C&O 355
Lecture 18

N. Harvey

Topics

- Semi-Definite Programs (SDP)
- Solving SDPs by the Ellipsoid Method
- Finding vectors with constrained distances

LP is great, but…

- Some problems cannot be handled by LPs

- **Example:** Find vectors $v_1,\ldots,v_{10} \in \mathbb{R}^{10}$ s.t.
  - All vectors have unit length: $\|v_i\| = 1 \;\; \forall i$
  - Distance between vectors is: $\|v_i - v_j\| \in [1/3, 5/3] \;\; \forall i \neq j$
  - Sum-of-squared distances is maximized

- Not obvious how to solve it.
- Not obvious that LP can help.

- This problem is **child's play** with SDPs

How can we make LPs more general?

- An (equational form) LP looks like:

$$\begin{array}{lll} \max & c^{\mathsf{T}} x & \\ \text{s.t.} & Ax & = b \\ & x & \geq 0 \end{array}$$

- In English:
  - Find a non-negative vector x
  - Subject to some linear constraints on its entries
  - While maximizing a linear function of its entries
- What object is “more general” than vectors?
- How about **matrices**?

Generalizing LPs: Attempt #1

- How about this?

$$\begin{aligned} \max \quad & c^\mathsf{T} X & \\ \text{s.t.} \quad & AX & = b \\ & X & \geq 0 \end{aligned}$$

- In English:
  - Find a “non-negative matrix” X
  - Subject to some linear constraints on its entries
  - While maximizing a linear function of its entries
- Does this make sense? Not quite…
  - What is a “non-negative matrix”?
  - Objective function $c^\mathsf{T}X$ is not a scalar
  - AX is not a vector

What is a “non-negative matrix”?

- Let’s define “non-negative matrix” by “symmetric, positive semi-definite matrix”.
  - So our “variables” are the entries of an n x n symmetric matrix.
  - The constraint “$X \geq 0$” is replaced by “X is PSD”

- **Note**: The constraint “X is PSD” is quite weird. We’ll get back to this issue.

Vectorizing the Matrix

- A dxd matrix can be viewed as a vector in $\mathbb{R}^{d^2}$. (Just write down the entries in some order.)

$$\boxed{\begin{array}{cc}1 & 2\\ 3 & 4\end{array}} \Rightarrow \boxed{\begin{array}{c}1\\ 2\\ 3\\ 4\end{array}}$$

- A dxd **symmetric** matrix can be viewed as a vector in $\mathbb{R}^{d(d+1)/2}$.

$$X = \boxed{\begin{array}{cc}1 & 2\\ 2 & 4\end{array}} \Rightarrow x = \boxed{\begin{array}{c}1\\ 2\\ 4\end{array}}$$

- **Our notation:** X is a dxd symmetric matrix, and x is the corresponding vector.

Semi-Definite Programs

$$\begin{aligned} \max \quad & c^{\mathsf{T}} x \\ \text{s.t.} \quad & Ax = b \\ & \boxed{X \text{ is PSD}} \end{aligned}$$

This constraint looks suspiciously non-linear

- Where
  - $x \in \mathbb{R}^n$ is a vector
  - A is a m×n matrix, $c \in \mathbb{R}^n$ and $b \in R^m$
  - X is a d×d symmetric matrix, where n = d(d+1)/2, and x is vector corresponding to X.
- In English:
  - Find a symmetric, positive semidefinite matrix X
  - Subject to some linear constraints on its entries
  - While maximizing a linear function of its entries

Review of Eigenvalues

- A complex dxd matrix M is **diagonalizable** if $M = U D U^{-1}$, where D and U have size dxd, and D is diagonal.
  (This expression for M is called a "**spectral decomposition**")
- The diagonal entries of D are called the **eigenvalues** of M, and the columns of U are corresponding **eigenvectors**.
- An eigenvector of M is any vector y s.t. $My = \lambda y$, for some $\lambda \in \mathbb{C}$.

