Semidefinite programming and computational aspects of entanglement **IHP Fall 2017**

Lecture 1: November 3

Lecturer: Anand Natarajan *Scribe: Aram and Anand*

Roughly speaking, this class is about h_{Sep} and ω^*.

1.1 Convex optimization

1.1.1 Basics

A convex set in $\mathbb{R}^n$ is a set $S \subseteq \mathbb{R}^n$ such that for any $x, y \in K$ and $\lambda \in [0,1]$, $\lambda x + (1-\lambda)y \in K$. Typically we will consider compact sets but this is not necessary.

A key property of convex sets is the *separating hyperplane theorem*. It states that any point outside of K can be separated from K with a hyperplane. Mathematically,

$$\begin{aligned} x \notin K \qquad \Leftrightarrow \qquad & \exists b \text{ s.t. } b^T x > 1 \\ & b^T y \leq 1, \forall y \in K. \end{aligned} \tag{1.1}$$

The vector b defines the separating hyperplane. The set of all such possible hyperplanes is itself a convex set, called the *polar* of K.

$$K* = \{b : b^T y \leq 1, \forall y \in K.\}. \tag{1.2}$$

There are two basic questions we can ask.

- Membership: given K, x, is $x \in K$?
- Optimizing linear functions: given K, y, calculate $\max_{x \in K} \langle y, x \rangle$.

A useful fact: the max of a linear function is always achieved on the boundary of the set. (Draw a picture here).

As always when we deal with real numbers, it's important to be careful about numerical precision. Let $S(K, \delta)$ be the exterior δ-ball of K:

$$S(K, \delta) := \{x : \exists y \in K, \|y - x\| \leq \delta\},$$

and let $S(K, -\delta)$ be the interior δ-ball of K:

$$S(K, -\delta) := \{x \in K : S(x, \delta) \subseteq K\}.$$

Then the "correct" versions of the basic problems are

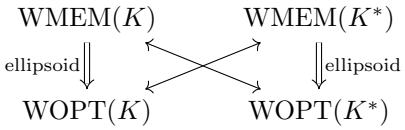
- WMEM: given K, x, δ, decide whether $x \in S(K, -\delta)$ or $x \notin S(K, \delta)$, promised that one of the two is the case.
- WOPT: given K, $b \in \mathbb{R}^n$, δ, find $y \in S(K, \delta)$ such that $b^T x \leq b^T y + \delta$ for all $x \in S(K, -\delta)$.

Note that for WOPT we allow two types of error, each (for convenience) parametrized by the same tolerance δ: there is δ error in the actual value of the objective function (i.e. $b^T y$) and there is a δ error in the definition of K, i.e. we can replace with the sets $S(K, \pm\delta)$ in ways that add error.

These two problems are equivalent to each other, in the sense that given an oracle for one, we can solve the other in polynomial time and poly invocations of the oracle. To prove this we first observe that WMEM for K^* is equivalent to WOPT for K, and vice versa. To go from WMEM(K^*) to WOPT(K) we use binary search to search for the minimum $\gamma > 0$ such that $\gamma b \in K*$, and use $1/\gamma$ for our estimate of WOPT(K). In the other direction we are given b and we report it belonging to $K*$ if our algorithm for WOPT(K) reports a value $\leq 1 + \delta$. (This is being a bit sloppy in terms of the precision parameter.)

Additionally note that $K^{**} = K$, i.e. the polar of the polar is the original set. This statement is basically equivalent to the separating hyperplane theorem.

So far we have all the single arrows in the following diagram.



We still need to prove the double arrows labeled "ellipsoid." This reduction is done with the ellipsoid algorithm. (Other methods, like interior point, are also possible. Their performance is better but are less universal and have more complicated proofs.)

Here is a sketch of the ellipsoid algorithm. To solve WOPT, binary search over guesses for optimum value γ, and check whether $K' = K \cap \{x : b^T x \geq \gamma\}$ is nonempty. Thus we have reduced to WMEM on K', which we would like to use our WMEM(K) algorithm for. To do this, draw a big ellipsoid containing K'. (Here we use the fact that K is compact, and indeed further assume that we know an upper bound on the size of a ball centered at the origin containing K. Later we will also need a lower bound on volume of K.) Then, iteratively, check whether the center of the ellipsoid is in K', using WMEM(K). Otherwise, find a separating hyperplane between K' and the center; in some cases our WMEM algorithm will return a hyperplane, and if not there is a reduction to find one, which we will not discuss here. Resize your ellipsoid so it's on the right side of the separating hyperplane and still contains K', and repeat, until you find a point in K' or your ellipsoid is too small. For this last step we assume we know a lower bound on the volume of K. Since the ellipsoid's volume shrinks by a bounded amount in each iteration, this guarantees termination in polynomial time.

