18.218 Topics in Combinatorics Spring 2021 – Lecture 16

Dor Minzer

In this lecture, we will discuss the Max-Cut problem in more detail. We will show the Goemans-Williamson algorithm, and show that assuming the Unique-Games Conjecture presented last time, this algorithm is tight.

1 The Goemans-Williamson algorithm

Recall that last time, we have seen a $\frac{1}{2}$-approximation algorithm for the Max-Cut problem. In 1995, Goemans and Williamson showed that (surprisingly), this simple algorithm is not optimal, and that there is a better approximation algorithm that achieves $\alpha_{GW} \approx 0.87856$ times the optimum in this problem. Their algorithm is very geometric in spirit, and is a prominent example of the use of semi-definite programming relaxations in order to solve optimization problems.

1.1 The integer programming relaxation

We first phrase the Max-Cut problem as an integer program. For each vertex $v \in V$ we create a variable x_v, whose value is supposed to be in $\{-1, 1\}$. The idea is that $x_v = 1$ will represent that v is on the left side, and $x_v = -1$ will represent that v is on the right side. Thus, if $(u, v) \in E$, then $x_u x_v = -1$ iff (u, v) crosses the cut, and otherwise $x_u x_v = 1$. Therefore, the following program solves Max-Cut

$$\begin{array}{lll} \max & \frac{1}{2} \sum\limits_{(u,v)\in E} 1 - x_u x_v & \\ \text{subject to} & x_v \in \{-1, 1\} & \forall v \in V. \end{array}$$

However, integer programming is NP-hard in general. Hence it seems that making this formulation doesn't advance us anywhere. That being said, this formulation does motivate us to look at higher dimensional, *semi-definite program* formulation of the problem (SDP).

1.2 The semi-definite programming relaxation

In the SDP formulation of the problem, instead of having a sign ± 1 for each x_u, we allow x_u to take any value in the unit ball in $\mathbb{R}^m$ (where m has to be chosen appropriately).

$$\begin{array}{lll} \max & \frac{1}{2} \sum\limits_{(u,v)\in E} 1 - \langle x_u, x_v \rangle & \\ \text{subject to} & \|x_v\|_2 = 1 & \forall v \in V. \end{array}$$

The good feature of this program, is that one can solve this optimization problem now. [1] The bad feature of this program is that a solution no longer gives us a cut; at least not in a straight-forward. But now we get to

[1] At least approximately, and thanks to the convexity that this has introduced to the problem. This is really an optimization problem over the cone of PSD matrices; the matrix here is the matrix of inner products $J = (\langle x_u, x_v \rangle)_{u,v \in V}$. We will not elaborate on this fact further in this course.

the amazing part: one can actually take a vector solution to the SDP program, and salvage from it a pretty good cut!

Here's the idea. Suppose the optimum size of the cut in our graph G is $\rho\,|E|$, where $\rho \in [1/2, 1]$, and let $\{x_v\}_{v\in V}$ be a solution to SDP program. First, it is clear that the optimum of the SDP program is at least $\rho\,|E|$ (why?), so in particular

$$\frac{1}{2}\sum_{(u,v)\in E} 1 - \langle x_u, x_v\rangle \geqslant \rho\,|E|\,.$$

We now generate a randomized cut from the vector solution. Take a random vector h from the unit ball in $\mathbb{R}^m$, and define

$$L = \{\, v \mid \langle x_v, h\rangle \leqslant 0\}\,; \qquad\qquad R = \{\, v \mid \langle x_v, h\rangle > 0\}\,.$$

Our goal is to analyze the expected number of edges that crosses the cur (L, R). Fix an edge $(u, v) \in E$; then the probability that (u, v) is cut is $\theta_{u,v}/\pi$, where $\theta_{u,v}$ is the angle between u and v. Thus, by linearity of expectation the expected size of the cut is

$$\sum_{(u,v)\in E}\frac{\theta_{u,v}}{\pi} = \sum_{(u,v)\in E}\frac{\mathsf{Arccos}(\langle x_u, x_v\rangle)}{\pi} \geqslant \sum_{(u,v)\in E}\alpha_{GW}\,(1-\langle x_u, x_v\rangle) \geqslant \alpha_{GW}\rho\,|E|\,.$$

Here, $\alpha_{GW} = \min_{z\in[-1,1]}\frac{\mathsf{Arccos}(z)/\pi}{(1-z)/2} \approx 0.878\ldots$; given this expectation calculation, standard tools allows one to design an approximation algorithm that achieves this approximation ratio.

