

This course will be concerned with semidefinite programming, the study of optimization problems that include constraints that certain matrices be positive semidefinite. We will confine ourselves to problems where all other constraints and functions are linear (or sometimes quadratic). The subject also goes by the name of linear matrix inequalities in control theory. I am generally assuming mathematical maturity, i.e., some linear algebra and real analysis. Familiarity with linear programming at a graduate level (like ORIE 6300) and numerical linear algebra (like CS 6210) will be useful but not essential.

I can be reached at 229 Rhodes Hall, at [mjt7@cornell.edu](mailto:mjt7@cornell.edu), [miketodd@orie.cornell.edu](mailto:miketodd@orie.cornell.edu), [miketodd@cs.cornell.edu](mailto:miketodd@cs.cornell.edu), or at 255-9135. There is a crude web site for the course at [people.orie.cornell.edu/~miketodd/or6327/or6327.html](http://people.orie.cornell.edu/~miketodd/or6327/or6327.html)

A rough outline for the course is:

- Introduction to SDP problems;
- Applications;
- Duality;
- the Central Path; and
- Algorithms (mostly interior-point methods).

We will be concentrating on SDP problems in primal standard form and in the corresponding dual form. The problem in primal form can be written as

$$(P) \quad \begin{aligned} \min_X \quad & C \bullet X \\ & A_i \bullet X = b_i, \quad i = 1, \dots, m, \\ & X \succeq 0, \end{aligned}$$

where the data  $C, A_i, i = 1, \dots, m$ , are real symmetric  $n \times n$  matrices while  $b$  is a real  $m$ -vector and the variable  $X$  is a real symmetric  $n \times n$  matrix. (There are also interesting SDP problems where instead the matrices above are complex Hermitian matrices instead of real symmetric, but we'll confine ourselves to the real case for simplicity.)

Here  $U \bullet V := \text{trace}(U^T V) = \sum_j \sum_k u_{jk} v_{jk}$  for any two matrices of the same dimensions, and  $U \succeq 0$  means that matrix  $U$  is symmetric and positive semidefinite (abbreviated psd), i.e., that  $z^T U z$  is nonnegative for all  $z$ . We also write  $U \succ 0$  to mean  $U$  is positive definite (pd), i.e.,  $z^T U z$  is positive for all nonzero  $z$ ;  $V \succeq W$  or  $W \preceq V$  for  $V - W \succeq 0$ ; and similarly  $V \succ W$  or  $W \prec V$  for  $V - W \succ 0$ .

Using the same data, we can construct an SDP problem in dual form:

$$\max_y \quad \begin{aligned} & b^T y \\ & \sum_{i=1}^m y_i A_i \preceq C; \end{aligned}$$

the constraint can alternatively be written  $C - \sum_{i=1}^m y_i A_i \succeq 0$ . This is a *linear matrix inequality* on the variable  $y$ : a requirement that a matrix depending linearly (by which we mean affinely!) on some variables be psd.

It is convenient to introduce a *slack matrix*  $S$  and rewrite the problem as

$$(D) \quad \begin{array}{l} \max_{y,S} \quad b^T y \\ \sum_i y_i A_i + S = C, \\ S \succeq 0. \end{array}$$

Here the variable  $y$  is a real  $m$ -vector while  $S$  is a real symmetric  $n \times n$  matrix.

We have been a little coy in calling (D) a problem in dual form rather than the problem dual to (P), but it will be a while before we establish that it is the Lagrangian dual or prove strong duality (under appropriate conditions). We will (almost) show weak duality below.

Our problems (P) and (D) above can be compared to the usual primal-dual pair for linear programming:

$$\begin{array}{ll} \min_x & c^T x \\ & a_i^T x = b_i, \quad i = 1, \dots, m, \\ & x \geq 0, \end{array} \quad \begin{array}{ll} \max_{y,s} & b^T y \\ & \sum_i a_i y_i + s = c, \\ & s \geq 0. \end{array}$$

Here the  $a_i$ 's,  $c$ ,  $x$ , and  $s$  are  $n$ -vectors, and the inequality is as usual interpreted componentwise.

