Lecture 20

In this lecture we see the connection between **PCP** and hardness of approximation. In particular, we show that the **PCP** Theorem implies that MAX-3SAT cannot be approximated within $(1-\epsilon)$ for some $\epsilon > 0$, unless $\mathbf{P} = \mathbf{NP}$. Also, we show that Max Clique cannot be approximated within any constant factor.

1 Probabilistically Checkable Proofs

Recall the following definition and theorem:

Definition 1 *Given functions $r(n)$, $q(n)$, we say that $L \in \mathbf{PCP}(r(n), q(n))$ if there is a polynomial time probabilistic verifier V, which is given x, oracle access to a proof π, uses $r(|x|)$ random bits, reads $q(|x|)$ bits of π and such that*

$$x \in L \Rightarrow \exists \pi \cdot \mathbf{Pr}[V^\pi(x) = 1] = 1$$
$$x \notin L \Rightarrow \forall \pi \cdot \mathbf{Pr}[V^\pi(x) = 1] \leq \tfrac{1}{2}.$$

Theorem 1 (**PCP** Theorem)
$\mathbf{NP} = \mathbf{PCP}(O(\log n), O(1))$.

2 PCP and MAX-3SAT

2.1 Approximability

Definition 2 (MAX-3SAT) *Given a 3CNF formula φ (i.e. with at most 3 variables per clause), find an assignment that satisfies the largest number of clauses.*

Note that MAX-3SAT generalizes 3SAT, and so cannot be solved in polynomial time unless $\mathbf{P} = \mathbf{NP}$. However, it is easy to approximate it within a factor of 2:

Algorithm. Output the best solution between all-true and all-false.

Analysis. Every clause is satisfied by at least one of the two solutions, therefore one of the solutions satisfies at least half of the clauses.

It is possible to do better. The Karloff-Zwick algorithm runs in polynomial time and satisfies $\geq \frac{7}{8}$ times the optimal number of clauses.

Some problems admit polynomial time algorithms with approximation ratio $\geq (1-\epsilon)\ \forall \epsilon$. The **PCP** Theorem implies this is not the case for MAX-3SAT, as we see next.

2.2 Inapproximability

Theorem 2
The **PCP** *Theorem implies that there is an $\epsilon > 0$ such that $(1-\epsilon)$-approximation of MAX-3SAT is* **NP***-hard.*

Proof: Fix any **NP**-complete problem L. By the **PCP** Theorem, $L \in \mathbf{PCP}(O(\log n), O(1))$. Let V be the verifier for L.

Given an instance x of L, our plan is to construct a 3CNF formula φ_x on m variables such that (for some $\epsilon > 0$ to be determined)

$$\begin{array}{lll} x \in L & \Rightarrow & \varphi_x \text{ is satisfiable} \\ x \notin L & \Rightarrow & \text{no more than } (1-\epsilon)m \text{ clauses of } \varphi_x \text{ are satisfiable.} \end{array} \tag{1}$$

Without loss of generality, assume that V makes non-adaptive queries (by reading all the bits that could possibly be needed at once). This assumption is valid because the number of queries was a constant and remains a constant. Let q be the number of queries.

Enumerate all random strings R for V. The length of each string is $r(|x|) = O(\log |x|)$, so the number of such strings is polynomial in $|x|$. For each R_i, V chooses q positions $i_1, \ldots, i_q$ and a Boolean function $f_R : \{0,1\}^q \to \{0,1\}$ and accepts iff $f_R(\pi(i_1), \ldots, \pi(i_q))$.

We want a Boolean formula to simulate this. Introduce Boolean variables $x_1, \ldots, x_l$, where l is the length of the proof. Now we need a correspondence between the number of satisfiable clauses and the probability that the verifier accepts.