Note that the calculation that we did here is eerily similar to the calculation we did to compute the stability of the majority function. This turns out not to be a coincidence, as we will see later on in this lecture.

1.3 The Goemans-Willaimson algorithm for almost bipartite graphs

With a more careful analysis, one can show that if the original size of the cut was very large, say $\rho = 1-\varepsilon$ for small ε, then the above analysis could be significantly improve.

Theorem 1.1. *Suppose $G = (V, E)$ has a cut of size $(1-\varepsilon)\,|E|$. Then the expected size of the cut in the Goemans-Williamson algorithm is at least $\left(1 - \frac{2}{\pi}\sqrt{\varepsilon} - O(\varepsilon^{1.5})\right)|E|$.*

2 A hardness result for Max-Cut

In this section, we prove the following result due to Khot, Kindler, O'Donnell and Mossel.

Theorem 2.1. *Assuming the Unique-Games Conjecture, for all $\rho \in (0,1)$ and $\varepsilon > 0$, given a graph $G = (V, E)$ it is NP-hard to distinguish between the following two cases:*

1. **YES case**: *G has a cut of fractional size at least $\frac{1}{2} + \frac{1}{2}\rho - \varepsilon$.*
2. **NO case**: *all cuts in G have fractional size at most $1 - \frac{1}{\pi}\mathsf{Arccos}(\rho) + \varepsilon$.*

In gap notations, gap-MaxCut[$\rho, 1 - \mathsf{Arccos}(\rho) + \varepsilon$] is NP-hard for all $\rho \in (0,1)$, $\varepsilon > 0$, assuming the Unique-Games Conjecture. Choosing $\rho = -z$ where achieves the minimum in the definition of α_{GW} (z turns out to be negative), this theorem implies the optimality of the Goemans-Williamson algorithm.

To prove this theorem we shall use gap preserving reductions. First, recall the statement of UGC:

Definition 2.2. *An instance of Unique-Games, denoted by Ψ, is composed of a bipartite, bi-regular graph $G = (V = L \cup R, E)$, a finite alphabet Σ, and a collection of constraints $\Phi = (\phi_e)_{e \in E})$ one for each edge. Each one the constraint ϕ_e is a 1-to-1 map, $\phi_e \colon \Sigma \to \Sigma$.*

For an edge e, the constraint ϕ_e defines a collection of tuples which are deemed as satisfactory assignments to the endpoints of the edge, which is $\{(\sigma, \phi_e(\sigma)) \mid \sigma \in \Sigma\}$.

Conjecture 2.3 (The Unique-Games Conjecture)**.** *For all $\eta > 0$, there exists $k \in \mathbb{N}$ such that given a Unique-Games instance Ψ, it is NP-hard to distinguish between:*

1. **YES case***:* $\mathsf{val}(\Psi) \geqslant 1 - \eta$*.*
2. **NO case***:* $\mathsf{val}(\Psi) \leqslant \eta$*.*

In other words, gap-UniqueGames$_k[1 - \varepsilon, \delta]$ is NP-hard.

We will show a polynomial time procedure $M \colon \Psi \to G$, that given an instance Ψ of Max-Cut, produces a graph G, such that:

1. If $\mathsf{val}(\Psi) \geqslant 1 - \eta$, then G has a cut of fractional size at least $\frac{1}{2} + \frac{1}{2}\rho - \varepsilon$.
2. If $\mathsf{val}(\Psi) \leqslant \eta$, then all cuts in G have fractional size at most $1 - \frac{1}{\pi}\mathsf{Arccos}(\rho) + \varepsilon$.

