PRINCETON UNIVERSITY
ORFE
Instructor:
Amir Ali Ahmadi

This lecture:

- Semidefinite programming (SDP)
    - Definition and basic properties
    - Review of positive semidefinite matrices
    - SDP duality
- SDP relaxations for nonconvex optimization
    - Combinatorial optimization
        - The Lovasz upper bound on the stability number of a graph
    - Polynomial optimization
        - Lower bounds for nonconvex polynomial minimization

- Semidefinite programming in standard form:

$$\min_{X \in S^{n\times n}} \ \mathrm{Tr}(CX)$$

$$\text{s.t.} \quad \mathrm{Tr}(A_i X) = b_i, \quad i=1,\dots,m,$$

$$X \succeq 0.$$

- Input data: $C \in S^{n\times n}$, $A_i \in S^{n\times n}$ $i=1,\dots,m$, $b_i \in \mathbb{R}$, $i=1,\dots,m$.

- Notation:
    - $S^{n\times n}$ denotes the space of $n \times n$ real symmetric matrices
    - "Tr" denotes the trace of a matrix; i.e., sum of its diagonal elements (which also equals the sum of its eigenvalues).

- SDP is an optimization problem over the space of symmetric matrices.
- It has two types of constraints:
    - Affine constraints in the entries of the decision matrix X.
    - A constraint forcing some matrix to be positive semidefinite.
        - This latter constraint is what distinguishes SDP from LP.

Why SDP?

The reasons will become more clear throughout this lecture, but here is a summary:

- SDP is a very natural generalization of LP.
- It is still a convex optimization problem (in the geometric sense).
- We can solve SDPs efficiently (in polynomial time to arbitrary accuracy). This is typically done by interior point methods, although other types of algorithms are also available.
- The expressive power of SDPs is much richer than LPs.
- When faced with a nonconvex optimization problem, SDPs typically produce much stronger bounds/relaxations than LPs do.
- Just like LP, SDP has a beautiful and well-established theory. Much of it mirrors the theory of LP.

Why the trace notation?

- It's just a convenient way of expressing affine constraints in the entries of a matrix:

$$Tr(AX) = \sum_{i,j} A_{ij} X_{ij} \qquad (A, X \in S^{n\times n})$$

e.g.,

$$Tr\left(\begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix}\begin{pmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{pmatrix}\right) = Tr\begin{pmatrix} a_{11}x_{11} + a_{12}x_{12} & * \\ * & a_{12}x_{12} + a_{22}x_{22} \end{pmatrix} = a_{11}x_{11} + 2a_{12}x_{12} + a_{22}x_{22}.$$

Positive semidefinite matrices (reminder)

Let $X \in S^{n\times n}$.

$X \succeq 0$ (read: X is positive semidefinite or psd)

$\Leftrightarrow$ $y^T X y \geq 0 \quad \forall y \in \mathbb{R}^n$.

$\Leftrightarrow$ All eigenvalues of X are ≥ 0.

$\Leftrightarrow$ Sylvester's criterion holds (see [CZ13], Sect. 3.4).

$\Leftrightarrow$ $X = MM^T$, for some $n \times k$ matrix M.

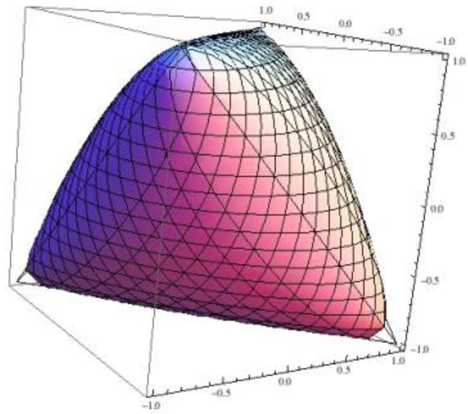
↳ Called a "Cholesky factorization".

Remark: $A \succeq B$ means $A - B \succeq 0$

Feasible set of SDPs

- The feasible set of an SDP is called a **spectrahedron**.
- Every polyhedron is a spectrahedron. (This is because every LP can be written as an SDP as we'll show shortly.)
- But spectrahedra are far richer geometric objects than polyhedra. (This is the reason why SDP is more powerful than LP.)
- Here is an example of a spectrahedron which is not a polyhedron:
  - The "elliptope":

$$\left\{(x,y,z) \;\middle|\; \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \succcurlyeq 0 \right\}$$



- Spectrahedra are always *convex sets*
  - The set of psd matrices forms a convex set (the proof was easy and given in a previous lecture).
  - Affine constraints define a convex set.
  - Intersection of convex sets is convex.