1.1.2 Examples

1.1.2.1 Efficiently solvable examples

Linear programming (LP). Given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, find $\max c^T x$ such that $Ax \leq b$. Here we interpret the $\leq$ entrywise, i.e. every entry in the vector on the LHS is $\leq$ the corresponding vector on the RHS. This can be thought of as optimizing a linear function ($c^T x$) over the polytope $\{x : Ax \leq b\}$.

Semidefinite programming (SDP). Primal form:

$$\begin{aligned} &\max_X \operatorname{Tr}(CX) \\ &\text{subject to } \operatorname{Tr}(A_i X) \leq b_i \ \forall i \in [m] \\ &X \succeq 0. \end{aligned} \tag{1.3}$$

Dual form:

$$\begin{aligned} &\min_y \langle y, b \rangle \\ &\text{such that } \sum_i y_i A_i \succeq C \\ &y \geq 0. \end{aligned} \tag{1.4}$$

Where does the dual come from? Suppose we want to find an upper bound on (1.3). For any $y \in \mathbb{R}^m, y \geq 0$ we can take linear combinations of the $\operatorname{Tr}(A_i X) \leq b_i$ inequalities to obtain

$$\sum_i y_i \operatorname{tr}(A_i X) \leq \langle y, b \rangle. \tag{1.5}$$

On the one hand, we would like to minimize the RHS, i.e. minimizing $\langle y, b \rangle$. On the other hand, we would like the LHS to upper bound (1.3). One way to guarantee this holds for any $X \succeq 0$ is to have $\sum_i y_i A_i \succeq C$. Thus we obtain (1.4). This implies that (1.4) is $\geq$ (1.3), a fact known as *weak duality*. Somewhat surprisingly, in many cases the two are equal, and when this happens we call this situation *strong duality*.

Trivial example: computing the operator norm of a matrix M:

$$\|M\| = \begin{array}{c} \max_X \operatorname{Tr}(MX) \\ \text{s.t. } \operatorname{Tr}(X) = 1 \cdot \\ X \succeq 0 \end{array}$$

This can equivalently be thought of as optimizing a homogenous quadratic polynomial over the unit sphere. Indeed if M is symmetric then $\|M\| = \max_v \sum_{i,j} M_{i,j} v_i v_j$ subject to the constraint $\sum_i v_i^2 = 1$.

We will see later some connections between SDPs and higher-degree polynomials.

1.1.2.2 Some harder examples

Max cut (aka optimizing a quadratic function over the cube): Given a graph adjacency matrix A,

$$\text{MAXCUT}(A) = \max_{x \in \{\pm 1\}^n} \frac{1}{4} \operatorname{Tr}\big(A(J - xx^T)\big),$$

where J is the all-ones matrix. For simplicity, let's ignore the constant term and flip the sign so we get

$$\text{MAXCUT}'(A) = \min_{x \in \{\pm 1\}^n} \operatorname{Tr}\big(A(xx^T)\big),$$

(N.b. there is a constant term which we have ignored) At first glance, this doesn't look convex! But we can convexify it. The *cut polytope* is

$$K = \operatorname{conv}(\{xx^T : x \in \{\pm 1\}^n\}).$$

Then we are minimizing a linear function over K:

$$\text{MAXCUT}'(A) = \min_{X \in K} \operatorname{Tr}(AX).$$

However, this is actually NP-hard!

This looks an awful lot like an SDP though. What makes it different? I claim that MAXCUT is equivalent to the following "almost SDP":

$$\text{MAXCUT}'(A) = \begin{array}{c} \min_X \operatorname{Tr}(AX) \\ \text{s.t.} X_{ii} = 1 \ \forall i \\ X \succeq 0 \\ \operatorname{rank}(X) = 1. \end{array}$$

Why? If X is rank one and PSD, it has the form $X = xx^T$ for some vector x (not necessarily of unit norm). The constraints $X_{ii} = 1$ imply that $x_i^2 = 1$, so the entries of x_i are in $\{\pm 1\}$.

