Approximation Algorithms and Hardness of Approximation May 14, 2013

Lecture 22

Lecturer: Alantha Newman *Scribes: Christos Kalaitzis*

1 The Unique Games Conjecture

The topic of our next lectures will be the Unique Games Conjecture. This conjecture implies that certain central problems are NP-hard to approximate within bounds close to the best known factors for these problems. For example, it would imply that the current best known factors for the Max-Cut and Vertex Cover problems are optimal. Thus, although the question is not yet resolved, and there is no clear consensus among the research community regarding its veracity, it is still quite instructive to examine it. We will begin by describing the core problem that is used in the reductions:

Definition 1 *An instance $L = (G((V,W),E),[M],\{\pi_{vu}\})$ of the* Unique Label Cover problem (ULC) *is a bipartite graph G with vertex set (V,W) and edges E. The set $[M]$ is a set of M labels. For each edge $(v,u) \in V \times W$, there is a permutation $\{\pi_{vu} : [M] \to [M]\}$. Our objective is to find an assignment $\ell : V \cup W \to [M]$ of labels such that we maximize the number of edges which are satisfied; an edge (u,v) is satisfied if $\ell(u) = \pi_{vu}(\ell(v))$.*

Specifically, the decision problem we will study is the following:

Definition 2 (GAP-ULC) *For some $\delta > 0$ and a given ULC instance L, ULC(δ) is the decision problem of distinguishing between the following two cases:*

(i) There is a labeling ℓ under which at least $(1-\delta)|E|$ edges are satisfied ($OPT(L) \geq (1-\delta)|E|$).

(ii) For any labeling l, at most $\delta|E|$ edges are satisfied ($OPT(L) < \delta|E|$).

The Unique Games Conjecture is essentially that this decision problem is NP-hard.

Conjecture 3 (Unique Games Conjecture (UGC)) *For any $\delta > 0$, there is an M such that it is NP-hard to decide ULC(δ) on instances with label set of size M.*

From an algorithmic perspective, it is easier to think of a restricted problem that is simpler to describe than Unique Label Cover in its most general form. Given a set of m linear equations of the form $x_i - x_j \equiv c_{ij} \pmod q$, the problem of Max-2-Lin-q is defined as finding the assignment to the variables $x_i \in [0, q-1]$ that maximizes the number of satisfied constraints. One of the implications of the UGC is the following theorem of [**?**]:

Theorem 4 *The Unique Games Conjecture implies that for every $\delta > 0$ and for all sufficiently large q, it is NP-hard to distinguish between the following instances of Max-2-Lin-q:*

(i) There is an assignment to the variables that satisfies at least a $1-\delta$ fraction of the constraints.

(ii) Any assignment of variables satisfies at most a $\frac{1}{q^{\frac{\delta}{2-\delta}}}$ fraction of the constraints.

It is easy to achieve a $\frac{1}{q}$-approximation algorithm for this problem. Assigning each variable a value uniformly at random from $[q]$ will satisfy each constraint with probability $\frac{1}{q}$. However, Theorem **??** shows that we need to beat this bound. We we note that Max-Cut is a special case of Max-2-Lin-q where $q = 2$ and for every edge (i,j) in the graph we have the constraint $x_i - x_j \equiv 1 \pmod 2$. In this special case, the variables take values in $\{0,1\}$. Thus, the constraint is satisfied (i.e. the edge is cut) iff the two variables corresponding to an edge have different values (i.e. the corresponding vertices are assigned to different sides of the cut).

2 Unique Games Hardness of Max-Cut

We will now exhibit the connection between the Unique Games Conjecture and the inapproximability of Max-Cut. Specifically, we will show that, assuming the UGC, it is NP-hard to approximate Max-Cut to within a factor larger than α_{GW}, the factor given by the Goemans-Williamson algorithm. Recall that α_{GW} had the following value:

$$\alpha_{\text{GW}} = \min_{0 \le \theta \le \pi} \frac{2}{\pi} \frac{\theta}{1 - \cos\theta} > 0.878.$$

In order to do this, we will prove the following theorem.