Let us introduce a little more notation:

- $\mathbb{R}^m$  denotes real  $m$ -dimensional Euclidean space, viewed as the space of real column  $m$ -vectors (all our vectors will be columns, so we can distinguish between inner products  $u^T v$  and outer products (rank-one matrices)  $uv^T$ );  $\mathbb{R}^{m \times n}$  denotes the space of real  $m$  by  $n$  matrices;
- $\mathbb{M}^n$  denotes the space of real symmetric  $n \times n$  matrices, while
- $\mathbb{M}_+^n$  and  $\mathbb{M}_{++}^n$  denote its subsets of psd and pd matrices, respectively;
- $\mathcal{A}$  denotes the linear mapping from  $\mathbb{M}^n$  to  $\mathbb{R}^m$  defined by

$$\mathcal{A}X := (A_i \bullet X)_{i=1}^m, \text{ while}$$

- $\mathcal{A}^*$  denotes the (adjoint) linear mapping from  $\mathbb{R}^m$  to  $\mathbb{M}^n$  defined by

$$\mathcal{A}^*y := \sum_i y_i A_i$$

(so  $\mathcal{A}$  and  $\mathcal{A}^*$  replace the usual  $m \times n$  matrix  $A$  in the primal-dual pair of LPs above).

Now we can write our problems more compactly as

$$(P) \quad \begin{array}{l} \min_X \quad C \bullet X \\ \mathcal{A}X = b, \\ X \succeq 0, \end{array} \quad (D) \quad \begin{array}{l} \max_{y,S} \quad b^T y \\ \mathcal{A}^*y + S = C, \\ S \succeq 0. \end{array}$$

We can mirror the proof of weak duality for linear programming:

**Proposition 1** (*Weak Duality*) Suppose  $X$  and  $(y, S)$  are feasible for  $(P)$  and  $(D)$  respectively. Then

$$C \bullet X - b^T y = S \bullet X \geq 0,$$

so that objective function values of primal feasible solutions always dominate those of dual feasible solutions.

**Proof:** We again have an (almost) one-line proof:

$$\begin{aligned} C \bullet X - b^T y &= (\mathcal{A}^* y + S) \bullet X - (\mathcal{A} X)^T y = (\sum_i y_i A_i + S) \bullet X - ((A_i \bullet X)_1^m)^T y \\ &= S \bullet X \geq 0. \end{aligned}$$

Here we use the linearity in each argument of the expression  $U \bullet V$  and also the fact (established later this week) that  $U \bullet V$  is nonnegative when both  $U$  and  $V$  are psd.  $\square$

As a corollary, feasible  $X$  and  $(y, S)$  are optimal if their objective values are equal, which holds iff  $S \bullet X$  or  $X \bullet S$  is zero. (Thus “No SeX Please, We’re British” or “No eXcesS of primal objective over dual” are sufficient conditions for optimality.)

By stacking its columns one above another, we can convert any  $m \times n$  matrix  $P$  into a vector

$$\text{vec}(P) := (p_{11}; p_{21}; \dots; p_{m1}; p_{12}; p_{22}; \dots; p_{m2}; \dots; p_{1n}; p_{2n}; \dots; p_{mn}),$$

where we use the MATLAB-like notation  $(z_1; z_2; \dots; z_r)$  to denote a column vector with the appropriate components. (We also use this notation to stack column vectors or matrices columnwise.) Then if  $p = \text{vec}(P)$ ,  $q = \text{vec}(Q)$ , we have  $p^T q = P \bullet Q$ . For symmetric  $n \times n$  matrices, this representation is wasteful (and not onto), so instead we use

$$\text{svec}(U) := (u_{11}; \sqrt{2}u_{12}; u_{22}; \sqrt{2}u_{13}; \sqrt{2}u_{23}; u_{33}; \dots; u_{nn}),$$

where only the entries on and above the diagonal are used and the  $\sqrt{2}$  factors are chosen so that we still have  $\text{svec}(U)^T \text{svec}(V) = U \bullet V$  for symmetric matrices  $U$  and  $V$ . This embeds  $\mathbb{M}^n$  into  $\mathbb{R}^{n(n+1)/2}$  isometrically. Using this embedding, we can write  $(P)$  and  $(D)$  above as vector optimization problems, but usually we stick to the matrix notation for clarity. (As above, we try to reserve  $P$ ,  $Q$ ,  $L$ , and  $R$  for possibly nonsymmetric matrices; other upper-case Roman letters usually refer to symmetric matrices.)

