18.408 Topics in Theoretical Computer Science Fall 2022
Lecture 13

Dor Minzer

In this lecture, we restate the basic form of the PCP theorem using the Label-cover problem. We show how to use it to prove a weak hardness for approximation result for the problem of finding the largest clique in a graph, and then amplify this hardness. We also discuss of improved forms of the PCP theorem that are most relevant towards applications in hardness of approximation.

1 The Label Cover Problem

We have seen most of the proof of the basic PCP theorem, asserting that for some absolute constant $\varepsilon > 0$, the problem gap-CSG$[1, 1-\varepsilon]$ is NP-hard on instances with $O(1)$ queries and alphabet size $O(1)$. In the previous lecture, we have also seen a transformation that reduces an instance of CSG with constant number queries to a different instance of CSG with 2 queries while preserving perfect completeness and keeping the soundness bounded away from 1. This last transformation gets us a version of the PCP theorem for special type of constraint satisfaction called (projection) label cover problems, defined as follows.

Definition 1.1. *An instance of Label-cover Ψ consists of a bi-partite graph $G = (L \cup R, E)$, two alphabets Σ_L, Σ_R and a projection constraint for each edge, $\Phi = \{\Phi_e\}_{e \in E}$. By projection constraint, we mean that for each $e \in E$, there is a map $\phi_e \colon \Sigma_L \to \Sigma_R$ such that*

$$C_e = \{ (\sigma, \phi_e(\sigma)) \mid \sigma \in \Sigma_L \}.$$

In words, a label cover instance Ψ is a constraint satisfaction graph wherein the underlying constraint graph is bipartite, and each constraint is of projection type. That is, for each edge $e = (u, v) \in E$, the label of u determines the label that v should get for the constraint on e to be satisfied. The value of a label-cover instance Ψ, denoted by $\mathsf{val}(\Psi)$, is the maximum fraction of constraints that can be satisfied in it.

1.1 The Basic PCP Theorem in the Language of Label Cover

We can formulate the basic PCP theorem we proved so far using the Label-cover problem, as follows:

Theorem 1.2. *There are absolute constant $\varepsilon > 0$ and $k \in \mathbb{N}$ such that the problem gap-Label-Cover$[1, 1-\varepsilon]$ is NP-hard on instances with alphabet size at most k.*

Theorem 1.2 is the analog of the Cook-Levin theorem for approximation problems, and as such almost all hardness of approximation results use it as a starting point. To motivate this discussion, note that one may associate with the Label-cover problem an approximation problem, in which the input is a Label-cover instance Ψ, and the goal is to approximate $\mathsf{val}(\Psi)$. In this language, Theorem 1.2 implies that there are $\varepsilon > 0$ and $k \in \mathbb{N}$ such that given an instance Ψ of Label-Cover, is NP-hard to approximate $\mathsf{val}(\Psi)$ within factor $1 - \varepsilon$. Recall that for a maximization problem A (such as label cover), we say an algorithm Alg is an α-approximation, for $0 < \alpha \leqslant 1$, if on an input Ψ of A it outputs a number $\mathsf{Alg}(\Psi)$ such that $\alpha \mathsf{Opt}(\Psi) \leqslant \mathsf{Alg}(\Psi) \leqslant \mathsf{Opt}(\Psi)$.

Corollary 1.3. *There exists $\varepsilon > 0$ such that given a label-cover instance Ψ with constant size alphabet, it is NP-hard to approximate $\mathsf{val}(\Psi)$ within factor $1-\varepsilon$.*

Proof. Assume there is an algorithm Alg that approximates $\mathsf{val}(\Psi)$ in polynomial time within factor $\frac{1}{1-\varepsilon}$, i.e. it outputs a number $\mathsf{Alg}(\Psi)$ satisfying that $(1-\varepsilon)\mathsf{val}(\Psi) \leqslant \mathsf{Alg}(\Psi) \leqslant \mathsf{val}(\Psi)$. We use it to solve gap-Label-Cover$[1, 1-\varepsilon]$, which finishes the proof.