Theorem 5 *For every $\rho \in (-1, 0)$ and any $\epsilon > 0$, there is some $\delta > 0$ such that there is a PCP for ULC(δ) in which the verifier reads two bits of the proof and accepts if they are unequal and which the following holds:*

- *If $L \in$ULC(δ), the verifier accepts with probability at least $c \geq \frac{1-\rho}{2} - \epsilon$.*
- *If $L \notin$ULC(δ), the verifier accepts with probability at most $s \leq \frac{1}{\pi} \arccos\rho + \epsilon$.*

Assuming the UGC, Theorem **??** implies that it is NP-hard to approximate Max-Cut within a factor better than α_{GW}. Why is this the case? For an instance L of ULC(δ), consider the PCP given by the theorem. Let G be a graph that has the bits of the proof as vertices. Put an edge between two vertices if there is a non-zero probability of begin sampled by the verifier. All edges have unit weight. A proof, which is an assignment of $\{-1, 1\}$ to the bits, corresponds to a cut in G, and

$$\Pr[\text{Verifier accepts}] \quad = \quad \frac{\text{Number of edges crossing cut}}{\text{Total number of edges}}.$$

Assuming UGC, this is a gap-introducing reduction for Max-Cut with a gap of

$$\frac{\frac{\arccos\rho}{\pi} + \epsilon}{\frac{1-\rho}{2} - \epsilon} > \alpha_{\text{GW}}.$$

This result is very surprising, since it tells us that the best known algorithm for Max-Cut is the best we can achieve if we assume the UGC. However, later on Prasad Raghavendra showed a connection between the UGC and the integrality gap of a certain form of SDP relaxation. Since the SDP relaxation used in the Goemans-Williamson algorithm has an integrality gap of α_{GW}, this result provides an alternate proof of the fact that Max-Cut is hard to approximate to within better than α_{GW} assuming the UGC.

2.1 Designing a 2-Query Long Code Test

In order to construct the actual PCP verifier, we need to design a test that distinguishes between functions that are "dictators" and functions that are far from being dictators. We say a function is a dictator if it corresponds to an actual labeling. In this case, we want to design a 2-query test such that a long code encoding of such a function will pass with probability lower bounded by the completeness (in the satisfiable case (i)). In the case that there does not exist a good assignment (case (ii)), it is possible that a function that is far from a dictator could still pass the test. Thus, we need to use the test to determine if a function is "far" from a dictator function.

Let us consider the following candidate test that checks if a function $f : \{-1, 1\}^n \to \{-1, 1\}$ is a long code.

1. Pick $x \in \{-1, 1\}^m$ uniformly at random.
2. Set $y = -x$.

3. Accept if $f(x) \neq f(y)$.

If f is a dictator, then this test will always pass. We therefore have completeness 1. However, the soundness is problematic. There are functions that are clearly not dictators that will also pass this test: (Recall that $\chi_S(x) = \prod_{i \in S} x_i$.)

- If $f(x) = \chi_S(x)$, where $|S|$ is odd.
- If f is the majority function: $f(x) = \operatorname{sgn} \sum_{i=1}^{m} x_i$ and m is odd.

A modified way to perform this test is the following procedure, which we will call the **Long Code Test**:

1. Pick $x \in \{-1, 1\}^m$ uniformly at random.
2. Pick μ as follows: for each $i \in [m]$, independently set μ_i to be 1 with probability $\frac{1+\rho}{2}$, and -1 with probability $\frac{1-\rho}{2}$.
3. Set $y \in \{-1, 1\}^m$ such that $y_i = \mu_i x_i$.
4. Accept iff $f(x) \neq f(y)$.

2.2 Completeness and Soundness of the Long Code Test

Let us now analyze the completess and soundness of this test.

Completeness: If $f(x) = x_i$ for some i (i.e. f is a dictator function), then the Long Code test accepts if $f(x) \neq f(y)$, which occurs iff $x_i \neq \mu_i x_i$, or equivalently if $\mu_i = -1$. By the construction of μ, this happens with probability $\frac{1-\rho}{2}$.