In particular, once we show this procedure, this proves Theorem 2.1 (why?). This is the type of reductions that most often appear in TCS.

2.1 Dictatorship vs no-influential-coordinates paradigm

A basic paradigm to prove hardness of approximation results proceeds by constructing instances of the problem we're interested in over the Boolean cube, wherein good solutions corresponds to dictatorship functions, whereas any function that only has small individual influences is automatically guaranteed to not be a good solution. In our case, we would like to design a (weighted) graph over $\{-1, 1\}^n$, such that

1. For any $i \in [n]$, the dictatorship cut, i.e. $L = \{x \mid x_i = 1\}$, $R = \{x \mid x_i = -1\}$, contains many edges.
2. If $f \colon \{-1, 1\}^n \to \{-1, 1\}$ is balanced, and has no influential coordinates, then the cut that it defines does not contain many edges.

So how would we design such graph in this case? Let $\rho > 0$ be a parameter, best thought of as close to 1, i.e. $\rho = 1 - \varepsilon$. We look at the graph corresponding to $-\rho$ correlated points, i.e. for each $x \in \{-1, 1\}^n$, the distribution over its neighbours is the distribution $T_{-\rho} x$.

1. For any $i \in [n]$, the dictatorship cut, i.e. $L = \{x \mid x_i = 1\}$, $R = \{x \mid x_i = -1\}$ contains edges of total weight $\frac{1}{2} + \frac{1}{2}\rho$ (why?).
2. If $f \colon \{-1, 1\}^n \to \{-1, 1\}$ is balanced odd function, and has no influential coordinates, then the size of the cut is

$$\Pr_{\substack{x \\ y \sim T_{-\rho} x}}[f(x) \neq f(y)] = \frac{1}{2}(1 - \mathsf{Stab}_{-\rho}(f)) = \frac{1}{2} + \frac{1}{2}\mathsf{Stab}_{\rho}(f) \leqslant \frac{1}{2} + \frac{1}{2}\left(1 - \frac{2}{\pi}\mathsf{Arccos}(\rho)\right) + o(1),$$

which is equal to $1 - \frac{1}{\pi}\mathsf{Arccos}(\rho) + o(1)$. Here, we used the Majority is Stablest theorem.

Thus, using the Majority is Stablest theorem we managed to construct a graph on the Boolean cube wherein dictators correspond to good cuts, and functions that have no influential coordinates correspond to bad cuts. In the rest of this lecture, we will see how to transfer this construction into a hardness result, assuming UGC.

2.2 A reduction from Unique-Games to Max-Cut

We are now ready to present the reduction. Let $\rho = 1 - \varepsilon$. Starting with a bi-partite UG instance $\Psi = (V \cup U, E, \Sigma, \Phi)$, we wish to construct a Max-Cut instance with the properties described above. The idea will be to introduce, for each vertex $v \in V$ a separate hybercube $\{-1,1\}^{\Sigma}$, and using a cut in that hypercube to encode the label that v is supposed to get in Ψ. More specifically, we will want to associate with each label σ of v which is supposed to have high value; this will be the dictatorship cut, i.e. the cut defined by $f_v(x) = x_\sigma$. Once we do that, we will be able to argue that if Ψ has a good assignment, then the graph we produce G will have a large cut corresponding to the dictatorship functions in each hypercube.

To ensure soundness, we must take care of two potential issues:

1. Penalizing cuts that are defined by functions that do not "resemble" any dictatorship. We have already dealt with this issue the last section, wherein we argued that in that case the cut size would be at most $1 - \frac{1}{\pi}\mathsf{Arccos}(\rho) + o(1)$ if f does not have any coordinate with significant low-degree influence.
2. Penalizing violating the constraints of Ψ. Namely, suppose we have two vertices $v \in V$, $u \in U$ that have an edge between them, and they have been assigned by dictatorship functions $f_v(x) = x_{\sigma_v}$, $f_u(x) = y_{\sigma_u}$, but σ_v, σ_u do not satisfy the constraint between v and u in Ψ. In that case, we would want to penalize this cut, as it does not correspond to a good assignment in Ψ. To deal with this issue, our edges will not really be inside the hypercube of each vertex v, but rather *across hypercubes*. For that, it is important to note that there is a natural bijection between the hypercube of v and the hypercube of u respecting the constraint between them, which is simply $x \to y$ where $y_i = x_{\phi_{u,v}(i)}$.

