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

Lecture 15: Set Cover hardness and testing Long Codes

Nov. 21, 2005

Lecturer: Venkat Guruswami *Scribe: Atri Rudra*

1 Recap

We will first recall the reduction from Label Cover to Set Cover that was covered in the last lecture. We start with a label cover instance

$$\mathcal{L} = (G = ((V_1, V_2), E), \Pi, \Sigma)$$

(where $|\Sigma| = m$) and we get an instance of set cover

$$\mathcal{S} = (U = E \times B, \{S_{wa} \mid w \in V_1 \cup V_2, a \in \Sigma\}).$$

Recall that $(B; C_1, \ldots, C_m)$ was an (m, ℓ)-system, that is, if the union of at most ℓ sets of the form C_i or $\overline{C}_j$ is B then the collection must contain both C_k and $\overline{C}_k$ for some $1 \leq k \leq m$. Let $Value(\mathcal{L})$ denote the largest fraction of edges satisfied by an assignment of labels to $V_1 \cup V_2$. We now recall the completenss and soundness claims from last class:

1. If $Value(\mathcal{L}) = 1$ then $\mathcal{S}$ has a set cover of size $\mid V_1 \mid + \mid V_2 \mid$.
2. If $Value(\mathcal{L}) < \epsilon$ then all set covers of $\mathcal{S}$ have size at most $\frac{\ell}{8}(\mid V_1 \mid + \mid V_2 \mid)$, provided $\frac{2}{\ell^2} > \epsilon$.

Thus, we have a gap of $\frac{\ell}{8}$.

2 Completing the Hardness of approximation of Set Cover

To complete the reduction we start with the fact that for every $\epsilon > 0$, $\mathrm{GapLC}_{1,\epsilon}$ is NP-hard. As was mentioned earlier, one can construct an (m, ℓ)-system with $|B| = O(2^{2\ell} m^2)$ (more on this soon). If ℓ and m are constants then the size of $\mathcal{S}$ is $O(|E|)$ and the reduction proceeds in polynomial time. Finally for any constant c, setting $\ell = 8c$ and $\epsilon < 2/\ell^2$, we have

Theorem 2.1. *For any constant c, it is* NP*-hard to approximate* SET-COVER *within a factor of c.*

In fact, under a stronger assumption, we have the following result–

Theorem 2.2. *There exists a $c > 0$ such that* SET-COVER *cannot be approximated within a factor of $c \log N$ (where N is the universe size) unless* $\mathrm{NP} \subseteq \mathrm{TIME}(n^{O(\log \log n)})$.

Proof. We will investigate the parameters in more details to prove the above result — the reduction itself is the same as in Theorem 2.1.

To start with $\mathrm{GapLC}_{1,\epsilon}$ being NP-hard, we used Raz's parallel repetition theorem with $k = O(\log(1/\epsilon))$ repetitions, which implies that $|E| = n^{O(k)} = n^{O(\log(1/\epsilon))}$ and $m = 2^{O(k)} = (1/\epsilon)^{O(1)})$. Further, we will show in Section 2.1 that an (m, ℓ)-system B exists with $|B| = O(2^{2\ell}m^2)$. Hence the universe size is

$$N = |U| = |E||B| = n^{O(\log(\frac{1}{\epsilon}))}2^{O(\ell)}\left(\frac{1}{\epsilon}\right)^{O(1)}.$$

Further, the running time is $N^{O(1)}$. Finally, if we have $\epsilon < 2/\ell^2$, then we will have a gap of $\ell/8$.

To complete the proof we instantiate ℓ and ϵ:

$$\ell = \log n \log\log n,$$

$$\epsilon = \frac{1}{\log^3 n}.$$

Note that with the above, $\epsilon < 2/\ell^2$ holds. Further, $N = n^{O(\log\log n)}2^{O(\log n \log\log n)} = 2^{O(\ell)}$. Thus, the gap is $\ell/8 = \Omega(\log N)$ as required. Finally the run time of the reduction is $N^{O(1)}$=$n^{O(\log\log n)}$ which is the reason for the assumption $\mathrm{NP} \not\subseteq \mathrm{TIME}(n^{O(\log\log n)})$. □

2.1 Construction of an (m, ℓ)-system

We begin with a related notion of (m, ℓ)-*universal family*.

Definition 2.3. *Let B be a collection of binary strings of length m, that is, $B \subseteq \{0,1\}^m$. B is an (m, ℓ)-universal family if for all $1 \le i_1 < i_2 \cdots < i_\ell \le m$ and every possible $a \in \{0,1\}^\ell$, there exists $x \in B$ such that $x_{i_1} = a_1, x_{i_2} = a_2, \ldots, x_{i_\ell} = a_\ell$.*

The following lemma shows that the notions of (m, ℓ)-system and (m, ℓ)-universal family are closely related.

Lemma 2.4. *Let B be an (m, ℓ)-universal family and define $C_i = \{x \in B \mid x_i = 1\}$. Then $(B; C_1, \ldots, C_m)$ is an (m, ℓ)-system.*

Proof. For the sake of contradiction, assume that there exist $D_{i_1}, D_{i_2}, \ldots, D_{i_\ell}$ such that $D_{i_1} \cup D_{i_2} \cup \cdots \cup D_{i_\ell} = B$, where each D_{i_j} is either C_{i_j} or $\overline{C}_{i_j}$ (note that this implies that there are no j and k such that $D_{i_j} = \overline{D}_{i_k}$). Define

$$a_j = \begin{cases} 0 & \text{if } D_{i_j} = C_{i_j} \\ 1 & \text{if } D_{i_j} = \overline{C}_{i_j} \end{cases}$$

As B is an (m, ℓ)-universal family, there exists an $x \in B$ such that $x_{i_j} = a_j$ for $1 \le j \le \ell$. Now note by construction, $x \notin D_{i_j}$ for any j, which implies that $x \notin \cup_{j-1}^{\ell} D_{i_j} = B$: a contradiction. □

Armed with this lemma, we will now just look for an (m, ℓ)-universal family.