Soundness: If f is not a dictator function, then the probability that it passes the test is:

$$\begin{aligned}
\Pr_{x,\mu}[f(x) \neq f(y)] &= \Pr_{x,\mu}[f(x) \neq f(\mu x)] \\
&= \mathop{\mathbb{E}}_{x,\mu}[\frac{1 - f(x)f(\mu x)}{2}] \\
&= \frac{1}{2} - \frac{1}{2} \mathop{\mathbb{E}}_{x,\mu}[f(x)f(\mu x)].
\end{aligned}$$

We want to show that when f is far from a dictator, this quantity is small, so that the test will pass with low probability. The quantity is related to the notion of *stability* of the function, which we will show (by the end of the section) is small for functions that are far from dictators.

2.2.1 Stability of Non-Dictator Functions

We define the stability of a function f as follows:

Definition 6 *The* stability *of a function f, denoted by $Stab_\rho(f)$, is $\mathbb{E}_{x,\mu}[f(x)f(\mu x)]$.*

We will also use the following fact relating the stability of a function with its Fourier coefficients.

Fact 7 *For a function f, $Stab_\rho(f) = \sum_{S \subseteq [m]} \hat{f}^2(S)\rho^{|S|}$.*

Notice that the stability of a dictator function is exactly ρ (since its output only depends on one input), and, as previously noted, a dictator function therefore passes the test with probability $\frac{1-\rho}{2}$. Let us now consider some other functions that are not dictators or are "far" from dictators and see what stability these functions have. Note that $\rho \in (-1, 0)$ and therefore $\mu_i = -1$ with probability at least half. Thus, when a function is far from a dictator, ideally it is the case that the stability is close to zero. Indeed, for a linear function χ_S, its stability goes to zero as $|S|$ goes to infinity, since a lot of weight is put on the Fourier coefficients corresponding to sets with large size. Its probability of passing our test therefore tends to $\frac{1}{2}$, as desired for a function far from a dictator.

Another important non-dictator function to consider (and, as we will see, one of the most important ones) is the majority function. For this function, the probability of acceptance approaches $\frac{\arccos \rho}{\pi}$, as m approaches infinity. Why should this be the case? Recall that in the Goemans-Williamson rounding algorithm for Max-Cut, we saw that the probability of cutting an edge was proportional to the angle between the two unit vectors corresponding to the vertices of that edge. In that algorithm, we had two given vectors, and we checked to see if they fell on either side of randomly chosen hyperplane. In our Long Code test, the expected dot product between between vectors x and y is ρ. We can also think of this test as checking if the two vectors x and y fall on either side of the hyperplane denoted by the vector of all 1's. In this scenario, however, the hyperplane is fixed and the vector y is chosen at random but with the angle between x and y set to be $\arccos \rho$ in expectation.

Our goal is now to show that if a function f passes the test with probability at least $\frac{\arccos \rho}{\pi} + \epsilon$, for some $\epsilon > 0$, then f must be similar to a dictator function. That is, one of its variables has a large influence on the outcome of f.

2.2.2 Influential Variables

For a function f, the influence of a variable x_i is defined to be:

$$Inf_i(f) = \Pr_{x \in \{-1,1\}^m}[f(x) \neq f(x_1, x_2, \cdots - x_i, x_{i+1}, \ldots, x_n)].$$

In other words, if we negate the i^{th} coordinate of input x, what is the probability that the value of the function changes? The influence of a variable also has the following definition.

Fact 8 *For a function f, $Inf_i(f) = \sum_{S: i \in S} \hat{f}(S)^2$.*

Let us look at the influence of variables in some simple functions:

- $Inf_i(x_j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$
- $Inf_i(\chi_S(x))= \begin{cases} 0 & \text{if } i \in S \\ 1 & \text{if } i \notin S \end{cases}$
- Inf_i(parity)=1
- Inf_i(majority)= $\Theta(\frac{1}{\sqrt{m}})$.

We will also need the notion of low-degree influence, which is denoted by $Inf_i^{\leq d}(f)$.

Definition 9 $Inf_i^{\leq d}(f) = \sum_{S: i \in S, |S| \leq d} \hat{f}(S)^2$.

Our goal now becomes to show that a function has stability upper bounded by that of the majority, or the function has high low-defree influence, and can therefore be decoded, i.e. it is close to a dictator. We state the following theorem that, roughly speaking, says that the majority function is most stable among all functions where all the coordinates have low influence. Note that stability ranges from $(-1, 0)$, so what we actually mean when we say that "majority is stablest" is that it has the highest probability of passing our Long Code test (among all functions where all the coordinates have low influence).