- **Fact:** Every **real symmetric** matrix M is diagonalizable.
  In fact, we can write $M = U D U^T$, where D and U are **real** dxd matrices, D is diagonal, and $U^T = U^{-1}$. (U is "orthogonal")
- **Fact:** For real symmetric matrices, it is easy to compute the matrices U and D. ("Cholesky Factorization", very similar to Gaussian Elim.)
- **Summary:** Real symmetric matrices have real eigenvalues and eigenvectors and they're easily computed.

Positive Semidefinite Matrices (again)

- Assume M is a symmetric, d x d matrix
- **Definition 1:** M is PSD iff $\exists$V s.t. M = $V^T V$.
- **Definition 2:** M is PSD iff $y^T M y \geq 0$ $\forall y \in \mathbb{R}^d$.
- **Definition 3:** M is PSD iff all eigenvalues are ≥ 0.

- **Claim:** Definition 3 $\Rightarrow$ Definition 1.
- **Proof:** Since M symmetric, M = U D U^T where D is diagonal and its diagonal entries are the eigenvalues.
  Let W be diagonal matrix W=$D^{1/2}$, i.e., $W_{i,i} = \sqrt{D_{i,i}}$
  Then M = U^T W W U = $U^T W^T$ W U = $(WU)^T$ (WU) = V^T V, where V = WU. $\square$

Positive Semidefinite Matrices (again)

- Assume M is symmetric
- **Definition 1:** M is PSD iff $\exists$V s.t. M = $V^T V$.
- **Definition 2:** M is PSD iff $y^T My \geq 0\ \forall y \in \mathbb{R}^d$.
- **Definition 3:** M is PSD iff all eigenvalues are $\geq$ 0.
- **Notation:** Let M[S,T] denote submatrix of M with row-set S and column-set T.
- **Definition 4:** M is PSD iff det(M[S,S])$\geq$0 $\forall$S.
- Definitions 1-3 are very useful. Definition 4 is less useful.

The PSD Constraint is Convex

- **Claim:** The set C = { x : X is PSD } is a convex set.
- **Proof:**

By Definition 2, X is PSD iff $y^T X y \geq 0 \; \forall y \in \mathbb{R}^d$.

Note: $y^T X y = \sum_{i=1}^{d} \sum_{j=1}^{d} X_{i,j} y_i y_j$. For each fixed y, this is a linear inequality involving entries of X.

Each inequality defines a half-space, and is convex.

So $C = \{ \; x \; : \; y^T X y \geq 0 \;\; \forall y \in \mathbb{R}^n \; \}$

So C is the intersection of an (infinite!) collection of convex sets, and hence is convex. (See Asst 4, Q1)

■

Remark: This argument also shows that C is closed.

What does the PSD set look like?

- Consider 2x2 symmetric matrices.
- M = $\boxed{\begin{matrix} \alpha & \beta \\ \beta & \gamma \end{matrix}}$
- Let C = { x : X is PSD }.
- By Definition 4, M is PSD iff det(M[S,S])$\geq$0 $\forall$S.
- So C = { (α,β,γ) : $\alpha\geq$0, $\gamma\geq$0, $\alpha\gamma$-$\beta^2\geq$0 }.

Note: Definitely not a polyhedral set!