- When we say an SDP is a convex optimization problem, we mean this in the geometric sense.
  - The objective is an affine function of the entries of the matrix.
  - The feasible set is a convex set.
  - However, the feasible set is not written in the explicit functional form "convex function≤ 0, affine function$=0$".

- One can write an SDP as an infinite LP:
  - Replace $X \succcurlyeq 0$, with linear constraints $y_i^T X y_i \geq 0, \forall y \in \mathbb{R}^n$.
  - Can reduce this be a countable infinity by only taking $y \in \mathbb{Z}^n$.
- One can also write an SDP in standard functional form as a nonlinear program:
  - Replace $X \succcurlyeq 0$, with $2^n - 1$ minor inequalities coming from Sylvester's criterion.
- However, treating the matrix constraint $X \succcurlyeq 0$ directly is often the right thing to do.

LP as a special case of SDP

- Consider an LP:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ \text{s.t.} \quad & a_i^T x = b_i, \quad i = 1, \dots, m, \\ & x \geq 0. \end{aligned}$$

- This can be written as the following SDP (why?)

$$\min_{X} \quad \mathrm{Tr}\left(\begin{pmatrix} c_1 & & 0 \\ & \ddots & \\ 0 & & c_n \end{pmatrix} \begin{pmatrix} X_{11} & & 0 \\ & \ddots & \\ 0 & & X_{nn} \end{pmatrix} \right)$$

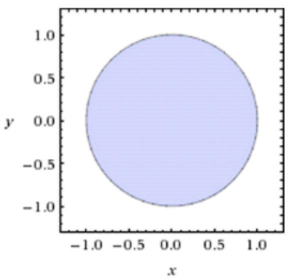
$$\text{s.t.} \quad \mathrm{Tr}\left(\begin{pmatrix} a_{i,1} & & 0 \\ & \ddots & \\ 0 & & a_{i,n} \end{pmatrix} \begin{pmatrix} X_{11} & & 0 \\ & \ddots & \\ 0 & & X_{nn} \end{pmatrix} \right) = b_i, \quad i = 1, \dots, m,$$

$$\begin{pmatrix} X_{11} & & 0 \\ & \ddots & \\ 0 & & X_{nn} \end{pmatrix} \succeq 0.$$

- So LP is really a special case of SDP where all matrices are diagonal -- positive semidefiniteness for a diagonal matrix simply means nonnegativity of its diagonal elements.

- Like we mentioned already, geometry of SDP is far more complex than LP.
- For example, unlike polyhedra, spectrahedra may have an infinite number of extreme points. Here is a simple example:

$$\{(x,y) \mid x^2 + y^2 \leq 1\} = \left\{ (x,y) \;\middle|\; \begin{pmatrix} 1+x & y \\ y & 1-x \end{pmatrix} \succeq 0 \right\}.$$



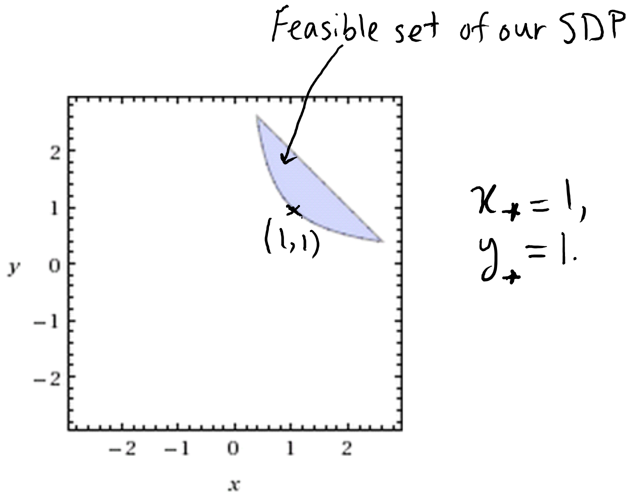
- This is the fundamental reason why SDP is not naturally amenable to "simplex-type" algorithms.
- On the contrary, interior points for LP (which we mentioned in class but did not cover) very naturally extend to SDP.

A toy SDP example and the CVX syntax

$$\min.\quad x+y$$
$$\text{s.t.}\quad \begin{bmatrix} x & 1 \\ 1 & y \end{bmatrix} \succeq 0,$$
$$x+y \leq 3.$$

```
cvx_begin
variables x y
minimize(x+y)
[x 1;1 y]==semidefinite(2);
x+y<=3;
cvx_end
```

- Exercise: write this SDP in standard form.
- Note: all SDPs can be written in standard form, but this transformation is not needed (most solvers do it automatically if they need to work with the standard form).



