$

The resulting system is now square.

$\mathcal{T}$ is an orthogonal projection

The optimality conditions become

$$\begin{pmatrix} Z + C - \mathcal{A}^* y \\ b - \mathcal{A}(X) \\ \mathcal{T}(ZX - \mu I) \end{pmatrix} = 0.$$

This does not introduce new nonlinearities but does guarantee that we map between the same spaces. In fact, we can prove that we do not lose information in the optimality conditions when we only consider the upper triangular parts.

LEMMA

Suppose that $Z \succ 0$. Then the linear operator $\mathcal{T}_Z(\cdot) := \mathcal{T}(Z\cdot)$ is a one-one mapping on $\mathcal{S}_n$. In addition, if $\mathcal{T}_Z(X) = \mu I$, for some $X = X^t$, then $ZX = \mu I$.

The Newton direction is obtained from solving the system

$$\begin{aligned} \Delta Z - \mathcal{A}^*(\Delta y) &= -F_d \\ -\mathcal{A}(\Delta X) &= -F_p \\ \mathcal{T}(Z\Delta X + \Delta Z X) &= -\mathcal{T}(ZF_{Z,X}). \end{aligned}$$

Equivalently,

$$\begin{bmatrix} 0 & -\mathcal{A}^* & I \\ -\mathcal{A} & 0 & 0 \\ \mathcal{T}_Z(\cdot) & 0 & \mathcal{T}(\cdot X) \end{bmatrix} \begin{pmatrix} \Delta X \\ \Delta y \\ \Delta Z \end{pmatrix} = \begin{pmatrix} -F_d \\ -F_p \\ -\mathcal{T}_Z(X) + \mu I \end{pmatrix}.$$

```
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);

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
psi = L(:)' * X( :);              % and primal costs
mu = Z( :)' * X( :)/( 2*n);       % initial complementarity
iter=0;                           % iteration count

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 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
```

486 66 Mhz, PC using a C encoded version;
# of iterations does not seem to depend significantly on n
hh gives the number of hours, mm the number of minutes, ss the seconds.
random unweighted graphs with edge probability $p = 0.5$.
stopping condition to 6 digits of accuracy

size n	iterations	hh:mm:ss
100	14	48
150	12	2:30
200	12	6:30
250	13	14:24
300	14	30:24
400	14	1:24:36
500	14	2:13.18

included triangle constraints;
favorable to use Mehrotra's predictor – corrector approach
system is solved twice with the same factorization but with different right hand sides.
The triangle inequalities were included successively according to their amount of violation
programs have n equality and $4\binom{n}{3}$ inequality constraints in the dual program.
The final solution satisfies all these constraints.
# of iterations astonishingly small
e.g. problems with $n = 100$ satisfies $4\binom{100}{3} = 646,800$ inequality constraints.

random graphs, edge-weights $\in \{-10, \ldots, 10\}$				
size n	cut	upper bnd (%)	hh:mm:ss	iters
30	330	330.0001 (0.0)	54	59
40	619	619.0000 (0.0)	2:02	60
50	708	708.0001 (0.0)	1:04:22	166
70	1215	1248.7142 (2.8)	4:42:21	220
100	2440	2540.2528 (4.1)	37:56:05	266
random unweighted graphs, edge probability 0.5				
size n	cut	upper bnd (%)	hh:mm:ss	iters
30	143	143.0000 (0.0)	56	62
40	245	245.6864 (0.3)	59:19	247
50	374	375.3689 (0.4)	1:54:18	250
70	723	726.0399 (0.4)	5:57:22	257

Simple convex program:

$$(\mathbf{CP}) \; p^* = \begin{array}{ll} \text{min} & f(x) \\ \text{s.t.} & x \in \Omega \end{array}$$

where:

$f : X \to \Re$ - differentiable, convex function

$\Omega \subset X$ is an abstract inner-product space

Ω - convex set

Pshenichnyi Condition:

$x^* \in \Omega$ solves (CP) **iff** $\nabla f(x^*) \in (\Omega - x^*)^+$

polar cone

$$K^+ = \{\phi \in X^* : \langle \phi, k \rangle \geq 0, \; \forall k \in K\}$$

special case when Ω closed convex cone

$x^* \in \Omega$ solves (CP)

iff

$\nabla f(x^*) \in (\Omega)^+$ and

$\langle \nabla f(x^*), x^* \rangle = 0$ (compl. slack.)

special case when $\Omega = \Re^n_+$
if $z, x \in \Re^n_+$
$z^t x = 0$ iff $z_i x_i = 0$ for all i

special case when $\Omega = \mathcal{P}$, psd matrices
if $Z, X \in \mathcal{P}$
$\langle Z, X \rangle = \operatorname{trace} ZX = 0$ iff $ZX = 0$

However, these special cases do not hold in general. The purpose of *complicating* the complementarity condition is to obtain the *powerful* central path concept. (Avoids convergence to nonfeasible -but stationary - points.)

for some y and X feasible for $\mathcal{A}X = b$

$$p^* = \max \operatorname{trace}(CX) + y^t(b - \mathcal{A}X) \quad \text{s.t.} \quad X \succeq 0$$

Optimality Conditions,
$F(X, y, Z) = 0$, $X, Z \succeq 0$:

$$\nabla F : \begin{cases} C - \mathcal{A}^* y - Z = 0, & \text{dual feasibility} \\ b - Ax = 0 & \text{primal feasibility} \\ \langle Z, X \rangle = 0 & \text{complementary slackness} \end{cases}$$

Recall:
$\langle Z, X \rangle = \operatorname{trace} ZX = 0$ iff $ZX = 0$

The above system is **underdetermined**
Why not use simpler underdetermined system?