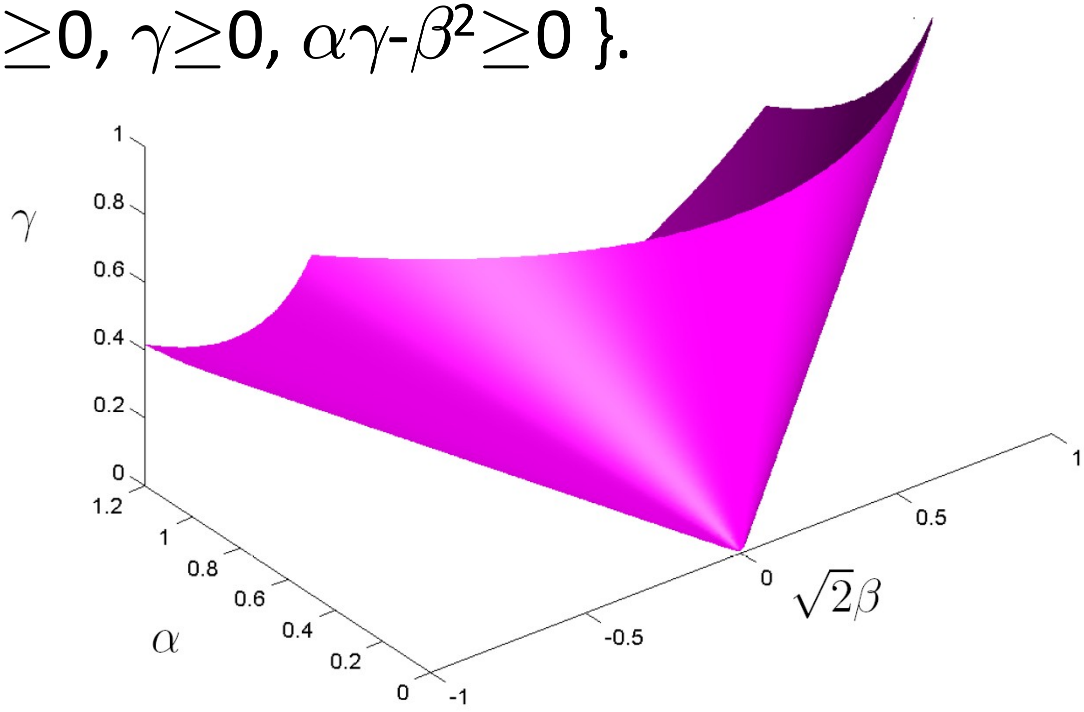


Image from Jon Dattorro "Convex Optimization & Euclidean Distance Geometry"

Semi-Definite Programs

$$\begin{array}{ll} \max & c^\top x \\ \text{s.t.} & Ax = b \\ & \cancel{X \text{ is PSD}} \\ & y^\top X y \geq 0 \;\; \forall y \in \mathbb{R}^d \;\} \text{ Definition 2} \end{array}$$

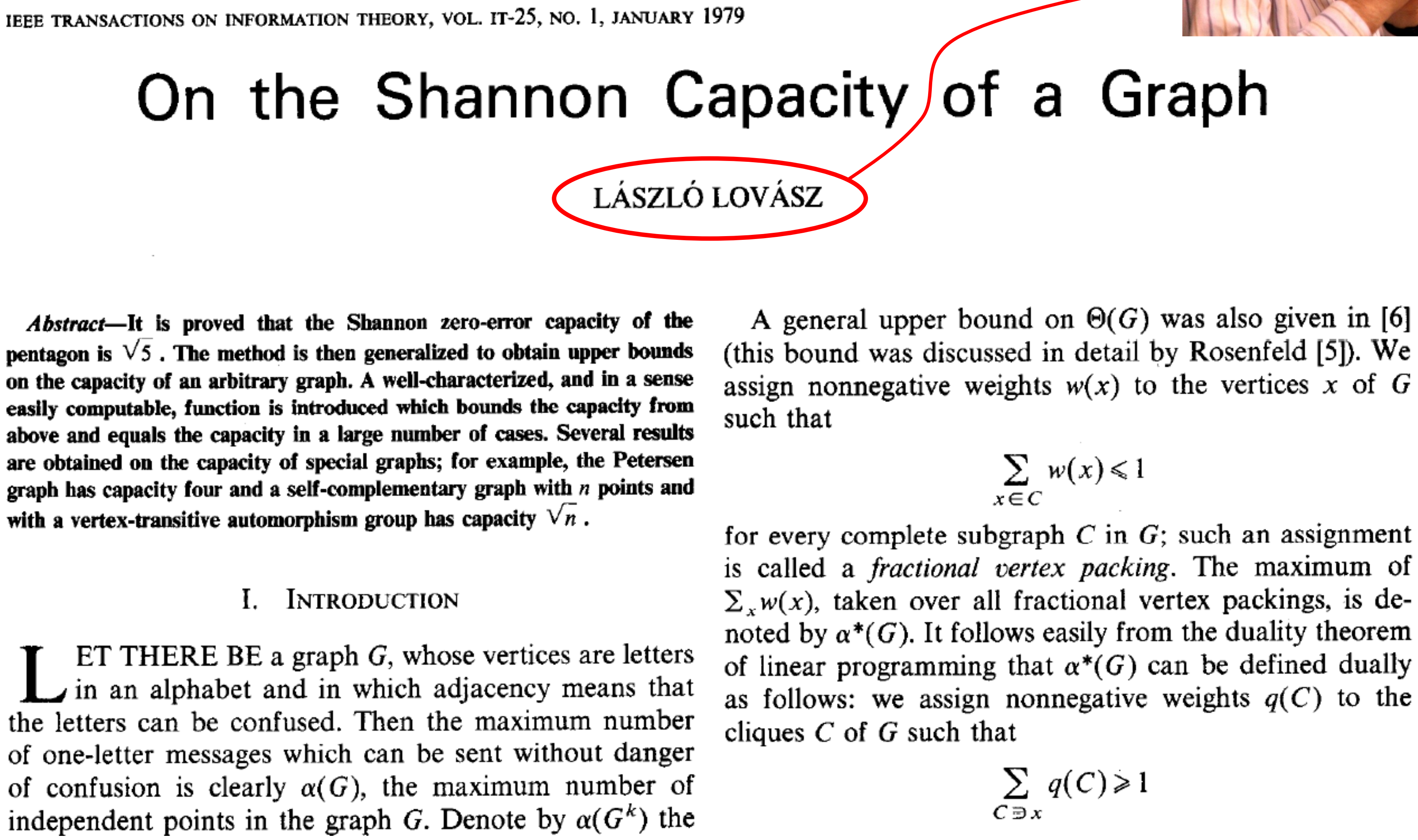
This constraint looks suspiciously non-linear

