CSE 533: The PCP Theorem and Hardness of Approximation **(Autumn 2005)**

Lecture 10: Hardness of approximating clique, FGLSS graph

Nov. 2, 2005

Lecturer: Venkat Guruswami and Ryan O'Donnell *Scribe: Ioannis Giotis*

1 PCP Theorem recap

We will review the most important points in the PCP theorem proof (a description of this also appears at the end of the notes for the previous lecture). Let's start by noting that the main theme in the proof was a gap producing reduction. Unlike the original PCP theorem proof which created the gap in one step and the bulk of the proof was about fixing all the parameters, mainly the alphabet size, the proof we presented was creating the gap slowly in a sequence of steps. At each step, we were transforming G into G', where both of them were constraint graphs over some alphabet Σ_0. Beginning with $\text{gap}(G) \geq 1/n$, we applied the process $O(\log n)$ times where at each step the following properties were maintained.

$$
\begin{aligned}
\text{gap}(G') &\geq 2\text{gap}(G) \\
\text{size}(G') &= O(\text{size}(G))
\end{aligned}
$$

Pictorially, the process looks like this

$G \longrightarrow H$	$\longrightarrow$	H'	$\longrightarrow$	G'
$\downarrow$	$\uparrow$	$\downarrow$	$\uparrow$	$\downarrow$
constant degree expander	gap amplification $\text{gap}(H') \geq t \cdot \text{gap}(H)$	constraint graph over Σ',$\lvert\Sigma'\rvert \gg \lvert\Sigma\rvert$	• composition with an assignment tester • Error Correcting Codes	constraint graph again over Σ_0

The most important technical tools we used in the proof were

- Edge-expansion and behavior of random walks in expanders
- Error-correcting codes (for modular composition using an assignment tester)
- Construction of the assignment tester (which was a "direct", albeit inefficient, PCP reduction)
  - Linear functions, Hadamard code
  - Linearity (or Hadamard codeword) testing, self-correction of Hadamard code.
  - Fourier Analysis over the Boolean hypercube
  - Basic linear algebra

2 Hardness of approximating $CLIQUE$

Remark 2.1. *The PCP theorem is a great tool that can be used to examine the approximability of problems. As in the typical complexity theory, once we have one Hardness of Approximation result, we are able to extend this to many problems using the method of reduction. In fatc, in many cases we only have to look at the existing $NP-completeness$ reductions and find that these reductions "preserve gaps".*

We will proceed to present the hardness of approximating $CLIQUE$ via such a gap-preserving reduction from $3SAT$. The "usual" NP-completeness reduction from $3SAT$ to $CLIQUE$ itself is gap-preserving and can prove the result below (Exercise). We will describe a slightly different reduction with an eye towards generalizing it to work with any PCP as starting point.

Definition 2.2. $GapCLIQUE_\rho$*:Given $\rho \leq 1$, a graph G and an integer k, we wish to distinguish between the following two cases*

YES: $\omega(G) \geq k$*, G has a clique of size k*

NO: $\omega(G) < \rho k$*, G has no clique of size $\geq \rho k$.*

To show the hardness of approximating $CLIQUE$, we will now show that the respective gap problem is NP-hard.

Theorem 2.3. $\exists \rho < 1$*, such that $GapCLIQUE_\rho$ is NP-hard.*

Proof. Given a formula ϕ we wish to output a graph G and an integer k. In particular we wish to reduce in the following way

$$\phi \in 3SAT \Rightarrow \omega(G) \geq k \tag{1}$$
$$\phi \notin 3SAT \Rightarrow \omega(G) < \rho k \tag{2}$$

Thanks to the PCP theorem we can instead use a reduction that begins with a gapE3SAT instance. In other words, it suffices to give a polynomial time reduction of the form $\psi \rightarrow < G, k >$ such that

$$\psi \in E3SAT \Rightarrow \omega(G) \geq k \tag{3}$$
$$\text{Any asignment satisfies} < (1-\epsilon) \text{ fraction of } \psi\text{'s clauses} \Rightarrow \omega(G) < \rho k \tag{4}$$

Let ψ be on n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_n$. We create m rows of vertices, one for each clause. Each row consists of 7 vertices corresponding to all the satisfying assignments that satisfy the respective clause. Recall that there are precisely 7 partial assignments on 3 variables that make a clause satisfiable and 1 that makes it unsatisfiable. For example, for a clause of the form $(x_1, x_2, \overline{x_3})$ the 7 vertices would correspond to the assignments $000, 010, 011, 100, 101, 110, 111$.