Indeed, given an instance Ψ of label cover, we run $\mathsf{Alg}(\Psi)$ and get a number s; we accept if $s > 1-\varepsilon$ and otherwise reject. Note that if $\mathsf{val}(\Psi) = 1$, then by the guarantee of the algorithm $\mathsf{Alg}(\Psi) \geqslant 1-\varepsilon$, hence we accept. If $\mathsf{val}(\Psi) < 1-\varepsilon$, then $\mathsf{Alg}(\Psi) < 1-\varepsilon$ hence we reject. Thus the described algorithm runs in polynomial time and solves gap-Label-Cover$[1, 1-\varepsilon]$. □

Thus, at least morally speaking one may expect one to get more hardness of approximation results from Theorem 1.2. Indeed, shortly after the proof of Theorem 1.2 (and actually even during earlier stages of it), researchers have been exploring connections between it and approximation problems, and today we will begin seeing some of this wonderful theory.

2 Hardness of Approximating the Maximum Clique

Our first example is the maximum clique problem. Recall that given a graph $H = (V, E)$, a clique on H is a subset of vertices $S \subseteq V$ such that any distinct $u, v \in S$ have an edge between them in H. The goal in the maximum clique problem is to find, given a graph H, a clique of the largest possible size.

Clique is one of the classical NP-hard problems studied in the early 70's, and finding the largest possible clique in a given graph H is NP-hard. Today, we will see that even approximating the largest clique in a graph is NP-hard. For that, we introduce the appropriate gap notations for clique, and gap-preserving Karp reductions. For $0 < \beta \leqslant \alpha \leqslant 1$, an input to the problem gap-Clique$[\alpha, \beta]$ is a graph H promised to either contain a clique of fractional size at least α, or not contain a clique of fractional size β, and the goal is to distinguish between these two cases.

2.1 The Basic Hardness of Approximation Result for Clique

We prove the following result:

Theorem 2.1. *There are absolute constants $0 < \beta \leqslant \alpha \leqslant 1$ for which gap-Clique$[\alpha, \beta]$ is NP-hard.*

The proof of Theorem 2.1 is by a polynomial time reduction from Theorem 1.2. Namely, we show a polynomial time map from an instance Ψ of label cover to a graph H such that:

1. **Completeness:** If $\mathsf{val}(\Psi) = 1$, then $\mathsf{Clique}(H) \geqslant \alpha$.
2. **Soundness:** If $\mathsf{val}(\Psi) < 1-\varepsilon$, then $\mathsf{Clique}(H) < \beta$.

We leave it to the reader to verify that such reduction indeed implies that gap-Clique$[\alpha, \beta]$ is NP-hard.

Proof of Theorem 2.1. Let $\Psi = (G = (L \cup R, E, \Sigma_L, \Sigma_R, \Phi))$ be a label cover instance as in Theorem 1.2. We construct a graph $H = (V', E')$ as follows. For each edge $e \in E$ of Ψ and a pair of labels to its endpoints that satisfy the constraint on e, that is $(\sigma_1, \sigma_2) \in \Phi_e$, we create a vertex $v_{e,\sigma_1,\sigma_2} \in V'$. As for the edges in H, we connect v_{e,σ_1,σ_2} and $v_{e',\sigma_1',\sigma_2'}$ by an edge if e, σ_1, σ_2 and e, σ_1', σ_2' that are consistent. This completes the description of the reduction.