- Where
  - $x \in \mathbb{R}^n$ is a vector
  - A is a m$\times$n matrix, $c \in \mathbb{R}^n$ and $b \in R^m$
  - X is a d$\times$d symmetric matrix, where n = d(d+1)/2, and x is vector corresponding to X.
- Replace suspicious constraint with Definition 2
- This is a convex program with **infinitely many** constraints!

History of SDPs

- Implicitly appear in this paper:

We met him in Lecture 12



IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-25, NO. 1, JANUARY 1979

On the Shannon Capacity of a Graph

LÁSZLÓ LOVÁSZ

Abstract—It is proved that the Shannon zero-error capacity of the pentagon is $\sqrt{5}$. The method is then generalized to obtain upper bounds on the capacity of an arbitrary graph. A well-characterized, and in a sense easily computable, function is introduced which bounds the capacity from above and equals the capacity in a large number of cases. Several results are obtained on the capacity of special graphs; for example, the Petersen graph has capacity four and a self-complementary graph with n points and with a vertex-transitive automorphism group has capacity $\sqrt{n}$.

I. INTRODUCTION

LET THERE BE a graph G, whose vertices are letters in an alphabet and in which adjacency means that the letters can be confused. Then the maximum number of one-letter messages which can be sent without danger of confusion is clearly $\alpha(G)$, the maximum number of independent points in the graph G. Denote by $\alpha(G^k)$ the

A general upper bound on $\Theta(G)$ was also given in [6] (this bound was discussed in detail by Rosenfeld [5]). We assign nonnegative weights $w(x)$ to the vertices x of G such that