Every row in the graph will be an independent set. Edges between rows are joining vertices whose assignments are consistent. For example a vertex that corresponds to a clause $(x_1, x_2, \overline{x_3})$

and an assignment 011 will be connected to a vertex of a clause $(x_1, \overline{x_4}, x_5)$ and of assignment 000 as they share the same assignment on x_1. If there are no shared variables between the vertices, their assignments are considered consistent and hence we place an edge.

Since every row is an independent set, it is clear that the largest clique in the graph will be of size at most m. On the other hand, if $\psi \in \mathit{E3SAT}$ it is easy to see that by picking one vertex from each row corresponding to the satisfying assignment, we will form a clique of size m.

For the other direction we will show that if $\omega(G) \geq \rho m$, then ψ is ρ-satisfiable. To see this, we observe cliques in G correspond to partial assignments to variables of ψ. Such an assignment is obtained as follows. If the clique contains a vertex from the row corresponding to some clause C, then the three variables in that clause are assigned based on which of the 7 assignments the clique picked in that row. This assignment is consistent since C is a clique, and thus a variable is given the same value in all the clauses containing it that are picked by C. Extend the partial assignment to all variables in an arbitrary way. The resulting assignment satisfies all the clauses for which C contained a vertex from the corresponding row. This means at least ρm clauses are satisfied by the assignment. □

2.1 A general reduction from a PCP verifier

An alternative way to look at the above is as a prover-verifier test. The verifier randomly picks clauses and verifies them and we can think of each row as a random test. We'll now present more formally a general reduction from a PCP verifier to CLIQUE.

Suppose $\mathit{3SAT} \in \mathrm{PCP}_{1,s}[r, q]$. This means that for any 3CNF formula ϕ, there is a probabilistic polynomial time verifier V that checks a binary proof π claiming that $\phi \in \mathit{3SAT}$ by tossing r random coins, picking (up to) q locations of π to query based on r, ϕ, and then accepting/rejecting based on the bits in those q locations. The verifier V satisfies the following requirement:

$$\phi \in \mathit{3SAT} \Rightarrow \exists \pi \ s.t.\ V \text{accepts with prob } 1 \tag{5}$$

$$\phi \notin \mathit{3SAT} \Rightarrow \forall \pi\ ,\ V \text{accepts with prob } < s \tag{6}$$

We call r and q respectively the randomness complexity and query complexity (or number of queries) of the verifier V.

Remark 2.4. *We can now state the PCP theorem as*

$$\textit{PCP theorem: } \exists q = O(1)\ s.t.\ \mathit{3SAT} \in \mathrm{PCP}_{1,1/2}[O(\log n), q]$$

The previous reduction from gap3SAT to CLIQUE can be generalized by defining a graph based on any PCP verifier (with the earlier construction being the one for the "simple" verifier that picked a random clause and checked that it was satisfied). Such a graph is called the FGLSS graph after Feige, Goldwasser, Lovasz, Safra and Szegedy who first realized the connetion between clique approximation and probabilistic proof checking in their seminal FOCS 1991 paper that spurred intense activity on PCPs and ultimately the discovery of the PCP theorem. Even though we presented the natural connection between NP-hardness of gap3SAT and PCP at the very beginning of this class, in fact the initial connection between approximation and PCPs was for gapCLIQUE, via the FGLSS graph, and only later was the connection to gap3SAT realized.

We now describe the $FGLSS$ graph for a PCP verifier with randomness and query complexi r, q respectively. The graph consists of $R = 2^r$ rows of vertices, each row with $\leq 2^q$ vertices. The vertices in each row correspond to the views on queried bits that make the verifier accept. We place edges between vertices that represent consistent partial views. In particular, this means there are no edges within a row. Two rows that query disjoint sets of bits have a complete bipartite graph between them.

As with the reduction from $3SAT$, cliques in the FGLSS graph correspond to partial proofs and their size corresponds to the number of random choices of the verifier for which it accepts this partial proof. In particular, the existence of a clique of size M implies the existence of a proof that V accepts with probability at least $M/2^r$, and vice versa.

