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

Lecture 2: November 9

Lecturer: Aram Harrow *Scribe: Anand*

(Notes available at http://web.mit.edu/aram/www/teaching/sdp.html.)

As introduced last time, we're interested in the computational complexity of h_{Sep} and ω^*. To understand the computational complexity of a problem, there are two ways to attack it: algorithms and computational hardness. (We can also look at structural properties of the problem, like behavior of random instances, but mostly we'll stick to these two things.)

Today: h_{Sep} and computational hardness. This is especially interesting because, as we'll see, this is one of the rare problems whose complexity lies in the intermediate region between polynomial and exponential: our best lower-bounds are quasipolynomial, and understanding this would help underestand other intermediate problems like Unique Games, which have been a hot topic in complexity theory.

Recall

$$\text{Sep} = \text{Sep}(d, 2) = \text{conv}\{\alpha \otimes \beta : \alpha, \beta \in D_d\},$$

where D_d is the set of d-dim dnesity matrices.

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

We study hardness in terms of NP-completeness or NP-hardness. Recall that NP is the class of problems where there exists a witness to the solution that you can verify in polynomial time, using a classical computer. QMA is the quantum version of this class: there exists a *quantum state* which is a witness to the solution of the problem, which can be checked by a quantum computer in polynomial time.

Example of a function in NP: let $g \in \mathsf{P}$ be an efficiently computable function, and take

$$f(x) = \bigvee_y g(x, y).$$

If $f(x) = 1$, then there exists a witness for this fact, namely a value of y s.t. $g(x, y) = 1$. And this can be verified in polynomial time.

Example: Given M, λ, $f(M, \lambda) = 1$ if $\|M\| \geq \lambda$. The witness for this is a vector $|\psi\rangle \neq 0$ s.t. $\|M|\psi\rangle\| \geq \lambda \||\psi\rangle\|$. Hence this problem is in NP. But it's also in P: just diagonalize the matrix.

Example: MAX-CUT. Here the witness is the cut, but there's no efficient algorithm we know of. In fact, MAX-CUT is NP-complete: it is contained in NP, and it is "NP-hard": for any other function $f \in \mathsf{NP}$, if there is an efficient algorithm for MAX-CUT, then there exists an efficient algorithm for f.

An important subtlety: you have to talk about the "threshold" version of MAX-CUT, i.e deciding whether there exists a cut that cuts at least k edges. If $k = |E|$, then the problem is easy, because you just check whether the graph is bipartite by coloring it. However for general k, it's hard, and in fact, even to approximate the maximum number of edges cut, it can still be NP-hard.

Some more examples of NP-complete problems: 3-SAT.

$$x \in \{0, 1\}^n, \quad \phi(x) = \bigwedge_{i=1}^m C_i, \quad C_i = x_{i_1} \vee x_{i_2} \vee x_{i_3},$$

where each of the variables in C_i can optionally be negated. The function is $f(\phi) = 1$ iff $\exists x$ s.t. $\phi(x) = 1$. This is clearly in NP; it turns out to also be NP-complete, essentially because any verification circuit for an NP-problem can be turned into a 3-SAT formula. (This is called the Cook-Levin theorem.) Incidentally, it's sufficient to take $m = O(n)$ (in the worst case, m could be $\Omega(n^3)$).

These problems are instances of CSPs: Fix an alphabet Σ, and consider strings $x \in \Sigma^n$ and clauses $C_1, \ldots, C_m$, where each clause is a function $C_i : \Sigma^k \to \{0, 1\}$, and define the function

$$\phi(x) = \frac{1}{m} \sum_{i=1}^m C_i(x_{S_i}),$$

where S_i is a subset of $[n]$ of size k. The *value* of the CSP is

$$\text{val}(\phi) = \max_x \phi(x).$$

An important subclass of these is *unique games*. Here $k = 2$, and the clauses obey the property that $\forall x \in \Sigma$, there exists a unique $y \in \Sigma$ such that $C_i(x, y) = 1$. E.g. MAX-CUT: if a set a vertex to be 1, its neighbor has to be 0, and vice versa. This suggests a greedy strategy to find an assignment: just set a variable and follow the constraints, but you can easily run into contradictions (e.g. MAX-CUT on a triangle—you can't cut all the edges).