$x_* = 1,$
$y_* = 1.$

SDP Duality

- Just like LP, SDP has a nice duality theory.
- Every SDP has a dual, which itself is an SDP. The primal and dual SDPs bound the optimal value of each other.

Primal SDP

$$\min_{X \in S^{n \times n}} \ \mathrm{Tr}(CX)$$
$$\text{s.t.}\quad \mathrm{Tr}(A_i X) = b_i,\quad i=1,\dots,m,$$
$$X \succeq 0.$$

Dual SDP

$$\max_{y \in \mathbb{R}^m} \ b^T y$$
$$\text{s.t.}\quad \sum_{i=1}^{m} y_i A_i \preceq C.$$

Theorem (SDP weak duality). For any X feasible to the primal and any y feasible to the dual, we have $Tr(CX) \geq b^T y$.

- Can you prove this theorem?
- Hint. Prove the following fact first:

$$A \succeq 0,\ B \succeq 0 \Rightarrow Tr(AB) \geq 0.$$

- Contrast this theorem with weak duality of LP:

(P) $\min.\ c^T x$ $\quad$ (D) $\max\ b^T y$

$Ax = b$ $\quad$ $A^T y \leq c$

$x \geq 0$

Proof: $c^T x = x^T c \underset{\substack{\uparrow \\ A^T y \leq c \\ x \geq 0}}{\geq} x^T A^T y = y^T A x \underset{\substack{\uparrow \\ Ax = b}}{=} y^T b = b^T y.$ $\square$

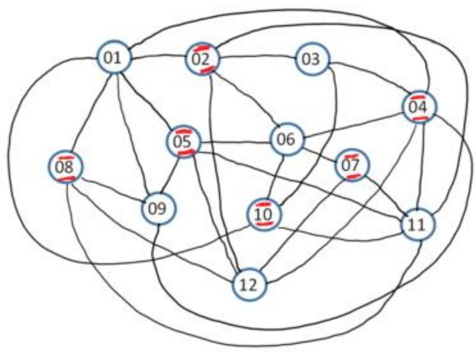
Theorem (SDP strong duality). If the primal and dual SDPs are both strictly feasible (i.e., if there exists a solution that makes the matrix which needs to be positive semidefinite, positive definite), then both problems achieve their optimal value and $Tr(CX_*) = b^T y_*$ (i.e., the optimal values match).

- Unlike LP, SDP strong duality needs some (mild) assumptions to hold--in this case strict feasibility.
- We do not prove this theorem as we didn't even prove strong duality for LP. But the proof can be found in most standard textbooks.

Applications of SDP in nonconvex optimization

- SDPs are among the most powerful algorithmic tools for finding good bounds on the optimal value of difficult nonconvex optimization problems.
- We present two such applications here, one in combinatorial optimization and one in polynomial optimization.

An SDP upper bound for the stable set number of a graph



- Stable set of a graph: a subset of the nodes that share no edges among them. Finding large stable sets has many applications in scheduling.
- The size of the largest stable set of a graph G is denoted by $\boldsymbol{\alpha(G)}$ and it's called the **stability number of the graph**.
- For example, we have $\alpha = 6$ in the graph shown here.
- Computing α is in general a very difficult problem. We will formalize what we mean by this in the next lecture. But think about how you would prove that there is no stable set of size > 6? Seems difficult.

- Yet, this is exactly what **convex relaxations** do!
- We have already seen an LP relaxation for this problem. We will review it again.
- We will also see a new SDP relaxation which produces a bound that's always no worse than the LP bound and often way better. This SDP is due to László Lovász.

Consider $G(V,E)$, $|V|=n$.

Integer program

$$\alpha(G) = \max_x \sum x_i$$
$$\text{s.t.} \quad x_i + x_j \leq 1, \text{ if } i,j \in E,$$
$$x_i \in \{0,1\}, \quad i=1,\dots,n.$$

Linear programming relaxation

$$LP_{OPT} = \max_x \sum x_i$$
$$\text{s.t.} \quad x_i + x_j \leq 1, \text{ if } i,j \in E,$$
$$0 \leq x_i \leq 1, \quad i=1,\dots,n.$$

<u>Claim</u>: $\alpha(G) \leq LP_{OPT}$.