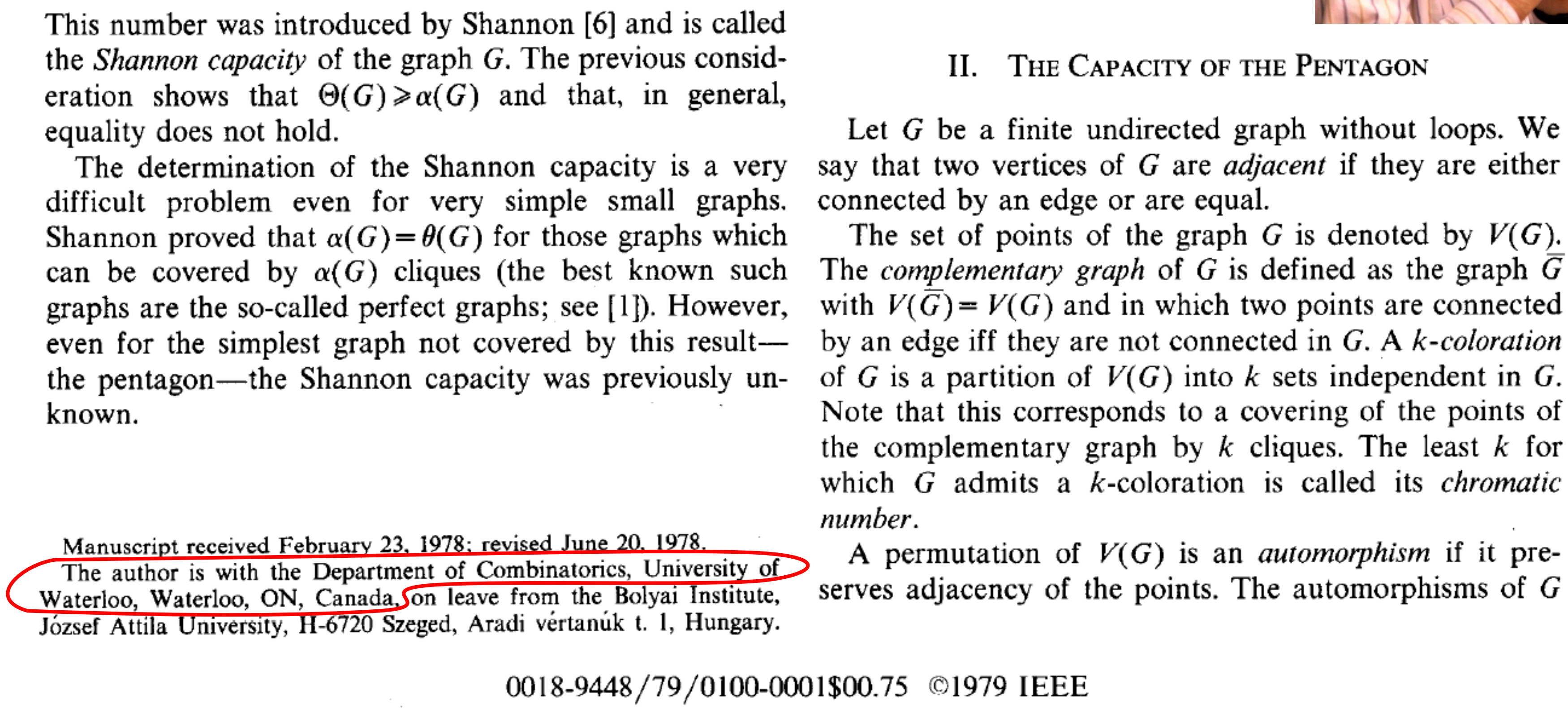
$$\sum_{x \in C} w(x) \leqslant 1$$

for every complete subgraph C in G; such an assignment is called a *fractional vertex packing*. The maximum of $\Sigma_x w(x)$, taken over all fractional vertex packings, is denoted by $\alpha^*(G)$. It follows easily from the duality theorem of linear programming that $\alpha^*(G)$ can be defined dually as follows: we assign nonnegative weights $q(C)$ to the cliques C of G such that

$$\sum_{C \ni x} q(C) \geqslant 1$$

History of SDPs

- Scroll down a bit…



This number was introduced by Shannon [6] and is called the *Shannon capacity* of the graph G. The previous consideration shows that $\Theta(G) \geqslant \alpha(G)$ and that, in general, equality does not hold.

The determination of the Shannon capacity is a very difficult problem even for very simple small graphs. Shannon proved that $\alpha(G)=\theta(G)$ for those graphs which can be covered by $\alpha(G)$ cliques (the best known such graphs are the so-called perfect graphs; see [1]). However, even for the simplest graph not covered by this result—the pentagon—the Shannon capacity was previously unknown.

Manuscript received February 23, 1978; revised June 20, 1978.

The author is with the Department of Combinatorics, University of Waterloo, Waterloo, ON, Canada, on leave from the Bolyai Institute, József Attila University, H-6720 Szeged, Aradi vértanúk t. 1, Hungary.

II. The Capacity of the Pentagon

Let G be a finite undirected graph without loops. We say that two vertices of G are *adjacent* if they are either connected by an edge or are equal.

