APPLICATIONS OF SEMIDEFINITE PROGRAMMING

(Linear Programming for the 90's and 00's)

Henry Wolkowicz
University of Waterloo

Central Theme:

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

OUTLINE

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

- SDP relaxation:
  from dual of Lagrangian dual

- several examples, e.g. QAP, GP,

- specific example: Euclidean distance matrix problems

INTRODUCTION

Semidefinite Programming
looks just like
Linear Programming

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

$\preceq$ denotes the Löwner partial order
$A \preceq B$ if $B - A \succeq 0$

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

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

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

$\mathcal{P}$ - 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** and Euclidean distance 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, $\quad \mathcal{A}^* y = \sum_i y_i A_i$

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

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

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

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

for the primal

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

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

Direct SDP Relaxation

Max-Cut Problem

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

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*

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

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

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

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

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

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

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

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 $c_0 = 0$)

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

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

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

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

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

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

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

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

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

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

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

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

and

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

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

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

adjoint operators are:

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

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

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

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

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

```
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

Euclidean Distance Matrix Completion Problem

$n \times n$ symmetric nonnegative (elementwise) matrix $D = (d_{ij})$ with zero diagonal is called a *pre-distance matrix*.

if there exist points $x^1, x^2, \ldots, x^n$ in $\Re^r$ such that

$$d_{ij} = \|x^i - x^j\|^2, \qquad i, j = 1, 2, \ldots, n, \qquad (1)$$

then D is called a *Euclidean distance matrix* (EDM).

The smallest value of r is called *the embedding dimension* of D. Note that r is always $\leq n-1$.

a pre-distance matrix D is a EDM if and only if D is negative semidefinite on

$$M := \left\{ x \in \Re^n : x^t e = 0 \right\},$$

the orthogonal complement of e, where e is the vector of all ones.

Define the $n \times n$ orthogonal matrix

$$Q := \left[\frac{1}{\sqrt{n}} e \mid V \right], \qquad Q^t Q = I. \tag{2}$$

Thus $V^t e = 0$ and $V^t V = I$. Moreover, the subspace M can be represented as the range of the $n \times (n-1)$ matrix V and

$$J := VV^t = I - \frac{ee^t}{n} \tag{3}$$

is the orthogonal projection onto M.

Now define the *centered* and *hollow* subspaces

$$\begin{aligned} \mathcal{S}_C &:= \{B \in \mathcal{S}_n : Be = 0\}, \\ \mathcal{S}_H &:= \{D \in \mathcal{S}_n : \operatorname{diag}(D) = 0\}, \end{aligned} \tag{4}$$

where diag(D) denotes the column vector formed from the diagonal of D.

define the two linear operators acting on $\mathcal{S}_n$

$$\mathcal{K}(B) := \operatorname{diag}(B)\, e^t + e \operatorname{diag}(B)^t - 2B, \tag{5}$$

and

$$\mathcal{T}(D) := -\tfrac{1}{2} JDJ. \tag{6}$$

The operator $-2\mathcal{T}$ is an orthogonal projection onto $\mathcal{S}_C$; thus it is a self-adjoint idempotent.

Theorem 1 *The linear operators satisfy*

$$\mathcal{K}(\mathcal{S}_C) = \mathcal{S}_H,$$
$$\mathcal{T}(\mathcal{S}_H) = \mathcal{S}_C,$$

and $\mathcal{K}_{|\mathcal{S}_C}$ *and* $\mathcal{T}_{|\mathcal{S}_H}$ *are inverses of each other.*

$\square$

a hollow matrix D is EDM
if and only if
$B = \mathcal{T}(D)$ is positive semidefinite, ($B \succeq 0$)

Equivalently, D is EDM if and only if $D = \mathcal{K}(B)$, for some B with $Be = 0$ and $B \succeq 0$. In this case the embedding dimension r is given by the rank of B.