Theorem 10 (Generalized Majority is Stablest) *For any $-1 < \rho < 0$ and $\epsilon > 0$, there exists $\tau > 0$ and $d \in \mathbb{N}$, such that if some function $f : \{-1,1\}^m \to \{-1,1\}$ has $Inf_i^{\leq d} < \tau$ for all $i \in [m]$, then*

$$\frac{1}{2} - \frac{1}{2}Stab_\rho(f) < \frac{\arccos \rho}{\pi} + \epsilon.$$

3 Reduction from Unique Label Cover to Max-Cut

We are now ready for the reduction from ULC. Given an instance of GAP-ULC, $L = (G((V,W),E),[M],\{\pi_{vu}\})$, we describe the test the PCP verifier will perform. Here is our first attempt:

- Pick an edge (v, w) in G uniformly at random.
  - Let f_v, f_w be the supposed long codes of the vertices labels.
- Let π denote the permutation corresponding to edge (v, w).
- Pick $x \in \{-1,1\}^m$ uniformly at random, and pick $\mu \in \{-1,1\}^m$ according to $\frac{1-\rho}{2}$-biased distribution.
- Accept iff $f_v(x) \neq f_w(y \circ \pi)$, where $y \in \{-1,1\}^m$ and $y_i = \mu_i x_i$.
  - $y \circ \pi$ is the vector $(y_{\pi(1)}, y_{\pi(2)}, \ldots, y_{\pi(m)})$.

This test is a reasonable attempt, but it is actually flawed. This is because setting f_v to return 1 and f_w to return -1 will always cause this test to accept, since G is bipartite. This just demonstrates the care that we must use when using our 2-query test in a reduction. Finally, let us describe a valid test that the PCP verifier can perform:

- Pick a vertex $v \in V$ uniformly at random.
- Pick two random edges (v, w) and (v, w').
- Let π and π' be the permutations corresponding to edges (v, w) and (v, w') respectively.
- Pick $x \in \{-1,1\}^m$ uniformly at random and $\mu \in \{-1,1\}^m$ according to a $\frac{1-\rho}{2}$-biased distribution and set $y = \mu x$.
- Accept iff $f(x \circ \pi) \neq f(y \circ \pi')$.

3.1 Completeness

We will now analyze this verifier. Suppose that for the given Unique Label Cover instance L, there is a labeling $\sigma : (V \cup W) \to [M]$ that satisfies at least a $1-\delta$-fraction of the constraints. Suppose that f_w is the actual long code that we expect for $\sigma(w)$, i.e. $\sigma(w)$ is the $\sigma(w)^{th}$ dictator function.

Since both edges were chosen uniformly at random, the probability that both edges (v, w) and (v, w') are satisfied by the labeling σ is at least $1 - 2\delta$. In this case, what is the probability that the test succeeds? Note that $f_w(x \circ \pi) = (x \circ \pi)_{\sigma(w)} = x_{\pi(\sigma(w))}$. Therefore, in this case, since σ satisfies (v, w), we have:

$$\pi(\sigma(w)) = \sigma(v).$$

We also have:

$$f_{w'}(y \circ \pi') = (y \circ \pi')_{\sigma(w')} = y_{\pi'(\sigma(w'))}.$$

Since σ satisfies (v, w'), we know that

$$\pi'(\sigma(w')) = \sigma(v).$$

Recall that the test succeeds iff $x_{\sigma(v)} \neq y_{\sigma(v)} = (\mu x)_{\sigma(v)}$. Note that this occurs exactly when $\mu_{\sigma(v)} = -1$, which happens with probability $\frac{1-\rho}{2}$. Thus, the total probability of acceptance is at least $(1-2\delta)(\frac{1-\rho}{2})$, which is greater than $\frac{1-\rho}{2} - \epsilon$, if ϵ is chosen to be greater than 2δ.