The set of points of the graph G is denoted by $V(G)$. The *complementary graph* of G is defined as the graph $\overline{G}$ with $V(\overline{G})=V(G)$ and in which two points are connected by an edge iff they are not connected in G. A *k-coloration* of G is a partition of $V(G)$ into k sets independent in G. Note that this corresponds to a covering of the points of the complementary graph by k cliques. The least k for which G admits a k-coloration is called its *chromatic number*.

A permutation of $V(G)$ is an *automorphism* if it preserves adjacency of the points. The automorphisms of G

0018-9448/79/0100-0001$00.75 ©1979 IEEE

SDPs were discovered in U. Waterloo C&O Department!

Solving Semi-Definite Programs

$$\begin{aligned} \max \quad & c^\intercal x \\ \text{s.t.} \quad & Ax = b \\ & y^\intercal X y \geq 0 \;\; \forall y \in \mathbb{R}^d \end{aligned}$$

- There are **infinitely many** constraints!
- But having many constraints doesn't scare us: we know the **Ellipsoid Method**.
- To solve the SDP, we:
  - Replace objective function by constraint $c^\intercal x \geq \alpha$, and binary search to find (nearly) optimal alpha.
  - Need to design a separation oracle.

Separation Oracle for SDPs

Solve: $$\left\{ \begin{array}{lll} c^{\mathsf{T}} x & \geq \alpha & \\ a_i^{\mathsf{T}} x & = b_i & \forall i \\ y^{\mathsf{T}} X y & \geq 0 & \forall y \in \mathbb{R}^d \end{array} \right\}$$

Separation Oracle

Is $z \in P$?

If not, find a vector a s.t. $a^{\mathsf{T}}x < a^{\mathsf{T}}z \quad \forall x \in P$

- Easy to test if $Az=b$, and if $c^{\mathsf{T}}z \geq \alpha$
  - If not, either a_i or $-a_i$ or c gives the desired vector a
- How can we test if $y^{\mathsf{T}} Z y \geq 0 \quad \forall y$?

Separation Oracle for SDPs

Solve: $$\left\{ \begin{array}{ll} c^{\mathsf{T}} x & \geq \alpha \\ a_i^{\mathsf{T}} x & = b_i \ \ \forall i \\ y^{\mathsf{T}} X y & \geq 0 \ \ \forall y \in \mathbb{R}^d \end{array} \right\}$$

Separation Oracle

Is $z \in P$?

If not, find a vector a s.t. $a^T x < a^T z \ \ \forall x \in P$

- How can we test if $y^T Z y \geq 0 \ \ \forall y$?
- **Key trick:** Compute eigenvalues & eigenvectors!

If all eigenvalues $\geq$ 0, then Z is PSD and $y^T Z y \geq 0 \ \forall y$.

If y is a non-zero eigenvector with eigenvalue λ<0, then

$$Zy = \lambda y \quad \Rightarrow \quad y^T Z y = y^T \lambda y = \lambda \, y^T y = \lambda \, \|y\|^2 < 0$$

Thus the constraint $y^T X y \geq 0$ is violated by Z!

Separation Oracle for SDPs

Solve: $\left\{ \begin{array}{lll} c^{\mathsf{T}} x & \geq \alpha & \\ a_i^{\mathsf{T}} x & = b_i & \forall i \\ y^{\mathsf{T}} X y & \geq 0 & \forall y \in \mathbb{R}^d \end{array} \right\}$

Separation Oracle

Is $z \in P$?

If not, find a vector a s.t. $a^{\mathsf{T}} x < a^{\mathsf{T}} z \;\; \forall x \in P$

- **Summary:** SDPs can be solved (approximately) by the Ellipsoid Method, in polynomial time.
  There are some issues relating to irrational numbers and the radii of balls containing and contained in feasible region.
- SDPs can be solved efficiently in practice (approximately), by Interior Point Methods.