Thus, if we wanted to approximate MAXCUT, we could *relax* the problem by removing the rank consraint on X. This would give an SDP, whose answer is always an upper bound on the true value.

Another example of a very similar flavor is nuclear/atomic norms. Roughly speaking, if you could minimize a linear function over the set

$$K = \operatorname{conv}(\{x \otimes y \otimes z : \|x\| = \|y\| = \|z\| = 1\}),$$

then you could compute low-rank decompositions of tensors with three legs.

1.2 Intro to separable states

We will introduce h_{Sep}, also known as BSS ("best separable state").

For some Hilbert spaces $\mathcal{H}_A, \mathcal{H}_B$, define

$$\text{Sep} = \text{Sep}(\mathcal{H}_A, \mathcal{H}_B) = \operatorname{conv}\left\{|\alpha\rangle\langle\alpha| \otimes |\beta\rangle\langle\beta| : \langle\alpha|\alpha\rangle = \langle\beta|\beta\rangle = 1, \alpha \in \mathcal{H}_A, \beta \in \mathcal{H}_B\right\}. \tag{1.6}$$

These correspond to unentangled mixed states. In other words ρ is entangled iff $\rho \notin \text{Sep}$.

By contrast for pure states $|\psi\rangle$ is product if $|\psi\rangle = |\alpha\rangle \otimes |\beta\rangle$ and is entangled if it cannot be written in this form. In this case a state is correlated iff it is entangled. Mixed-state entanglement is more complicated because we want to define classically correlated states as being not entangled.

Using Sep we can define

$$h_{\text{Sep}}(M) = \max_{\rho \in \text{Sep}} \text{Tr}[M\rho]. \tag{1.7}$$

Computing h_{Sep} corresponds to the problem $OPT(\text{Sep})$. We will typically work with WOPT, i.e. allowing some small error.

Physical Interpretation. Let $0 \preceq M \preceq I$, so $\{M, I - M\}$ is a two-outcome measurement, whose outcomes we call "yes" and "no". Then $h_{\text{Sep}}(M)$ is the largest possible probability of obtaining "yes" from an unentangled state when we perform measurement M.

Entanglement witnesses. These are operators $M \succeq 0$ for which $h_{\text{Sep}}(M) < \|M\|$. This means that there exist ρ for which $\text{Tr}[M\rho] > \max_{\sigma \in \text{Sep}} \text{Tr}[M\sigma]$. Thus ρ is entangled and M is a "witness" to this fact. Bell inequalities (once we fix our measurement strategies) are examples of this; thus, such M can be carried out by remote non-communicating parties.

Polynomial formulation of h_{Sep}. Since the maximum is achieved by an extreme point we have

$$\begin{aligned} h_{\text{Sep}}(M) = \max_{\alpha,\beta} & \sum_{i,j,k,l} M_{ijkl} \alpha_i \beta_j \alpha_k^* \beta_l^* \\ \text{s.t.} & \sum_i |\alpha_i|^2 = 1 \\ & \sum_i |\beta_i|^2 = 1 \\ & \alpha \in \mathbb{C}^{n_1}, \beta \in \mathbb{C}^{n_2} \end{aligned} \tag{1.8}$$

The PPT relaxation. PPT means "positive partial transpose." Since $h_{\text{Sep}}(M)$ is hard to compute we'd like relaxations. One relaxation is $h_{\text{Sep}}(M) \leq \|M\|$. Equivalently, we can say that $\text{Sep} \subset \mathcal{D} = \{\rho : \text{Tr}[\rho] = 1, \rho \succeq 0\}$, where $\mathcal{D}$ is the set of all states (density matrices). However, this can be far from tight. Taking $M = |\Phi_n\rangle\langle\Phi_n|$ where $|\Phi_n\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^n |i\rangle \otimes |i\rangle$ we have $\|M\| = 1$ and $h_{\text{Sep}}(M) = 1/n$. In general the best bound we can prove is $h_{\text{Sep}(n_1,n_2)}(M) \geq \|M\| / \min(n_1, n_2)$.