For every R we add clauses that represent $f_R(x_{i_1}, \ldots, x_{i_q})$. This can be done with 2^q SAT clauses. We need to convert clauses of length q to length 3 with additional variables. e.g. $x_2 \vee x_{10} \vee x_{11} \vee x_{12}$ becomes $(x_2 \vee x_{10} \vee y_R) \wedge (\overline{y}_R \vee x_{11} \vee x_{12})$. This requires at most $q2^q$ 3SAT clauses. In general there is a series of auxiliary variables for each R.

If $z \in L$, then there is a proof π such that $V^\pi(z)$ accepts for every R. Set $x_i = \pi(i)$ and auxiliary variables in the right way, all clauses are satisfied.

If $z \notin L$, then for every assignment to $x_1, \ldots, x_l$ (and to y_R's), the corresponding proof $\pi(i) = x_i$ makes the verifier reject for half of the $R \in \{0,1\}^{r(|z|)}$. For each such R, one of the clauses representing f_R fails. Therefore a fraction $\epsilon = \frac{1}{2}\frac{1}{q2^q}$ of clauses fails. □

It turns out if this holds for every **NP**-complete problem, then the **PCP** Theorem must be true.

Theorem 3
If there is a reduction as in (1) for some problem L in **NP***, then $L \in \mathbf{PCP}(O(\log n), O(1))$, i.e. the* **PCP** *Theorem holds for that problem.*

Proof: We describe how to construct a verifier for L. V on input z expects π to be a satisfying assignment for φ_z. V picks $O(\frac{1}{\epsilon})$ clauses of φ_z at random, and checks that π satisfies all of them. Randomness is $O(\frac{1}{\epsilon}\log m) = O(\log |z|)$. $O(\frac{1}{\epsilon}) = O(1)$ bits are read in the proof.

$$\begin{aligned} z \in L &\Rightarrow \varphi_z \text{ is satisfiable} \\ &\Rightarrow \exists \pi \text{ such that } V_\pi(z) \text{ always accept.} \end{aligned}$$

$$\begin{aligned} z \notin L &\Rightarrow \forall \pi \text{ a fraction } \tfrac{1}{\epsilon} \text{ of clauses are unsatisfied} \\ &\Rightarrow \forall \pi\ V^\pi(z) \text{ will reject with probability } \geq \tfrac{1}{2} \quad \text{(details omitted).} \end{aligned}$$

□

2.3 Tighter result

What is the best known value of ϵ in Theorem 2 ? We state without proof the following result:

Theorem 4 (Håstad)
There is a **PCP** *verifier for* **NP** *that uses $O(\log n)$ bits of randomness which, based on R, chooses 3 positions i_1, i_2, i_3 and a bit b and accepts iff $\pi(i_1) \oplus \pi(i_2) \oplus \pi(i_3) = b$. The verifier satisfies*

$$z \in L \Rightarrow \exists \pi \cdot \mathbf{Pr}[V^\pi(x) = 1] \geq 1 - \epsilon \tag{2}$$
$$z \notin L \Rightarrow \forall \pi \cdot \mathbf{Pr}[V^\pi(x) = 1] \leq \tfrac{1}{2} + \epsilon \tag{3}$$

Note the slightly non-standard bounds for the probabilities above. If we had "$= 1$" in Equation (2), we could find a proof π by solving a system of linear equations (which would imply $\mathbf{P} = \mathbf{NP}$). For every R, one can encode $x_{i_1} \oplus x_{i_2} \oplus x_{i_3} = b$ with 4 clauses in 3CNF.

If $z \in L$, then a fraction $\geq (1 - \epsilon)$ of clauses are satisfied.

If $z \notin L$, then for a $(\frac{1}{2} - \epsilon)$ fraction of R, $\frac{1}{4}$ of clauses are contradicted.