- **Example:** Find vectors $v_1,\ldots,v_{10} \in \mathbb{R}^{10}$ s.t.
  - All vectors have unit length: $\|v_i\| = 1 \;\; \forall i$
  - Distance between vectors is: $\|v_i - v_j\| \in [1/3, 5/3] \;\; \forall i \neq j$
  - Sum-of-squared distances is maximized

- Why does this example relate to SDPs?
- **Key observation:** PSD matrices correspond directly to vectors and their dot-products.
- Given vectors $v_1,\ldots,v_d$ in $\mathbb{R}^d$, let V be the d$\times$d matrix whose ith column is v_i.
- Let $X = V^T V$. Then X is PSD and $X_{i,j} = v_i^T v_j \;\; \forall i,j$.

- **Example:** Find vectors $v_1,\ldots,v_{10} \in \mathbb{R}^{10}$ s.t.
  - All vectors have unit length: $\|v_i\| = 1 \;\; \forall i$
  - Distance between vectors is: $\|v_i - v_j\| \in [1/3, 5/3] \;\; \forall i \neq j$
  - Sum-of-squared distances is maximized

- **Key observation:** PSD matrices correspond directly to vectors and their dot-products.
- Given vectors $v_1,\ldots,v_d$ in $\mathbb{R}^d$, let V be the $d \times d$ matrix whose ith column is v_i.
- Let $X = V^T V$. Then X is PSD and $X_{i,j} = v_i^T v_j \;\; \forall i,j$.
- Conversely, given a $d \times d$ PSD matrix X, find spectral decomposition $X = U D U^T$, and let $V = D^{1/2} U$.
- To get vectors in $\mathbb{R}^d$, let v_i = ith column of V.
- Then $X = V^T V \;\; \Rightarrow \;\; X_{i,j} = v_i^T v_j \;\; \forall i,j$.

- **Example:** Find vectors $v_1,\ldots,v_{10} \in \mathbb{R}^{10}$ s.t.
  - All vectors have unit length: $\|v_i\| = 1 \;\forall i$
  - Distance between vectors is: $\|v_i - v_j\| \in [1/3, 5/3] \;\forall i \neq j$
  - Sum-of-squared distances is maximized

- **Key observation:** PSD matrices correspond directly to vectors and their dot-products:

  If X PSD, it gives vectors $\{ v_i : i=1,\ldots,d \}$ where $X_{i,j} = v_i^T v_j$.

- Also, distances and lengths relate to dot-products:

  $\|u\|^2 = u^T u$ and $\|u-v\|^2 = (u-v)^T(u-v) = u^T u - 2v^T u + v^T v$

- So our example is solved by the SDP:

max $\Sigma_{i,j} (X_{i,i} - 2X_{i,j} + X_{j,j})$	(i.e., $\Sigma_{i,j} \|v_i - v_j\|^2$)
s.t. $X_{i,i} = 1$	(i.e., $\|v_i\| = 1$)
$X_{i,i} - 2X_{i,j} + X_{j,j} \in [1/9, 25/9]$	(i.e., $\|v_i - v_j\| \in [1/3, 5/3]$)
X is PSD	

- **Example:** Find vectors $v_1,\ldots,v_{10} \in \mathbb{R}^{10}$ s.t.
  - All vectors have unit length: $\|v_i\| = 1\ \forall i$
  - Distance between vectors is: $\|v_i - v_j\| \in [1/3, 5/3]\ \forall i \neq j$
  - Sum-of-squared distances is maximized

- Our example is solved by the SDP:

max $\Sigma_{i,j} (X_{i,i} - 2X_{i,j} + X_{j,j})$	(i.e., $\Sigma_{i,j} \|v_i - v_j\|^2$)
s.t. $X_{i,i} = 1$	(i.e., $\|v_i\| = 1$)
$X_{i,i} - 2X_{i,j} + X_{j,j} \in [1/9, 25/9]$	(i.e., $\|v_i - v_j\| \in [1/3, 5/3]$)
X is PSD	

- Note objective function is a linear function of X's entries
- Note constraints are all linear inequalities on X's entries

SDP Summary

- Matrices can be viewed as vectors.
- Can force a matrix to be PSD using infinitely many linear inequalities.
- Can test if a matrix is PSD using eigenvalues. This also gives a separation oracle.
- PSD matrices correspond to vectors and their dot products (and hence to their distances).
- So we can solve lots of optimization problems relating to vectors with certain distances.
  - In applications of SDP, it is often not obvious why the problem relates to finding certain vectors…