3.2 Soundness

Now we want to prove that if every labeling satisfies at most a δ-fraction of the constraints, then the test is accepted with probability at most $\frac{\arccos\rho}{\pi} + \epsilon$. In order to do this, we will prove the contrapositive: If a test accepts with probability at least $\frac{\arccos\rho}{\pi} + \epsilon$, then there is a labeling that satisfies at least $\delta|E|$ constraints. By Markov's inequality, we know that there is at least an $\frac{\epsilon}{2}$-fraction of the vertices (let us call them "good") such that, if one of them is picked, then the test passes with probability at least $\frac{\arccos\rho}{\pi} + \frac{\epsilon}{2}$. Recall that the probability the test passes is:

$$\frac{1}{2} - \frac{1}{2} \mathop{\mathbb{E}}_{x,\mu}[\mathop{\mathbb{E}}_{w,w'}[f_w(x \circ \pi) \cdot f_{w'}(\mu x \circ \pi')]].$$

Let $g_v(x) = \mathop{\mathbb{E}}_{w \sim v}[f_w(x \circ \pi_{vw})]$. Then, conditioned on the selection of v, the probability the test accepts is:

$$\frac{1}{2} - \frac{1}{2} \mathop{\mathbb{E}}_{x,\mu}[g_v(x) g_v(\mu x)] \quad = \quad \frac{1}{2} - \frac{1}{2}\mathrm{Stab}_\rho(g_v). \tag{1}$$

If v is a good vertex, then the above expression (**??**) is at least:

$$\frac{\arccos\rho}{\pi} + \frac{\epsilon}{2}.$$

Applying the "Majority is Stablest" Theorem (Theorem **??**), we see that g_v must have at least one coordinate with large d-degree influence. In other words, there is some j such that:

$$\mathrm{Inf}_j^{\leq d}(g_v) \quad = \quad \sum_{S: j \in S, |S| \leq d} \hat{g}_v(S)^2 > \tau.$$

We will assign to vertex v the label j. Now we will label the neighbors of v. The key idea is a good fraction of these neighbors, it is the case that for a neighbor w the influence of the function f_w has a high influence in the coordinate π_{vw}^{-1}. If we choose labels for w from among these high-influence coordinates, we will have a good chance of satisfying the constraint on the edge (v, w).

We can expand the Fourier coefficients of the function g_v (see [**?**] for details) to obtain:

$$\tau \quad \leq \quad \mathop{\mathbb{E}}_{w}[\mathrm{Inf}_{\pi^{-1}(j)}^{\leq d}(f_w)].$$

This implies that at least a $\frac{\tau}{2}$-fraction of v's neighbors w have:

$$\mathrm{Inf}_{\pi_{vw}^{-1}(j)}^{\leq d}(f) \geq \frac{\tau}{2}.$$

Let $\mathrm{Cand}(w) = \{k \in [M] : \mathrm{Inf}_k^{\leq d}(f_w) \geq \frac{\tau}{2}\}$ be the set of candidate labels for vertex w. We will label the vertex w with a label chosen uniformly at random from the set $\mathrm{Cand}(w)$. Since the total d-degree influence cannot be more than d, if follows that $|\mathrm{Cand}(w)| \leq \frac{2d}{\tau}$. Therefore, the probability of satisfying the constraint on an edge (v, w) is at least:

$$\frac{\epsilon}{2}\frac{\tau}{2}\frac{1}{|\mathrm{Cand}(w)|} \;\; \geq \;\; \frac{\epsilon}{2}\frac{\tau}{2}\frac{\tau}{2d} \;\; \geq \;\; \delta.$$

Thus, the expected number of edges whose constraints are satisfied is at least $\delta|E|$, where δ is set to be a function of ϵ and τ.

References

[GO05] Venkatesan Guruswami and Ryan O'Donnell. Lecture Notes for CSE 533: The PCP Theorem and Hardness of Approximation. 2005.

[HCA+10] Prahladh Harsha, Moses Charikar, Matthew Andrews, Sanjeev Arora, Subhash Khot, Dana Moshkovitz, Lisa Zhang, Ashkan Aazami, Dev Desai, Igor Gorodezky, Geetha Jagannathan, Alexander S. Kulikov, Darakhshan J. Mir, Alantha Newman, Aleksandar Nikolov, David Pritchard, and Gwen Spencer. Limits of Approximation Algorithms: PCPs and Unique Games (DIMACS Tutorial Lecture Notes). *CoRR*, abs/1002.3864, 2010.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for Max-Cut and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.

This lecture was mainly based on the following sources: [**?**, **?**] and Chapter 9 from [**?**].