Another exampleof UG:

$$Ax = b \quad \text{over } \mathbb{F}_q.$$

What's special about UG? It's trivial to determine if the value is 1 or not: the greedy algorithm mentioned above works. But *approximating* the value can still be hard. (This is unlike 3-SAT, where deciding the value is 1 itself is NP-hard). The UG problem, for (Σ, c, s), is, given constraints $C_1, \ldots, C_m$, determine if

$$\begin{array}{rl} \max_x \phi(x) \geq c & (\text{"completeness"}) \\ \leq s & (\text{"soundness"}) \end{array}.$$

By the above discussion, UG is in P if $c = 1, s < 1$. The *Unique Games Conjecture* states that $\forall \epsilon$, there exists an alphabet Σ, such that $\text{UG}(\Sigma, c = 1 - \epsilon, s = \epsilon)$ is NP-complete. Note that this is a very weak notion of approximation!

Why should we care about this? Let's return to MAX-CUT. You'd think that the best approximation ratio you could get is 1/2. But actually, Goemans and Williamson found an algorithm (essentially the SDP relaxation we showed last time), for which you can get an approx ratio of $\alpha_{GW} \approx 0.878$. If the UGC is true, then you can show that this is optimal: achieving approx ratio $\alpha_{GW} + \epsilon$ is NP-complete, so the GW algorithm would be optimal. Subsequently, Raghavendra showed that this story holds for *all* CSPs: there's a simple SDP, derived in the same way as the GW algorithm, which is optimal if UGC is true.

Below, we will see a connection between h_{Sep} and UGC. Aram's dream: make enough progress on UGC to force classical computer scientists to learn QI. Haven't fully succeeded!

But before that, another example of a CSP: *3-coloring*. Can you color a graph with 3 colors such that adjacent vertices have different colors? Turns out that this is NP-complete. Why? Reduce from 3-SAT. Suppose we're given a 3-SAT instance ϕ with clauses $\{C_1, \ldots, C_m\}$. We want to create a graph G that's 3-colorable iff ϕ is satisfiable. To do this, we want to map assignments of ϕ to colorings of the graph.

Diagram showing gadgets here.

Finally, in passing: an important theorem about CSPs is the PCP theorem. This says that for (essentially) any NP-complete CSP, there's always some constants c, s such that a c, s approximation to $\phi(x)$ is also NP-complete.

Moving on to h_{Sep}. As input, we're given a matrix M s.t. $0 \leq M \leq I$, and the problem is to distinguish whether

$$h_{\text{Sep}(d,2)}(M \geq c \quad \text{or} \quad \leq s.$$

This problem is in NP as long as $c - s$ is not extremely tiny, because you can always give a description of ρ as a witness. This means that it takes time at most 2^d to solve it (because you could just search over all ρs). We consider the following cases:

- $c = 1, s = 1 - 1/\operatorname{poly}(d)$: this is as hard as a 3-SAT instance of length $O(d/\log d)$. So this case matches our crude upper bound of 2^d time.
- $c = 1, s = 1/2$: this is at least as hard as a 3-SAT instance of length $O(\log^2 d/\operatorname{poly}\log\log d)$. This gives a lower-bound on runtime of $d^{\log d}$, which is barely more than polynomial time.

  More intriguingly, there exists an algorithm by Brandao, Christandl, and Yard for h_{Sep} that computes $h_{\text{Sep}}(M) \pm \epsilon$ in time $\exp\left(O(\frac{\log^2 d}{\epsilon^2})\right)$, **if** M has the form

$$M = \sum_i A_i \otimes B_i, \quad A_i, B_i \geq 0, \sum_i A_i \leq I, B_i \leq I \, \forall i.$$

(These Ms are called "1-way LOCC".) This runtime matches the lower bound, but they apply to different classes of matrices! Is it just a coincidence that we get this strange, quasipolynomial scaling in both cases?

- The condition $M \leq I$ sets the scale of the problem in terms of $\|M\|$. So you can think of the previous cases as asking for an approx to $h_{\text{Sep}}(M)$ up to some multiplicative factor in $\|M\|$. But what about approximating $h_{\text{Sep}}(M)$ up to some multiplicative factor in $h_{\text{Sep}}(M)$ itself? For instance, take $c = 100/d, s = 10/d$. In this case, the problem is as hard as teh *small-set expansion problem*, which is beleived to have the same complexity as UGC.