A more general vector optimization problem is the conic programming problem (and its dual form)

$$\begin{aligned} \min_x \quad & c^T x \\ & a_i^T x = b_i, \quad i = 1, \dots, m, \\ & x \in K, \end{aligned} \qquad \begin{aligned} \max_{y,s} \quad & b^T y \\ & \sum_i a_i y_i + s = c, \\ & s \in K^*. \end{aligned}$$

where  $K$  is a closed convex cone in  $\mathbb{R}^n$  and  $K^*$  is the dual cone defined by

$$K^* := \{s \in \mathbb{R}^n : s^T x \geq 0 \text{ for all } x \in K\}.$$

It is easy to see that weak duality holds for this pair of problems also. In SDP, instead of  $K$  and  $K^*$ , we have the cone of psd matrices (in both places), so it will be key to show that this cone is self-dual, i.e., equal to its dual.

This is a universal form for convex optimization problems since any arbitrary convex problem,

$$(CP) \quad \begin{array}{ll} \min_x & f(x) \\ & x \in C, \end{array}$$

where  $f$  is a convex function and  $C$  a convex set, can be expressed as a conic program as follows:

i. We can assume that the objective function is linear, since  $(CP)$  is equivalent to

$$\min_{(x,\xi)} \quad \begin{array}{ll} & \xi \\ f(x) - \xi & \leq 0, \\ (x, \xi) & \in \bar{C}, \end{array}$$

where  $\bar{C} := C \times \mathbb{R}$ .

ii. We can assume that the constraints are in conic form, since

$$\min_x \quad \begin{array}{ll} c^T x \\ x \in C, \end{array}$$

is equivalent to

$$\min_{(x,\tau)} \quad \begin{array}{ll} c^T x \\ \tau = 1, \\ (x, \tau) \in K = \{(x, \tau) | \tau > 0, \frac{x}{\tau} \in C\}. \end{array}$$

We conclude the lecture with a little feel for what the cone of psd matrices looks like. First consider the case  $n = 2$ , and look at the set

$$\{(x; y; z) \in \mathbb{R}^3 : \begin{bmatrix} x & y \\ y & z \end{bmatrix} \succeq 0\}.$$

This set is defined by the inequalities  $x \geq 0$ ,  $z \geq 0$ , and  $xz \geq y^2$ , and the last can also be written  $((x+z)/2)^2 - ((x-z)/2)^2 \geq y^2$ , or, in the presence of the other two constraints,  $(x+z)/2 \geq \|((x-z)/2; y)\|_2$ . This can be viewed (after a little scaling) as a right circular cone in 3-dimensional space, with its axis in the direction  $(1; 0; 1)$ ; it is sometimes called the ice-cream, second-order, or Lorentz cone. We see that, while it has extreme rays, it has in fact an infinite number of them.

Next let us consider the case  $n = 3$ , but restrict ourselves to psd matrices whose diagonal entries are ones. These are *correlation matrices*, and the corresponding set is the *elliptope*. It is defined by

$$\{(x; y; z) \in \mathbb{R}^3 : \begin{bmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{bmatrix} \succeq 0\}.$$

Pictures of this “inflated tetrahedron” or “humbug” can be found at the two websites <http://www-user.tu-chemnitz.de/~helmborg/semidef.html>, [http://www.convexoptimization.com/dattorro/elliptope\\_and\\_fantope.html](http://www.convexoptimization.com/dattorro/elliptope_and_fantope.html), where you can see that it has four sharp “vertices” but otherwise a generally smooth boundary. The first of these websites is a very useful repository of information about SDP.

Next time we will present a very simple instance of the SDP in dual form to give an idea of its applicability, and list (and prove some of) a number of very useful facts about symmetric matrices.