To get a intuition for what the edges represent, we give a few examples. Suppose we have an edge $e \in E$ in the original graph, and two distinct pairs of labels that satisfy it, $(\sigma_1, \sigma_2) \neq (\sigma'_1, \sigma'_2)$; then the vertices v_{e,σ_1,σ_2} and $v_{e',\sigma'_1,\sigma'_2}$ **do not** have an edge between them. Thus, in particular, a clique can contain at most a single vertex of the form v_{e,σ_1,σ_2} for each $e \in E$. We will often refer to the collection of vertices $\{v_{e,\sigma_1,\sigma_2}\}_{\sigma_1 \in \Sigma_L, \sigma_2 \in \Sigma_R}$ as the cloud of e, and in this language we have observed that the cloud of each e forms an independent set in H. More generally, if we have two edges $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$ sharing a vertex – say the left one, i.e. $u_1 = u_2$ – as well as pairs (σ_1, σ_2) satisfying e_1 and (σ'_1, σ'_2) satisfying e'_1, then v_{e,σ_1,σ_2} and $v_{e',\sigma'_1,\sigma'_2}$ are connected by an edge only if $\sigma_1 = \sigma'_1$. Thus, if we have a clique of vertices in H, then for each vertex in the original graph $u \in L$, all vertices v_{e,σ_1,σ_2} in the clique such that the left endpoint of e is u agree on σ_1.

We denote $k_L = |\Sigma_L|$ and $k_R = |\Sigma_R|$, and note that since each constraint Φ_e has k_L satisfying pairs, the number of vertices in H' is $k_L \cdot |E| = k_L \cdot m$. We now prove the completeness of the reduction for $\alpha = 1/k_L$ and $\beta = (1-\varepsilon)/k_L$.

Completeness: We show that if Ψ is satisfiable, then H contains a clique of size m. Indeed, let $A_L \colon L \to \Sigma_L$ and $A_R \colon R \to \Sigma_R$ be satisfying assignments, and define

$$C = \{\, v_{e,\sigma_1,\sigma_2} \mid e = (u,v), A_L(u) = \sigma_1, A_R(v) = \sigma_2 \}\,.$$

Then C is a clique in H, and $|H| = m$.

Soundness: We show that if Ψ is at most $(1-\varepsilon)$ satisfiable, then the largest clique in H has size at most $(1-\varepsilon)m$. We do so counter-positively: we assume that C is a clique of size larger than $(1-\varepsilon)m$, and deduce from it a pair of assignments A_L and A_R that satisfy more than $1-\varepsilon$ fraction of the constraints in Ψ.

Take C to be a clique of size larger than $(1-\varepsilon)m$ in H. By our earlier observation for each $u \in L$ there is a value $\sigma_u \in \Sigma_L$ such that, if an edge $e \in E$ contains u, say $e = (u, v)$, is such that the clique C contains some vertex from the cloud of e, then such vertex must be v_{e,σ_1,σ_2} for $\sigma_1 = \sigma_u$. Thus, we can define $A_L(u) = \sigma_u$. Similarly, for each $v \in R$ there is a value $\sigma_v \in \Sigma_R$ such that, if an edge $e \in E$ contains v, say $e = (u, v)$, is such that the clique C contains some vertex from the cloud of e, then such vertex must be v_{e,σ_1,σ_2} for $\sigma_2 = \sigma_v$. Thus, we can define $A_R(v) = \sigma_v$

By our earlier observations C may contain at most a single vertex from the cloud of each $e \in E$, hence it follows that there are more than $(1-\varepsilon)m$ clouds from which C contains a vertex. Let $E' \subseteq E$ be the set of $e \in E$ such that C contains some vertex from the cloud of e. We argue that A_L, A_R satisfy all edges in E', hence they satisfy at least $|E'|/m > 1-\varepsilon$ fraction of the constraints. Indeed, if $e \in E'$ then there is a vertex of the form v_{e,σ_1,σ_2} in C. Writing $e = (u, v)$, by construction (σ_1, σ_2) satisfies the constraint Φ_e, and by the choice of A_L and A_R we have that $A_L(u) = \sigma_1$ and $A_R(v) = \sigma_2$. □

Just like in Corollary 1.3, Theorem 2.1 directly implies that it is NP-hard to approximate the size of the largest clique in a graph within factor β/α where α, β are the number from Theorem 2.1. Inspecting, we see that the α and β we get yield that $\beta/\alpha = 1-\varepsilon$ where $\varepsilon > 0$ is some positive absolute constant. This means that getting arbitrary good approximation of clique is NP-hard.