Proof: Obvious (why?). $\square$

Semidefinite programming relaxation

the all ones matrix.

$$SDP_{OPT} = \max_{X \in S^{n \times n}} \operatorname{Tr}(JX)$$
$$\text{s.t.} \quad \operatorname{Tr}(X) = 1$$
$$X_{ij} = 0 \text{ if } ij \in E,$$
$$X \succeq 0.$$

Claim: $\alpha(G) \leq SDP_{OPT}$.

Proof:

Let $x \in \{0,1\}^n$ be the indicator vector of a maximum stable set S.

(e.g., $x = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$ means that nodes # 1, 4, 6 form a maximum stable set S.)

Let $|S|$ denote the cardinality of S (i.e., $|S| = \alpha(G)$).

Let

$$\hat{X} = \frac{1}{|S|} x x^T.$$

We claim: (i) $\hat{X}$ is feasible to the SDP, and (ii) $Tr(J\hat{X}) = \alpha(G)$.

This would finish the proof as SDP_{OPT} can only be larger than the objective value at a feasible point $\hat{X}$.

- Proof that $\hat{X}$ is feasible:

– $\hat{X} \succeq 0$ because $y^T\left(\frac{1}{|S|} x x^T\right) y = \frac{1}{|S|} (y^T x)(y^T x)^T = \frac{1}{|S|} (y^T x)^2 \geq 0, \ \forall y.$

– $\hat{X}_{ij} \overset{\text{why?}}{=} \frac{1}{|S|} x_i x_j \overset{x \text{ is an indicator vector of a stable set.}}{=} 0.$

– $Tr(\hat{X}) = \sum \hat{X}_{ii} = \frac{1}{|S|} \sum_{i=1}^n x_i^2 = \frac{1}{|S|} |S| = 1.$

- Proof that $Tr(J\hat{X}) = \alpha(G)$:

– $Tr(J\hat{X}) = \frac{1}{|S|} Tr(J x x^T) \overset{\text{why?}}{=} \frac{1}{|S|} |S|^2 = |S| = \alpha(G).$ □

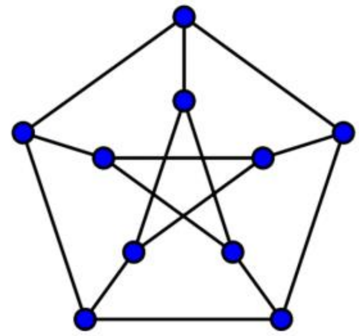
Examples.

$LP_{OPT} = 5$

$SDP_{OPT} = 4$

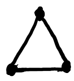
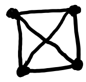
$\alpha(G) = 4.$

(find a stable set of size 4!)



- But the gap between the LP and SDP bound can be much larger.

Consider K_n: the complete graph on n nodes.

K_2  K_3  K_4  ...

$LP_{OPT} \geq n/2$ (set $x_i = 1/2 \ \forall i$ in LP and you are feasible.)

$SDP_{OPT} = 1.$

$$\max \ Tr(JX)$$

$$X = \begin{pmatrix} x_{11} & & 0 \\ & \ddots & \\ 0 & & x_{nn} \end{pmatrix} \succeq 0$$

$$Tr(X) = 1.$$

Note: Because off-diagonal is zero, $Tr(JX) = Tr(X) = 1. \Rightarrow SDP_{OPT} = 1.$

$\alpha(G) = 1$ (obvious).

$\therefore$ Gap between LP_{OPT} and SDP_{OPT} can be large.

(Not special to this instance. Quite typical.)

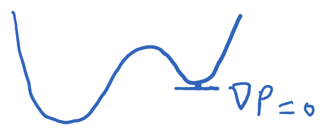
SDP lower bounds for nonconvex polynomial minimization*

- We are interested in this section in solving the following problem:

$$\min_{x \in \mathbb{R}^n} \; p(x),$$

where $p(x)$ is a multivariate polynomial.
(For example, $p(x) = 10x_1^4 - x_1^3 x_2 - x_2 + x_2^4$.)

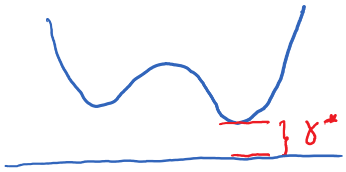
- This is unconstrained polynomial minimization. Without assuming that the objective function is convex, this problem can be very difficult. (You are asked to show on your homework that when p has degree 4, this problem is NP-hard.)
- Nevertheless, it turns out that by using semidefinite programming, we can find very good lower bounds on the optimal value and in fact often solve the problem globally. This approach also works in the constrained case, but we restrict attention to the unconstrained case for simplicity.
- Note that in absence of convexity, the descent methods we've seen earlier in class (e.g., gradient descent, Newton, etc.) can get stuck in local minima.



