CS221: Computational Complexity Prof. Salil Vadhan

Lecture 33: Håstad's Optimal PCP,
More Hardness of Approximation

12/13 *Scribe: Vitaly Feldman*

Contents

1 Announcements

- There will be a section on Monday (the last section)
- Professor Vadhan's office hours next week are Monday 1-2.

2 PCP and Hardness of Approximation

Today we will show several connections of the PCP theorem to the hardness of approximating optimization problems (Recap: the PCP theorem states that $\mathbf{PCP}[\log n, O(1)] = \mathbf{NP}$). In the last lecture we have shown the first such connection, namely the following theorem

Theorem 1 *The PCP theorem is true if and only if there exists an "amplifying" reduction R mapping 3-CNF formulas to 3-CNF formulas and an $\epsilon > 0$ such that*

1. *R is polynomially computable;*
2. *if ϕ is satisfiable then $R(\phi)$ is satisfiable;*
3. *if ϕ is not satisfiable then no assignment satisfies more than $1 - \epsilon$ fraction of $R(\phi)$'s clauses.*

Another way to view the above R is as a reduction from 3SAT to a certain *promise* problem. The promise problem is GapMax3SAT$_\epsilon$ and is defined by specifying the "yes" instances

$$Y = \{\phi \mid \phi \text{ is satisfiable}\}$$

and the "no" instances

$$N = \{\phi \mid \text{every assignment satisfies at most } (1-\epsilon) \text{ fraction of } \phi\text{'s clauses}\} .$$

A promise problem is a decision problem in which two disjoint sets Y and N are defined and "it is promised" that the input will come from one of these sets. (It is required to decide from which set the input comes). This definition differs from the standard language decision problem since there might exist instances which are not included in $Y \cup N$ and any answer can be given for those instances. Promise problems are often useful for capturing the approximation problems as decision problems (as in our case). With this definition, it is clear that Theorem 1 is equivalent to stating that $\textsc{GapMax3SAT}_\epsilon$ is **NP**-complete (for some $\epsilon > 0$) if and only if the PCP theorem is true.

Proof of Theorem 1: We have seen the first direction in the previous lecture. For the other direction we assume that R is a reduction with the above properties. With such a reduction a PCP proof for satisfiable ϕ is the assignment that satisfies $R(\phi)$. The verifier will use $O(\log n)$ random bits to choose a random clause of $R(\phi)$. It will then read the values assigned to the 3 variables of this clause and accept if and only if the clause is satisfied. Clearly, this verification procedure is complete, so we argue soundness. If ϕ is not satisfiable then, by the 3-rd property of R, the probability that the verifier will accept is at most $1 - \epsilon$. ■

Note that the PCP we construct above has query complexity 3 (rather than an arbitrary constant). Thus:

Corollary 2 *Assuming the PCP theorem,* $\mathbf{NP} = \mathbf{PCP}_{1,1-\epsilon}[\log n, 3]$ *for some* $\epsilon > 0$

Proof: Follows from the fact that the existence of the "amplifying" reduction can be proved using $O(1)$-bit version of the PCP theorem and implies the 3-bit version. ■

Corollary 3 *If the PCP theorem is correct then there exists an ϵ such that it is* **NP**-*hard to approximate* $\textsc{Max3SAT}$ *to within* $1 + \epsilon$.

Proof: If we can efficiently approximate $\textsc{Max3SAT}$ to within $\frac{1}{1-\epsilon} = 1+\epsilon'$ then we can distinguish between "yes" and "no" instances (in polynomial time).

With this strong connection between the approximation factor and ϵ in the PCP theorem it is important to optimize this ϵ.[1] Therefore a lot of work during the last decade was devoted to improving the parameters in the PCP theorem. One of the culminations of this work is the following theorem:

Theorem 4 (Håstad) *For every constant* ϵ, $\mathbf{PCP}_{1-\epsilon,\frac{1}{2}+\epsilon}[\log n, 3] = \mathbf{NP}$ *and if for the verifier's coin tosses r the generated positions are $i_1, i_, i_3$ then the verifier accepts if and only if* $\pi_{i_1} \oplus \pi_{i_2} \oplus \pi_{i_3} = b(r)$ *for some Boolean function b.*

3 Applications of the PCP Theorem for Proving Hardness of Approximation

The most direct application of Theorem 4 is the hardness of approximating E3LIN2.

[1] Unfortunately in our construction (in the proof of the first direction) ϵ goes to zero exponentially with the number of queries we allow to the verifier and linearly in the soundness parameter.

Definition 5 E3LIN2 – *given a system of linear equations mod 2, with exactly 3 variables per equation, find the assignment that maximizes the number of equations satisfied.*

Remark 6 The following related facts are known about this problem:

- Testing whether all the equations are satisfiable is very easy (for example by Gaussian elimination).
- Finding the assignment that satisfies the maximum number of equations is long-known to be **NP**-hard.
- There is a trivial 2-approximation for this problem. Either all zeros or all ones will satisfy at least half of the equations.

The remarkable corollary of the Håstad's result is that the trivial 2-approximation is tight.

Theorem 7 *For every positive constant ϵ, it is* **NP***-hard to approximate E2LIN3 within the factor of* $2-\epsilon$.

Proof: The reduction from SAT is straightforward. Given a formula ϕ construct a system of equations which represent all the equations checked by the $\mathbf{PCP}_{1-\epsilon,\frac{1}{2}+\epsilon}[\log n, 3]$ verifier for SAT on ϕ for different values of random bits (there are at most $2^{O(\log n)} = \text{poly}(n)$ of them). If ϕ is satisfiable then there exists an assignment that satisfies at least $1-\epsilon$ fraction of equations and if ϕ is not satisfiable then there exist no assignment that satisfies more than $\frac{1}{2}+\epsilon$ fraction of equations. A $\frac{1-\epsilon}{\frac{1}{2}+\epsilon} = 2-\epsilon'$-approximation of E3LIN2 is sufficient to distinguish between these two situations. ■

Another simple corollary of the Hästad's theorem is that MAX3SAT is **NP**-hard to approximate within $\frac{8}{7}-\epsilon$ for any positive constant ϵ. This result is also tight since there is an $\frac{8}{7}$-approximation algorithm.

Proof: We prove this result using an *approximation-preserving reduction* from E3LIN2. Given a system of linear equations we map each equation $x \oplus y \oplus z = b$ to AND of four clauses that represents the Boolean function which checks whether $x \oplus y \oplus z = b$:

$$(x \oplus y \oplus z = 0) \mapsto (\neg x \vee y \vee z) \wedge (x \vee \neg y \vee z) \wedge (x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z),$$

and similarly for $x \oplus y \oplus z = 1$. (These are just the CNF representations of the functions $\neg(x \oplus y \oplus z)$ and $x \oplus y \oplus z$, with one clause to rule out each non-satisfying assignment.) Now, if an assignment satisfies at least $1-\epsilon$ fraction of the equations then it satisfies at least $1-\epsilon$ fraction of the corresponding clauses. If at most $\frac{1}{2}+\epsilon$ fraction of the equations are satisfied then at least $\frac{1}{2}-\epsilon$ fraction of the equations are unsatisfied and therefore at least $\frac{1}{4}\left(\frac{1}{2}-\epsilon\right) = \frac{1}{8}-\frac{\epsilon}{4}$ fraction of clauses are not satisfied, that is, at most $\frac{7}{8}+\frac{\epsilon}{4}$ fraction of clauses are satisfied. Thus an $(\frac{8}{7}-\epsilon')$-approximation will be sufficient to distinguish between these two situations. ■