composite operators

$$\mathcal{K}_V(X) \;:=\; \mathcal{K}(VXV^t), \tag{7}$$

and

$$\mathcal{T}_V(D) \;:=\; V^t\mathcal{T}(D)V = -\tfrac{1}{2}V^tDV, \tag{8}$$

where V is defined in (2).

Lemma 1

$$\mathcal{K}_V(\mathcal{S}_{n-1}) = \mathcal{S}_H,$$
$$\mathcal{T}_V(\mathcal{S}_H) = \mathcal{S}_{n-1},$$

and $\mathcal{K}_V$ and $\mathcal{T}_V$ are inverses of each other on these two spaces.

$\square$

Corollary 1 *Suppose that V is defined as in (2). Then:*

$$\begin{aligned} \mathcal{K}_V(\mathcal{P}) &= EDM, \\ \mathcal{T}_V(EDM) &= \mathcal{P}. \end{aligned}$$

$\square$

Note that V as defined in (2) is not unique. In our code we use

$$V := \begin{bmatrix} y & y & \dots & y \\ 1+x & x & \dots, & x \\ x & 1+x & \dots, & x \\ \dots & \dots & \ddots & \dots \\ x & x & \dots, & 1+x \end{bmatrix}, \qquad (9)$$

where $x = \frac{-1}{n+\sqrt{n}}$ and $y = \frac{-1}{\sqrt{n}}$. With this choice, it can be easily verified that $V^t e = 0$, $V^t V = I$, and $VV^t = J$ as required by (2).

Program Formulation

Let A be a predistance matrix, i.e. symmetric, nonnegative (elementwise) and with 0 diagonal. Let H be an $n \times n$ symmetric nonnegative (elementwise) matrices and let $\|A\|_F = \sqrt{A^t A}$ denote the *Frobenius norm* of A. Consider the objective function

$$f(D) := \|H \circ (A - D)\|_F^2,$$

where $\circ$ denotes *Hadamard product*. The weighted, *closest Euclidean distance matrix problem* is

$$(CDM_0) \quad \mu^* := \begin{array}{cc} \text{min} & f(D) \\ \text{subject to} & D \in EDM. \end{array}$$

WLOG:
diag$(H) = 0$
graph of H is *connected*

define

$$f(X) := \|H \circ (A - \mathcal{K}_V(X))\|_F^2$$

$$(CDM) \quad \mu^* := \begin{array}{cc} \text{min} & f(X) \\ \text{subject to} & X \succeq 0. \end{array}$$

We can recover the optimal distance matrix using the optimal X and the relation

$$D = \mathcal{K}_V(X).$$

We can now state the dual problem

$$\begin{array}{cc} \max & f(X) + \langle y, b - \mathcal{A}(X)\rangle - \text{ trace } \Lambda X \\ \text{subject to} & \nabla f(X) - \mathcal{A}^* y - \Lambda = 0 \\ & \Lambda \succeq 0, (X \succeq 0). \end{array} \tag{10}$$

Strong duality holds (Slater's holds for the primal)

Slater's holds for the dual if and only if the graph of H is connected.

REMARKS:

An initial stepsize of 1.5 was taken; tests were also done with 2 and improved convergence was noted.

A centering parameter σ changes adaptively depending on the length of the stepsize, i.e. it decreases if the stepsize is large and increases to 1 if the stepsize is small.

The barrier parameter μ also changes adaptively; there is an aggressive decrease $\mu/4n$ if the stepsize is large; this decrease is smaller when small stepsizes are chosen.

Backtracking was initially done with the value .95 to be conservative, as is usually the case with SDP. However, tests were done with the aggressive .997 as is done in LP. This did not result in faster convergence. However, robustness was maintained.

The upper part of F', i.e. $[F'_{u1} \;\; F'_{u2}]$, need only be evaluated once at the beginning; while the lower (simpler) part of the operator needs to be evaluated at each iteration. Both evaluations take very little time compared to the solution of the least squares problem.

No predictor-corrector approach was used. Rather, centering was used to speed up convergence. However, the step lengths were consistently larger than 1 which suggests that no corrector step is needed, i.e. centering keeps the steps very close to the central path.

Though the initial starting points were feasible, dual feasibility was lost after the first iteration, i.e. unlike Newton's method, the Gauss-Newton direction does not maintain feasibility. However, complementarity and dual feasibility converged together to 0, i.e. at a similar rate. This might account for the strong convergence obtained with this direction.

The search direction was very robust, i.e. machine accuracy was attainable whenever asked for. This was the case even when the aggressive initial stepsize and reduction of μ were used.

Problem dimension was restricted to $n \leq 42$. This was because of the high cost of the least squares solution in MATLAB. (Done using qls, Version 1.6, from Pontus Matstoms, Linkoping University.) However, if one solves the least squares problem using, for example, a Lanczos approach, then one need never form the matrices which represent F' and thus can exploit sparsity and solve much larger problems. These tests are ongoing.

We have assumed, without loss of generality, that the graph of the matrix H was connected, i.e. if this was not the case then we can solve two smaller problems independently and more simply. In fact, one should always solve the smaller problems independently. Note that Slater's constraint qualification is always satisfied for the primal and so there is no duality gap. However, if Slater's constraint qualification fails for the dual (which is equivalent to the graph being disconnected), then the set of optimal solutions for the primal is unbounded and the algorithm has great difficulty converging. Here is an example of a class of problems where strong duality holds but interior-point methods have difficulty.