This almost finishes the informal overview of the reduction, except that if we were to execute the plan as is, we would get a bipartite graph (the sides being the hypercubes of V and the hypercubes of U), and to remedy that we only leave one of these sides alive, and take two steps in the graph of Ψ instead of one.

We now proceed to the formal construction of the reduction. Given $\Psi = (V \cup U, E, \Sigma, \Phi)$, we construct a weighted max-cut instance $G = (V', E', w)$ as follows.

- The vertices: For each $v \in V$ we construct a cube over Σ, $\{v\} \times \{-1,1\}^{\Sigma}$, which we refer to as the long-code of v. A ± 1 assignment to these vertices should be thought as a potential encoding one of the labels in Σ for v.
- The edges are weighted according to the following randomized process. Sample $u \in U$ and $v, v' \in V$ two neighbours of u independently. Let x be a uniformly chosen vector from $\{-1,1\}^{\Sigma}$, and sample $y \sim T_{-\rho} x$. Consider the points

  $$z = \phi_{u,v}(x), \qquad z' = \phi_{u,v'}(y), \qquad \text{where } \phi_{u,v}(y)_\sigma = y_{\phi_{(u,v)}(\sigma)} \ \forall \sigma \in \Sigma.$$

  The edge output by the process is (z, z').

We prove the following lemma, encapsulating the analysis of the reduction.

Lemma 2.4. *For all $\rho \in (0,1)$, $\delta > 0$ there is $\eta > 0$ such that:*

1. *Completeness: if Ψ is at least $1-\eta$ satisfiable, then there is a cut in G of weight at least $\frac{1}{2}(1+\rho) - \delta$.*
2. *Soundness: if Ψ is at most η satisfiable, then G has no cut whose weight exceeds $1 - \frac{1}{\pi}\mathsf{Arccos}(\rho) + \delta$.*

2.3 Analysis of the reduction

We now analyze the construction. First, we show the completeness of the construction, asserting that if Ψ is highly satisfiable, then there exists a large cut on the graph we have constructed.

Completeness

Suppose there is a coloring $A\colon V \cup U \to \Sigma$ satisfying at least $1 - \eta$ fraction of the edges. We assign ± 1 values to the cube of v according to the dictatorship assignment of $A(v)$. Namely, we define the cut in the graph G by

$$f(v, x) = x_{A(v)} \text{ for } (v, x) \in V \times \{-1, 1\}^{\Sigma}.$$

We analyze the weight of the cut defined by f. Looking at the process describing the weights of the edges in G', Since the graph of Ψ is regular, the marginal distribution of each one of the edges $(u, v), (u, v')$ is uniform; therefore the probability one of them is not satisfied by A is at most 2η, so with probability at least $1 - 2\eta$ both edges are satisfied.

Sample x, y as in the process, and look at $\phi_{(u,v)}(x), \phi_{(u,v')}(y)$. Note that $y_{A(u)} \neq x_{A(u)}$ with probability $\frac{1}{2} + \frac{1}{2}\rho$, and if that happens, since both edges (u, v) and (u, v') are satisfied, we get that

$$f(v, z) = z_{A(v)} = z_{\phi_{u,v}(A(u))} = x_{A(u)} \neq y_{A(u)} = {z'}_{\phi_{u,v}(A(u))} = f(v, z').$$

We conclude that the weight of edges crossing the cut is at least $\frac{1}{2} + \frac{1}{2}\rho - 2\eta$.