PPT is a better relaxation. Observe that if

$$\sigma = \sum_i p_i \, |\alpha_i\rangle\langle\alpha_i| \otimes |\beta_i\rangle\langle\beta_i| \,, \tag{1.9}$$

with $p_i \geq 0, \sum_i p_i = 1$ then

$$\sigma^\Gamma = \sum_i p_i \, |\alpha_i\rangle\langle\alpha_i| \otimes (|\beta_i\rangle\langle\beta_i|)^T = \sum_i p_i \, |\alpha_i\rangle\langle\alpha_i| \otimes |\beta_i\rangle\langle\beta_i|^* \succeq 0. \tag{1.10}$$

In general $X^\Gamma = (\text{id} \otimes T)(X)$ is called the "partial transpose," since we transpose only the second system. We say that a density matrix $\sigma \in \text{PPT}$ if $\sigma^\Gamma \succeq 0$. We have just shown that $\text{Sep} \subset \text{PPT}$.

This is better than the trivial $\text{Sep} \subset \mathcal{D}$ relaxation, equivalently, $\text{PPT} \subsetneq \mathcal{D}$. To see this, let $\rho = |\Phi_2\rangle\langle\Phi_2|$. Then

$$\rho = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \rho^\Gamma = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \not\succeq 0 \tag{1.11}$$

Thus we have an improved SDP relaxation

$$\begin{aligned} h_{\mathrm{Sep}}(M) \leq h_{\mathrm{PPT}}(M) = \max_{\rho} & \operatorname{Tr}[M\rho] \\ \text{s.t. } & \operatorname{Tr}[\rho] = 1 \\ & \rho \succeq 0 \\ & \rho^{\Gamma} \succeq 0 \end{aligned} \tag{1.12}$$

(By the way, the set of PPT states is $\{\rho \in \mathcal{D} : \rho^{\Gamma} \geq 0\}$.)

1.3 Intro to correlations

Alice and Bob share a possibly entangled state ρ. A referee sends them questions a, b respectively and they return answers x, y. The input-output statistics form some conditional probability distribution $p(x, y|a, b)$. The set of all such conditional p.d.'s that can be achieved with entangled states is called

$$\mathcal{Q} = \{p(x, y|a, b) : \text{realizable by quantum Alice, Bob}\}. \tag{1.13}$$

(We will be more formal later.) This is a convex set because given two strategies using states ρ_1, ρ_2 they can use the state $\lambda |1\rangle\langle 1| \otimes \rho_1 + (1-\lambda) |2\rangle\langle 2| \otimes \rho_2$, and use the first register to tell them which measurements to use. In this way they can simulate the mixture of any two strategies in $\mathcal{Q}$.

Suppose the referee samples questions from distribution $\pi(a, b)$ and evaluates the answers by assigning the score $V(x, y, a, b) \in \{0, 1\}$. This is called a *non-local game* $\mathcal{G} = (\pi, V)$ and its *value* is

$$\omega^*(\mathcal{G}) = \max_{p \in \mathcal{Q}} \sum_{a,b,x,y} \pi(a, b) V(x, y, a, b) p(x, y|a, b). \tag{1.14}$$

It corresponds to the maximum probability with which a game can be won with entangled strategies. For simplicity, we can write $\mathcal{G}(a, b, x, y) = \pi(a, b)V(x, y, a, b)$.

This looks somewhat different from what we've seen so far. But in the "polynomial optimization" picture it seems closer. We can write

$$\begin{aligned} \omega^*(\mathcal{G}) = \max_{\rho, \{A_a^x\}, \{B_b^y\}} & \sum_{x,y,a,b} \mathcal{G}(a, b, x, y) \operatorname{Tr}[\rho(A_a^x \otimes B_b^y)] \\ & \text{s.t. } A_a^x \succeq 0, \forall a, x \\ & \quad B_b^y \succeq 0, \ \forall b, y \\ & \quad \sum_x A_a^x \leq I, \ \forall a \\ & \quad \sum_y B_b^y \leq I, \ \forall b \\ \leq \max_{\{A_a^x\}, \{B_b^y\}} & \left\| \sum_{x,y,a,b} \mathcal{G}(a, b, x, y) A_a^x B_b^y \right\| \\ & \text{s.t. } A_a^x \succeq 0, \forall a, x \\ & \quad B_b^y \succeq 0, \ \forall b, y \\ & \quad \sum_x A_a^x \leq I, \ \forall a \\ & \quad \sum_y B_b^y \leq I, \ \forall b \\ & \quad [A_a^x, B_b^y] = 0 \, \forall a, b, x, y \end{aligned} \tag{1.15}$$