This is not producing a formula directly, but is enough to prove the hardness of approximation ratio:

$$\frac{1 - \frac{1}{4}(\frac{1}{2} - \epsilon)}{1 - \epsilon} = \frac{7}{8} + \epsilon'$$

3 Max Clique

3.1 Approximability

Definition 3 (Max Clique) *Given an undirected $G = (V, E)$, find the largest $C \subseteq V$ such that $\forall u, v \in C$, $(u, v) \in E$. When convenient, we use the complementary graph equivalent "Max Independent Set" which is the largest $I \subseteq V$ such that $\forall u, v \in I$, $(u, v) \notin E$.*

Finding a largest clique is **NP**-hard, so we ask about approximations. There is a simple $\frac{\log n}{n}$-approximation:

- arbitrarily partition the graph into subsets of size $\log n$

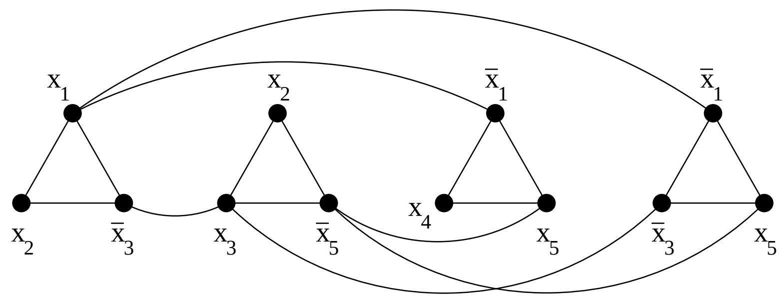

Figure 1: Graph construction corresponding to the 3CNF formula $\varphi = (x_1 \ \vee \ x_2 \ \vee \ \overline{x}_3) \wedge (x_2 \ \vee \ x_3 \ \vee \ \overline{x}_5) \wedge (\overline{x}_1 \ \vee \ x_4 \ \vee \ x_5) \wedge (\overline{x}_1 \ \vee \ \overline{x}_3 \ \vee \ x_5)$.

- solve exactly each subset
- report the largest clique found

Suppose there is a clique of size αn, then one of the subset contains a clique of size $\alpha \log n$, hence the approximation. Best known is a $\frac{(\log n)^2}{n}$-approximation algorithm.

3.2 Inapproximability

We apply the **PCP** Theorem to Max Independent Set. Given a 3CNF formula $\varphi = c_1 \wedge c_2 \wedge \ldots \wedge c_m$ over variables $x_1, \ldots, x_n$ arising from Theorem 2, we build a graph using the same construction as in the **NP**-completeness proof of Independent Set in Lecture 3 (see Figure 1).

The graph has $3m$ vertices. Note that G has an independent set of size m iff φ is satisfiable, and G has an independent set with k vertices iff φ has an assignment that satisfies k clauses. So the hardness results for MAX-3SAT translate directly to Max Independent Set. Below we show how this can be further improved.

Given $G = (V, E)$, consider $G' = (V', E')$ where $V' = V \times V$ and $((u_1, u_2), (v_1, v_2)) \in E'$ iff $(u_1, v_1) \in E$ OR $(u_2, v_2) \in E$. It can be shown that the size of the maximum independent set in G' is equal to (size of max ind. set in G)^2. As a sidenote, this result is not true for cliques (a clique in G' could be possibly larger than expected), unless we use a different definition of G'.

By a constant number of applications of the graph product operation, we get:

graph	G	G'	G''
size of ind. set if φ is satisfiable	m	m^2	m^4
size of ind. set if only $(1-\epsilon)m$ clauses of φ are satisfiable	$(1-\epsilon)m$	$(1-\epsilon)^2 m^2$	$(1-\epsilon)^4 m^4$

The ratio decreases without bound, therefore there can be no constant approximation algorithm for Max Clique. In fact:

THEOREM 5 (HÅSTAD)
If **NP** $\neq$ **ZPP**, *then* $\forall \epsilon > 0$, *Max Independent Set cannot be approximated within* $\frac{1}{n^{1-\epsilon}}$ *(n is the number of vertices).*