Soundness

In this part, we show that if the UG instance Ψ had no good satisfying assignments then the graph G does not have a large cut. This is usually done (and so will be our case) in a counter-positive way. Assuming we have a large cut in the graph, we will construct a good assignment for Ψ.

Let $f\colon V \times \{-1, 1\}^{\Sigma} \to \{-1, 1\}$ be a function corresponding to a large cut, that is a cut of size at least $\frac{1}{\pi}\mathsf{Arccos}(\rho) + \delta$. The fractional size of the cut is exactly

$$\Pr_{\substack{u,v,v' \\ x,y,z,z'}} \left[f(v', z) \neq f(v, z)\right].$$

Let ν be a vector from $\{-1, 1\}^{\sigma}$ such each coordinate is -1 with probability $\frac{1}{2}(1 - \rho)$. Then the previous probability is the same as

$$\Pr_{\substack{u,v,v' \\ x,\nu}} \left[f(v, \phi_{(u,w)}x) \neq f(v', \nu \cdot \phi_{(u,w')}x)\right].$$

Define for $u \in U$, $v \in V$

$$g_u(x) = \mathop{\mathbb{E}}_{v:(u,v)\in E} \left[f(v, \phi_{(u,v)}x)\right], \qquad g_v(x) = f(v, x).$$

Intuitively, u asks his neighbours what side it should be on, and takes the average of the suggestions. Then

$$\begin{aligned}
\Pr_{\substack{u,v,v'\\x,\nu}}\left[f(v,\phi_{(u,w)}x) \neq f(v',\nu\cdot\phi_{(u,w')}x)\right] &= \frac{1}{2}(1 - \mathop{\mathbb{E}}_{\substack{u,v,v'\\x,\nu}}\left[f(v,\phi_{(u,w)}x)f(v',\nu\cdot\phi_{(u,w')}x)\right]) \\
&= \frac{1}{2}\left(1 - \mathop{\mathbb{E}}_{\substack{u\\x,\nu}}\left[\mathop{\mathbb{E}}_{v}\left[f(v,\phi_{(u,w)}x)\right]\mathop{\mathbb{E}}_{v'}\left[f(v',\phi_{(u,w')}(\nu\cdot x))\right]\right]\right) \\
&= \frac{1}{2}(1 - \mathop{\mathbb{E}}_{\substack{u\\x,\nu}}\left[g_u(x)g_u(\nu\cdot x)\right]) \\
&= \frac{1}{2}(1 - \mathop{\mathbb{E}}_{u}[\mathsf{Stab}_{-\rho}[g_u]]).
\end{aligned}$$

We conclude that since the fractional size of the cut is at least $1 - \frac{1}{\pi}\mathsf{Arccos}(\rho) + \delta$, it holds that

$$\mathop{\mathbb{E}}_{u}[\mathsf{Stab}_{-\rho}[g_u]] < \frac{2}{\pi}\mathsf{Arccos}(\rho) - 1 - 2\delta.$$

Therefore for at least δ fractional of the u's, $\mathsf{Stab}_{-\rho}[g_u] < 1 - \frac{2}{\pi}\mathsf{Arccos}(\rho) - \delta$. We need a version of the Majority is Stablest theorem for negative correlation parameters.

Theorem 2.5. *For all $\rho \in (0,1)$, $\delta > 0$ there exist $d \in \mathbb{N}$, $\tau > 0$ such that if $f\colon \{-1,1\}^n \to [-1,1]$ is a function for which $\max_i I_i^{\leqslant d}[f] \leqslant \tau$, then*

$$\mathsf{Stab}_{-\rho}(f) \geqslant \frac{2}{\pi}\mathsf{Arccos}(\rho) - 1 - \delta.$$

Proof. Let f_{odd} be the odd part of f. Then f_{odd} is balanced and we have by the Fourier expression for stability that $\mathsf{Stab}_{-\rho}(f) \geqslant \mathsf{Stab}_{-\rho}(f_{\mathsf{odd}}) = -\mathsf{Stab}_{\rho}(f_{\mathsf{odd}})$. By the Majority is stablest theorem we get that for appropriate choice of d, τ, $\mathsf{Stab}_{\rho}(f_{\mathsf{odd}}) \leqslant 1 - \frac{2}{\pi}\mathsf{Arccos}(\rho) + \delta$. □