2.2 Hardness Amplification for Clique

Is it possible, though, to approximate the size of the largest clique in a graph within a not-so-good factor, say 10, or 100? It turns out not to be possible, and to do so we amplify the result of Theorem 2.1

Theorem 2.2. *There are absolute constants $0 < \beta \leqslant \alpha \leqslant 1$ such that for all $t \in \mathbb{N}$, the problem gap-Clique$[\alpha^t, \beta^t]$ is NP-hard.*

Thus, we get that for all $t \in \mathbb{N}$, approximating the largest clique within factor $\beta^t/\alpha^t = (1-\varepsilon)^t$ is NP-hard, and as we may pick t to be as large as we wish (but constant), any constant factor approximation for clique is NP-hard.

Proof of Theorem 2.2. We show a reduction from Theorem 2.1. Namely, for each $t \in \mathbb{N}$, we show a polynomial time reduction from gap-Clique$[\alpha, \beta]$ to gap-Clique$[\alpha^t, \beta^t]$.

Given an instance $G = (V, E)$ of gap-Clique$[\alpha, \beta]$, we produce a graph $G' = (V', E')$ as follows. The vertices of G' are all t-tuple of vertices from G, that is

$$V' = \{ (v_1, \ldots, v_t) \mid v_i \in V \; \forall i = 1, \ldots, t \} .$$

As for the edges, we connect $(v_1, \ldots, v_t)$ and $(u_1, \ldots, u_t)$ by an edge if for all $i = 1, \ldots, t$, either $(v_i, u_i) \in E$ or $v_i = u_i$. This completes the description of the reduction.

Completeness: We show that if G contains a clique of size at least $\alpha |V|$, then G' contains a clique of size at least $\alpha^t |V'|$. Indeed, let $C \subseteq V$ be a clique of size at least $\alpha |V|$, and define

$$C' = \{ (v_1, \ldots, v_t) \mid v_i \in C \; \forall i = 1, \ldots, t \} .$$

Then $|C'| = |C|^t \geqslant \alpha^t |V|^t = \alpha^t |V'|$, and C' is a clique in G'.

Soundness: We show that if G does not contain a clique of size $\beta |V|$, then G' does not contain a clique of size $\beta^t |V'|$. Indeed, let C' be any clique in G', and define for each $i = 1, \ldots, t$ the set

$$C_i = \left\{ v \in V \mid \exists (v_1, \ldots, v_t) \in C' \text{ such that } v_i = v \right. .$$

In words, C_i is the set of all possible vertices that appear as the ith coordinate of some vertex in C'. Note that C_i forms a clique in G; indeed, otherwise we would have $v, u \in C_i$ that are not adjacent, and so we may find $(v_1, \ldots, v_t)$ and $(u_1, \ldots, u_t)$ in C' such that $v_i = v$ and $u_i = u$, and by definition of the graph these two vertices are not adjacent in G', in contradiction to the fact that C' forms a clique. Thus, $|C_i| < \beta |V|$.

To finish the proof, note that $C' \subseteq C_1 \times C_2 \times \ldots \times C_t$, hence

$$|C'| \leqslant \prod_{i=1}^{t} |C_i| < \prod_{i=1}^{t} \beta |V| = \beta^t \; |V'|^t .$$

□

From Theorem 2.1 we get the following immediate corollary.

Corollary 2.3. *For all $C > 1$, approximating the maximum clique in a graph within factor C is NP-hard.*

2.3 PCP and Hardness of Approximation

Corollary 2.3 is an amazing consequence of the theory of PCPs, and to date this is the only known approach to proving hardness of approximation results for clique. In fact, almost all hardness of approximation results use the theory of PCPs and start from Theorem 1.2.