INTERMISSION
We'll now give sketches of the proofs of these cases.

Inverse-polynomial soundness For the first case, the best known results achieve $c = 1, s = 1 - O(1/d \log d)$. This line of work was started by Blier and Tapp, improved by Chiesa and Forbes, and the current best result was reached by Le Gall, Nakagawa, and Nishimura.

Let $\tau : [n] \to \{R, G, B\}$ be a coloring, and consider the state

$$|\phi_\tau\rangle = \frac{1}{\sqrt{n}} \sum_{v \in [n]} |v\rangle \otimes |\tau(v)\rangle .$$

Ideally, I should be able to learn how good my coloring is by measuring two copies of the above state. To this end, we design a measurement M which is the average of 3 measurements: $M = \frac{M_1 + M_2 + M_3}{3}$, each of which checks a different property of the state. These measurements satisfy the property that if τ is a valid 3-coloring, $\text{tr}[M_i(\phi_\tau \otimes \phi_\tau)] = 1$ for each i (implying that if there exists a valid coloring, then $h_{\text{Sep}}(M) = 1$). We also want the "soundness" condition to hold: if there is *no* valid 3-coloring, then for any separable state $\phi_1 \otimes \phi_2$, $\text{tr}[M\phi_1 \otimes \phi_2] \leq 1 - 1/\operatorname{poly}(n)$. We achieve this by designing the measurements M_1, M_2, M_3 to have the following properties:

1. Take $M_1 = (I + F)/2$, where $F |x\rangle \otimes |y\rangle = |y\rangle \otimes |x\rangle$. This is the projector onto the symmetric subspace. The idea is that this will force $\phi_1 = \phi_2$, approximately. To see this, compute:

   $$\text{tr}[M_1(\phi_1 \otimes \phi_2)] = \frac{1 + |\langle \phi_1 | \phi_2 \rangle|^2}{2}.$$

   If this trace is bigger than $1 - 1/\operatorname{poly}(n)$, then $|\phi_1\rangle \approx_{1/\operatorname{poly}(n0} |\phi_2\rangle$.

2. Next, we want to enforce that the states look like valid coloring state $|\phi_\tau\rangle$. We do this using the "consistency" test M_2. In words: measure ϕ_1, ϕ_2 and obtain outcomes v_1, c_1, v_2, c_2. Then do the following checks:

   - If $v_1 = v_2$, then require $c_1 = c_2$.
   - If $(v_1, v_2) \in E$, then require $c_1 \neq c_2$ (this is the only part of the construction that depends on the graph).

   As a matrix,

   $$\begin{aligned} M_2 = &\sum_v |v\rangle\langle v| \otimes |v\rangle\langle v| \otimes \sum_c |c\rangle\langle c| \otimes |c\rangle\langle c| \\ &+ \sum_{(v_1,v_2) \in E} |v_1\rangle\langle v_1| \otimes |v_2\rangle\langle v_2| \otimes \sum_{c_1 \neq c_2} |c_1\rangle\langle c_1| \otimes |c_2\rangle\langle c_2| \\ &+ \sum_{(v_1,v_2) \notin E, v_1 \neq v_2} |v_1\rangle\langle v_1| \otimes |v_2\rangle\langle v_2| \otimes I. \end{aligned}$$

   (Note that we reordered the registers so that the vertex registers come first, and then the color registers.)

3. Finally, we want to make sure that the coloring states are superpositions over all the vertices. This is the "uniformity test." Define

$$|u_v\rangle = \frac{1}{\sqrt{n}} \sum_{v \in [n]} |v\rangle$$
$$|v_c\rangle = \frac{|R\rangle + |G\rangle + |B\rangle}{\sqrt{3}}$$
$$M_3 = \left(I \otimes \left(I - |v_c\rangle \langle v_c| \right) + |u_v\rangle \langle u_v| \otimes |u_c\rangle \langle u_c| \right)^{\otimes 2}.$$

Now, obtaining $s = 1 - 1/\operatorname{poly}(d)$ is not too hard, and getting $1 - O(1/d \log d)$ requires more work. But basically it requires carefully analyzing the soundness of each of the individual tests.