We fix d, τ corresponding to ρ, δ as in Theorem 2.5 and apply it to get that there is i such that $I_i^{\leq d}[g_u] \geqslant \delta$. We call such u good. Define

$$\mathsf{List}_{\xi}(v) = \left\{ i \mid I_i^{\leqslant d}[g_v] \geqslant \xi \right\}.$$

Since the sum of the d degree influence is at most d, $|\mathsf{List}S(v)| \leqslant d/\xi$; the important point is that this quantity only depends on ρ, ε (and not on $|\Sigma|$). We finish by showing that if u is good and $i \in \mathsf{List}_{\delta}(u)$, then a non-negligible fraction of his neighbours v have $\phi_{(u,v)}(i) \in \mathsf{List}_{\delta/2}(w)$

$$\begin{aligned}
I_i^{\leqslant d}[g_u] &= \sum_{S: i\in S, |S|\leqslant d} \widehat{g_u}^2(S) \\
&= \sum_{S: i\in S, |S|\leqslant d} \mathop{\mathbb{E}}_{v:(u,v)\in E}\left[\widehat{g_v}(\phi_{(u,v)}S)\right]^2,
\end{aligned}$$

where we used the definition of low degree influence and the following simple lemma

Lemma 2.6. $\widehat{g_u}(S) = \mathbb{E}_{w:(u,w)\in E}\left[\widehat{g_w}(\phi_{(u,w)}S)\right]$.

We continue by using Jensen

$$
\begin{aligned}
I_i^{\leq d}[g_u] \leq \sum_{S: i\in S, |S|\leq d} \mathop{\mathbb{E}}_{v:(u,v)\in E}\left[\widehat{g_v}(\phi_{(u,v)}S)^2\right] &= \mathop{\mathbb{E}}_{v:(u,v)\in E}\left[\sum_{S:i\in S,|S|\leq d} \widehat{g_v}(\phi_{(u,v)}S)^2\right] \\
&= \mathop{\mathbb{E}}_{v:(u,v)\in E}\left[\sum_{T:\phi_{(u,v)}(i)\in T,|T|\leq d} \widehat{g_v}(T)^2\right] \\
&= \mathop{\mathbb{E}}_{v:(u,v)\in E}\left[I^{\leq d}_{\phi_{(u,v)}(i)}[g_v]\right].
\end{aligned}
$$

From $I_i^{\leq d}[g_u] \geqslant \delta$ and the above it follows that for at least $\delta/2$ fraction of the v neighbours of u it holds that $I^{\leq d}_{\phi_{(u,v)}(i)}[g_v] \geq \delta/2$, or in other words $\phi_{(u,v)}(i) \in \mathsf{List}(v)$.

Randomized assignment to the Unique-Games instance

Now we finish the proof. For each good $u \in U$ assign a label $i \in \mathsf{List}_{\delta}(u)$ randomly, and for each $v \in V$ assign a label from $\mathsf{List}_{\delta/2}(v)$ randomly. We now lower the probability a randomly chosen edge from Ψ is satisfied.

Choose (u, v) randomly. With probability at least δ, the vertex u is good, and conditioned on that with probability at least $\delta/2$, $\mathsf{List}_{\delta/2}(v)$ is not empty. We know that it has at most $2d/\delta$ elements and at least one of them matches label we assigned to u, and hence the probability the edge is satisfied conditioned on the previous events happening it at least $\frac{1}{2d/\delta} = \delta/2d$. We conclude that the probability that a random edge is satisfied is at least

$$
\delta \cdot \frac{\delta}{2} \cdot \frac{\delta}{2k} > \eta,
$$

in contradiction to the fact we started with a NO case instance of Unique-Games.

18.218 Topics in Combinatorics: Analysis of Boolean Functions
Spring 2021