STABILITY QUESTIONS

Let $f : \Re^n \to \Re$ and $g : \Re^n \to \Re$ be twice continuously differentiable functions

(e.g. f is linear part of log-barrier problem, i.e. it is the Lagrangian of (PSDP); while $g(x) = \det(x)$, is the determinant.)

consider

$$(NLP) \qquad \begin{array}{ll} \min & f(x) \\ \text{subject to} & g(x) \geq 0, \end{array}$$

Optimality under CQ:

$$\begin{aligned}
\nabla f(x) - \lambda \nabla g(x) &= 0 \\
\langle \lambda, g(x) \rangle &= 0 \\
g(x) &\geq 0 \\
\lambda &\geq 0.
\end{aligned}$$

$\mu > 0$ fixed log-barrier parameter
log-barrier problem is:

$$\min_x B_\mu(x) := f(x) - \mu \log g(x).$$

stationarity

$$0 = \nabla B_\mu(x_\mu) = \nabla f(x_\mu) - \frac{\mu}{g(x_\mu)} \nabla g(x_\mu), \; g(x_\mu) > 0.$$

we can assume $x_\mu \to x^*$ and $g(x_\mu) \to 0$
Therefore

$$\frac{\mu}{g(x_\mu)} \to \lambda^*$$

Hessian

$$\nabla^2 P_\mu(x) =$$

$$\left\{\nabla^2 f(x) - \frac{\mu}{g(x)}\nabla^2 g(x)\right\} + \left[\left(\frac{\mu}{g(x)}\right)\frac{1}{g(x)}\nabla g(x)\nabla g(x)^t\right].$$

first { } part converges

second [ ] part is rank one $\Rightarrow$ ill-conditioning

Fix: add new variable $\Delta v \in \Re$

$$\Delta v = -\frac{\mu}{g^2(x)}\nabla g(x)^t \Delta x.$$

Newton direction: $d = (\Delta x, \Delta v)^t$,

$$\begin{bmatrix} \nabla^2 f(x_\mu) - \frac{\mu}{g(x_\mu)}\nabla^2 g(x_\mu) & \nabla g(x_\mu) \\ \nabla g(x_\mu)^t & \frac{g(x_\mu)^2}{\mu} \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta v \end{pmatrix}$$

$$= \begin{pmatrix} -\nabla f(x_\mu) + \frac{\mu}{g(x_\mu)}\nabla g(x_\mu) \\ 0 \end{pmatrix}.$$

As $\mu \to 0$ the Hessian becomes

$$\begin{bmatrix} \nabla^2 f(x^*) - \lambda^* \nabla^2 g(x^*) & \nabla g(x^*) \\ \nabla g(x^*) & 0 \end{bmatrix}.$$

This matrix is nonsingular under the standard second order sufficient optimality conditions. i.e. that the Hessian of the Lagrangian is positive definite on the tangent space; equivalently, $\nabla^2 f(x^*) - \lambda^* \nabla^2 g(x^*)$ is positive definite on the orthogonal complement of $\nabla g(x^*)$.

alternative to overcome ill-conditioning

use perturbed KKT conditions
use substitution $\lambda = \frac{\mu}{g(x)}$ or $\lambda g(x) = \mu$

Then becomes

$$\begin{aligned} \nabla f(x) - \lambda \nabla g(x) &= 0 \\ \lambda g(x) &= \mu \\ g(x) &> 0 \end{aligned}$$

i.e. $\lambda g(x) = \mu$ avoids ill-conditioning that arises when differentiating $\lambda = \frac{\mu}{g(x)}$.

Summary:
two ways to avoid ill-conditioning of log-barrier problem.

1. increase the dimension of the problem by one - still finds the Newton direction

2. change the optimality conditions (perturb) direction based solely on the optimality conditions and not on a log-barrier problem
(not a Newton direction)

Alternate motivation for changing true log-barrier direction:

$X - \mu Z^{-1} = 0$

with linearization:

$X + \Delta X - \mu \left(Z^{-1} - Z^{-1} \Delta Z Z^{-1} \right) = 0$

is to get more linear version - multiply by Z:

$ZX - \mu I = 0$

with linearization:

$ZX - \mu I + Z \Delta X + \Delta Z X = 0$

MORE LINEARITY

SPEEDS UP NEWTON'S METHOD

but

results in overdetermined system