We now see that

$$If\ 3SAT \in \mathrm{PCP}_{1,s}[r, q] \tag{7}$$
$$then\ \ \phi \in 3SAT \Rightarrow \omega(G) = 2^r = R \tag{8}$$
$$\phi \notin 3SAT \Rightarrow \omega(G) \leq sR \tag{9}$$

The construction of the FGLSS graph can be performed in $2^{r+q} \cdot \mathrm{poly}(n)$ time. Using the above, we can thus conclude the following connection between PCPs and hardness of approximation $CLIQUE$.

Theorem 2.5. *If* $3SAT \in \mathrm{PCP}_{1,s}[O(\log n), O(\log n)]$ *then* $GapCLIQUE_s$ *is* NP*-hard.*

A simple serial repetition applied to the PCP theorem implies:

Theorem 2.6. *For every integer* k*,* $3SAT \in \mathrm{PCP}_{1,1/2^k}[O(k \log n), O(k)]$

Picking k to be an arbitrary large constant we conclude

Theorem 2.7. $\forall \epsilon > 0$*,* $GapCLIQUE_\epsilon$ *is* NP*-hard.*

To get a superconstant gap, we need to pick k to be superconstant. But this makes the randomness $r = O(k \log n)$ super-logarithmic and the reduction no longer runs in polynomial time. But we can conclude the following.

Theorem 2.8. $\forall \epsilon > 0$*,* $GapCLIQUE_{2^{-\log^{1-\epsilon} N}} \notin P$ *unless* $NP \subseteq \bigcup_{c \geq 0} TIME(2^{\log^c n})$*.*

Proof. Let $c \geq 1$ be an arbitrary integer. Pick $k = \log^c n$ where n is the size of the $3SAT$ formula. Then

$$N \leq 2^{r+q} < 2^{O(\log^{c+1} n)}$$

We then have

$$gap = \frac{1}{2^{\log^c n}} \simeq \frac{1}{2^{\log^{\frac{c}{c+1}} N}}$$

Picking c larger and larger, we get the claimed gap, and the reduction runs in quasi-polynomial, i.e., $2^{\mathrm{poly}(\log n)}$ time. $\square$

Once we show that $gapCLIQUE_{1,s}$ is NP-hard for some constant $s < 1$, the above strong hardness results can also be obtained using graph products (see the problem set). Moreover, the hardness factor can be strengthened to $n^{-\delta}$ for some $\delta > 0$ (again, see problem set). Much stronger hardness bounds that rule out a $n^{\epsilon-1}$-approximation algorithm for any $\epsilon > 0$ are known. However this is much harder to prove and we won't discuss this in the course.

3 Overview of the rest of the course

For the rest of the class, we will present hardness of approximation results and reap some of the benefits of our hard work in proving the PCP theorem. Unlike decision problems where we have a nice classification of most problems, for lots of approximation problems we don't really know where to classify them. Let's see the various "hardness" categories.

1. **Hardness of decision problems.** Here we have the usual classification of decision problems with the "mother" of all problems being $3SAT$.

2. **Slight hardness of approximation results.** In this category we consider problems for which hardness of $(1-\epsilon)$-approximating is shown for some $\epsilon > 0$. Alternatively, these are the problems for which there is no Polynomial Time Approximation Scheme (PTAS). The mother of all problems in this category is $GapE3SAT_{1,0.99}$.

3. **Optimal (very strong) hardness of approximation results.** This category captures the problems for which very strong hardness results are known. As an example consider $E3LIN$.

   Definition 3.1. *E3LIN: Given a bunch of 3 variable linear equations* $\bmod 2$ *(each of the form* $x_{i_1} \oplus x_{i_2} \oplus x_{i_3} = b_i$ *where* $b_i \in \{0, 1\}$*), we wish to find an assignment that satisfies a maximum number of the equations.*

   It is easy to see that there exists a $1/2$ approximation algorithm for $E3LIN$; just pick a random assignment (or alteratively, when each equation has exactly three distint variables, we can take the better of the all 0's and the all 1's assignment). On the other hand, we have the following theorem due to Håstad.

   Theorem 3.2. *For all* $\epsilon > 0$*,* $Gap - E3LIN_{1-\epsilon,1/2+\epsilon}$ *is* NP*-hard.*

   Corollary 3.3. *For any* $\epsilon > 0$*, there is no polynomial time* $(1/2+\epsilon)$*-approximation algorithm for E3LIN unless* $P = NP$*.*

   The mother of all problems in this category (of very strong or tight inapproximability results) is a constraint graph problem known as $Label\ Cover$. We will discuss hardness results for this problem beginning in the next lecture.