Why can't we do better than this construction? After all, using PCP, we can get NP-hardness for $c - s$ constant for 3-coloring, so you can imagine that all bad colorings actually violate a constant fraction of the edges. The problem is the consistency test: most of the time you're not even going to sample an edge when you measure. It seems impossible to improve on this with a test of this form.

Constant soundness Based on what I said earlier, we need to reduce from a 3-SAT instance of length n to an h_{Sep} instance over $d = 2^{\tilde{O}(\sqrt{n})}$. We will instead show an intermediate result, which is to reduce to h_{Sep} with $d = n$ and $O(\sqrt{n})$ subsystems. Note that the total dimension is the same, the difference is that we require that the state be product across more cuts. This result was due to Aaronson, Beigi, Drucker, Fefferman, and Shor in 2008.

In the inverse-poly case, we could basically use any CSP where clauses act on 2 variables at a time. Here we'll need to be more particular about the CSP. Let's use a boolean alphabet, and for each assignment $x = \{0,1\}^n$, consider a state

$$|\phi_x\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (-1)^{x_i} |i\rangle .$$

Using states of this form, it's not obvious how to check that x satisfies 3-SAT. However, it turns out that we *can* check that x satisfies a CSP called 2-out-of-4-SAT. In this CSP, each clause has 4 bits, and is true iff exactly 2 of the bits are true. This is NP-complete (can reduce from 3-SAT). Moreover, we can assume $O(n)$ clauses, where each bit shows up in only $O(1)$ clauses (do this by copying any variable that appears too many times, and forcing the copies to be equal), and that the fraction of satisfied clauses is either 1 or ≤ 0.9 (using the PCP theorem).

Now, if I'm given a state of the form $|\phi_x\rangle$, here's how to measure a clause C_j involving bits $i_{j_1}, i_{j_2}, \ldots, i_{j_4}$. First, project with

$$\Pi_j = |i_{j_1}\rangle \langle i_{j_1}| + \cdots + |i_{j_4}\rangle \langle i_{j_4}| .$$

By regularity, $\sum_j \Pi_j \leq O(1) \cdot I$, so I can complete these projectors into a measurement

$$M = \{\frac{\Pi_1}{m}, \ldots, \frac{\Pi_m}{c}, R\},$$

and the chance of getting the remainder outcome R is bounded by a constant:

$$\operatorname{tr}[R\rho] \leq 0.99.$$

This means, that with some constant probability, I've projected my state down to

$$\frac{(-1)^{x_{i_{j_1}}} |i_{j_1}\rangle + \cdots + (-1)^{x_{i_{j_4}}} |i_{j_4}\rangle}{2},$$

for some clause. To check the clause, measure whether this state is orthogonal to $\frac{1}{2}(|i_{j_1}\rangle + \cdots + |i_{j_4}\rangle)$. If x satisfies the clause, you will always be orthogonal, and if x does not satisfy the clause, then there is some probability of getting the other outcome.

So, we showed how to measure the value of an assignment given a state of the form $|\phi_x\rangle$. We will now use a similar trick to the previous case to force a general product state to look like many copies of a state of this form. The difference is now we have $\sqrt{n}$ copies and can get better soundness. The way the test works is to pick a matching of the variables $\Pi = \{(a_1, b_2), \ldots, (a_{n/2}, b_{n/2})\}$, and define a measurement whose elements are $|a_1\rangle\langle a_1| + |b_1\rangle\langle b_1|, \ldots, |a_{n/2}\rangle\langle a_{n/2}|\,|b_{n/2}\rangle\langle b_{n/2}|$. (Should it be span?) By the birthday paradox, you have a constant chance of getting the same pair $|a\rangle, |b\rangle$ twice. Now, ideally, your system is supposed to be in the state

$$\left((-1)^{x_a}|a\rangle + (-1)^{x_b}|b\rangle\right)^{\otimes 2}.$$

To check this, measure whether the state is orthogonal to $\frac{|ab\rangle - |ba\rangle}{\sqrt{2}}$.