- The way SDP solves this problem is fundamentally different than the way local search methods work.
- Instead of looking for a feasible point with lowest objective value, we look for the *largest possible lower bound.*
- The starting point is the following simple observation.

Lec13p12, ORF363/COS323

$$\min_{x \in \mathbb{R}^n} p(x) = \begin{bmatrix} \max_{\gamma \in \mathbb{R}} & \gamma \\ \text{s.t.} & p(x) - \gamma \geq 0 \quad \forall x \in \mathbb{R}^n \end{bmatrix} := \gamma^*.$$



The constraint in the optimization problem to the right requires us to force a certain polynomial $(p(x) - \gamma)$ to be *nonnegative.*

Defn. A polynomial p is *nonnegative* if $p(x) \geq 0, \forall x \in \mathbb{R}^n$.

- In general, testing if a polynomial (of degree 4 or larger) is nonnegative is NP-hard.

Defn. A polynomial p is a *sum of squares (sos)* if $p(x) = \sum_i q_i^2(x)$, for some other polynomials q_i.

- Clearly, if p is sos, then p is nonnegative. (The converse is not true in general.)
- However, we can check if a polynomial is sos by solving an SDP! (See the theorem on the next page).
- This allows us to obtain lower bounds on the optimization problem above by solving the optimization problem below, which as we explain next can be reformulated as an SDP:

$$\gamma_{sos} := \begin{bmatrix} \max_{\gamma} & \gamma \\ \text{s.t.} & \text{"}p(x) - \gamma \text{ is sos."} \end{bmatrix}$$

We have $\gamma_{sos} \leq \gamma^*$ (why?).

Theorem 1 *A multivariate polynomial p in n variables and of degree $2d$ is a sum of squares if and only if there exists a positive semidefinite matrix Q (often called the Gram matrix) such that*

$$p(x) = z^T Q z,$$

where z is the vector of monomials of degree up to d

$$z = [1, x_1, x_2, \ldots, x_n, x_1x_2, \ldots, x_n^d].$$

Proof ($\Rightarrow$): $Q \succcurlyeq 0 \xrightarrow{\text{Cholesky}} Q = M^T M$.

$\Rightarrow p(x) = z^T(x) M^T M z(x) = (M z(x))^T (M z(x)) = \| M z(x) \|^2$. $\square$

Example: Consider the task proving nonnegativity of the polynomial

$$\begin{aligned} p(x) \quad = \quad & x_1^4 - 6x_1^3x_2 + 2x_1^3x_3 + 6x_1^2x_3^2 + 9x_1^2x_2^2 - 6x_1^2x_2x_3 - 14x_1x_2x_3^2 \\ & +4x_1x_3^3 + 5x_3^4 - 7x_2^2x_3^2 + 16x_2^4. \end{aligned}$$

Since this is a form (i.e., a homogeneous polynomial), we take

$$z = (x_1^2, x_1x_2, x_2^2, x_1x_3, x_2x_3, x_3^2)^T.$$

One feasible solution to the SDP in (2) is given by

$$Q = \begin{pmatrix} 1 & -3 & 0 & 1 & 0 & 2 \\ -3 & 9 & 0 & -3 & 0 & -6 \\ 0 & 0 & 16 & 0 & 0 & -4 \\ 1 & -3 & 0 & 2 & -1 & 2 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 2 & -6 & 4 & 2 & 0 & 5 \end{pmatrix}.$$

Upon a decomposition $Q = \sum_{i=1}^{3} a_i a_i^T$, with $a_1 = (1, -3, 0, 1, 0, 2)^T$, $a_2 = (0, 0, 0, 1, -1, 0)^T$, $a_3 = (0, 0, 4, 0, 0, -1)^T$, one obtains the sos decomposition

$$p(x) = (x_1^2 - 3x_1x_2 + x_1x_3 + 2x_3^2)^2 + (x_1x_3 - x_2x_3)^2 + (4x_2^2 - x_3^2)^2.$$

Notes:

Semidefinite programming is not covered in [DPV08]. There are sections on SDP in [CZ13] and our reference book [BV04], but you are only responsible for a good understanding of the content of my notes.

References:

- [BV04] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
  http://stanford.edu/~boyd/cvxbook/

- [CZ13] E. K. P. Chong and S. H. Zak, An Introduction to Optimization, 4th Edition, Wiley Press, 2013.

- [DPV08] S. Dasgupta, C. Papadimitriou and U. Vazirani, Algorithms, McGraw Hills, 2008.