At a high level, the proof of Corollary 2.3 proceeded via two steps. In the first step we proved a weak hardness of approximation result for clique (in the form of Theorem 2.1), and in the second step we amplified it into a strong hardness of approximation result. It turns out that replicating the first step can be done for a vast class of approximation problems, and these are often referred to as APX hardness results. These

type of results say that for many problems there exists some constant factor within which it is NP-hard to approximate the optimum solution. Many of the combinatorial optimization problems that you know (such as 3SAT, Vertex-Cover, Set-Cover, Max-Cut etc.) fall into this category, and are hence at least somewhat hard to approximate.

Getting strong hardness of approximation results requires more efforts and more ideas. We were fairly lucky in the case of clique, for which we could directly perform amplification. The situation is more complicated though for other combinatorial optimization problems, which motivates the questions of if there are stronger forms of the PCP theorem that imply strong hardness of approximation results.

3 Extreme Versions of the PCP Theorem

Inspecting Theorem 1.2, one may wonder what additional features of it would be of help when proving hardness of approximation results. The above example of clique already highlights one important such aspect regarding the soundness of the result, and more specifically whether it could be taken to be close to 0. Additionally, one may observe that if we wish to investigate super-constant factor approximations for clique – say we want to show it is NP-hard to approximate within factor n^{ε} where n is the number of vertices in the graph – we need the soundness to be related to the size of the instance.[1] Lastly, and this will only become more apparent once we see a few hardness of approximation results, one could hope that the structure of the constraints Φ_e to be as restrictive and as simple as possible.

We summarize this discussion by stating a few aspects in which one may try to improve upon Theorem 1.2, as well as some buzzwords that are related to them.

1. **Hardness amplification.** Are there forms of Theorem 1.2 with small soundness? Namely, is it true that for every $\varepsilon > 0$, there is $k \in \mathbb{N}$ such that the problem gap-Label-Cover$[1, \varepsilon]$ is NP-hard on instances with alphabet size at most k? We will see that the answer to this question is positive, and towards this end introduce a technique known as *parallel repetition*.

2. **Sub-constant error PCPs.** Are there forms of Theorem 1.2 in which the soundness is vanishing with the instance size? How about forms of the theorem in which the soundness is polynomially small in the instance size? Note that in such cases, the alphabet will also have to be of a size which is growing with the instance size.

   PCPs with sub-constant errors are known, and we have already seen some of the ideas that go into constructing them (such as list-decoding in the plane versus point test), but proving it requires much more effort. Getting sub-constant error PCPs with 2 queries is even harder, but it is known by now. As for PCPs with polynomially small error, this is a well known open problem in the theory of PCPs known as the sliding scale conjecture.

3. **The simplicity of the constraints.** Once we see several PCP reductions, you will see that on top on needing small soundness, these reductions heavily use the fact we have projection constraints. It took time to realize, but it turns out that the simpler the structure the constraints is, the more useful PCP result one gets.

   One of the most important examples for such structures are d-to-1 constraints, by which we mean that not only is the map $\phi_e \colon \Sigma_L \to \Sigma_R$ a projection map, but it is also "not very far" from being a

[1] In the case of clique such results are known while the corresponding forms of Theorem 1.2 are not, since there are ways to work-around this issue.

permutation. In the extreme case of $d = 1$, one indeed wants each constraint ϕ_e to be a permutation map; such PCPs are conjectured to exist but are not currently known. The statement that such PCPs exists is a well-known conjecture in complexity theory that goes by the name the Unique-Games Conjecture. For the case $d = 2$, one wants each constraint ϕ_e to be 2-to-1, namely that each $\sigma_2 \in \Sigma_R$ has two pre-images under Σ_L. Such PCPs were conjectured to exist in the same paper introducing the Unique-Games Conjecture, and by now it is known how to construct them.

In the rest of this course, we will mainly discuss points 1 and 3 above. In particular, starting from the next lecture we will discuss the parallel repetition theorem, and long-code framework and how to use it to prove some optimal hardness of approximation results. We will then discuss the Unique-Games Conjecture, some of its consequences and recent developments regarding it.

18.408 Topics in Theoretical Computer Science: Probabilistically Checkable Proofs
Fall 2022