Theorem 2.5. *There exist explicit (m, ℓ)-universal family B of size $O(2^{2\ell}m^2)$. Moreover, B can be constructed in $2^{O(\ell)}m^{O(1)}$ time.*

Proof. (Sketch) We say a collection $B \subseteq \{0,1\}^m$ is (ℓ, γ)-independent if the following holds for every $i_1, i_2, \ldots, i_\ell$ and $a \in \{0,1\}^\ell$:

$$| \Pr_{x \in B} [x_{i_1} = a_1 \wedge \cdots \wedge x_{i_\ell} = a_\ell] - \frac{1}{2^\ell} | \leq \gamma.$$

Note that if B is (ℓ, γ)-independent for $\gamma < 2^{-\ell}$, then B must be an (m, ℓ)-universal family.

The high level view of the construction is then as follows:

1. Start with an ϵ-biased family $B \subseteq \{0,1\}^m$, i.e., a collection B such that for every nonzero linear function $L_S : \{0,1\}^m \to \{0,1\}$, $L_S(x) = \langle x, \chi_S \rangle$,

   $$| \sum_{x \in B} (-1)^{L_S(x)} | \leq \epsilon \, .$$

   That is, with respect to linear tests, the elements of B on average behave like random elements and have at most ϵ bias. Explicit constructions of such an ϵ-biased family of size $O\left(\frac{m^2}{\epsilon^2}\right)$ are known, see [1]. (When the elements of such a B are written in the form of a matrix with m columns, its columns span a linear code in which all nonzero codewords have (normalized) Hamming weight between $(1/2 - \epsilon)$ and $(1/2 + \epsilon)$.)

2. By Vazirani's XOR lemma (cf. [1]), an ϵ-biased family *is* itself $(\ell, (1 - 2^{-\ell})\epsilon)$-independent for any ℓ.

3. Choose $\epsilon = 2^{-\ell}$ to conclude that such a B must be an (m, ℓ)-universal family of size $O(m^2 2^{2\ell})$.

□

Remark 2.6. *It can be shown by (an easy application of) the probabilistic method that there exist an (m, ℓ)-universal family of size $O(2^\ell \ell \log m)$.*

3 Hardness of approximation of E3-LIN-2

Recall the E3-LIN-2 problem– given linear equations of the form $x_{i_1} + x_{i_2} + x_{i_3} = b_i \pmod 2$, find a binary assignment to the variables which satisfies as many equation as possible.

We will now look at Håstad's PCP, where each check that the verifier makes is of the XOR of 3 bits.

Theorem 3.1 (Håstad 1997)**.** *([2])For every $\epsilon, \delta > 0$, $\mathrm{NP} \subseteq \mathrm{PCP}_{1-\epsilon, 1/2+\delta}[O(\log n), 3]$, where the proofs are binary and every check made by the verifier is of the form "XOR of three proof bits equals some value".*

Remark 3.2. *The above result is tight, that is,* $s/c \geq 1/2$ *for any* $\mathrm{PCP}_{c,s}[O(\log n), 3]$ *unless* $\mathrm{P} = \mathrm{NP}$.

Remark 3.3. *One can get* $c = 1$ *and* $s = 1/2 + \delta$ *for arbitrarily small* $\delta > 0$ *with* 3 *queries. However, the checks are not XORs of proof bits (since they* cannot *be!)*

For the rest of this and the next lecture, we will prove Theorem 3.1. As "usual", our starting point would be the fact that $\mathrm{GapLC}_{1,\gamma}$ is NP-hard for all $\gamma > 0$.

To begin with, let us look at a natural verification strategy for a label cover instance $((G = (V_1, V_2), E), \Sigma, \Pi)$ given a labeling σ:

1. Pick a random edge $e = (u, v) \in E$.
2. Check $\Pi_e(\sigma(u)) = \sigma(v)$.

The above test has completeness 1 and soundness γ. However, the two queries are over a larger alphabet (recall that $|\Sigma| = \left(\frac{1}{\gamma}\right)^{O(1)}$).

Thus, the basic idea is to "simulate" the two queries over the larger alphabet Σ by reading just 3 bits– however, we lose on soundness ($1/2$ instead of γ). Note that the situation is similar to that of Assignment Testers where we could not afford to look at the complete assignment. In particular we need to check if $\Pi_e(a) = b$ without reading all of a and b. We will work with suitable encdings of a and b which leads us to the next section.

4 Long codes

We will use *long codes*. For any $a \in \Sigma = \{1, \ldots, m\}$, its corresponding long code is defined as

$$\mathrm{LONG}(a) = \mathrm{Had}(\vec{e}_a) = \langle x \cdot \vec{e}_a \rangle_{x \in \{0,1\}^m} = \langle x_a \rangle_{x \in \{0,1\}^m},$$

where in the above $\vec{e}_a$ is the vector in $\{0,1\}^m$ which has zeroes everywhere except at position a. Note that the Hadamard code maps m bits to 2^m bits while the long code maps m elements (or equivalently $\log m$ bits) to 2^m bits. In fact, the long code is so named because it is the longest possible code without repeating bits in the codeword.

We will find it more convenient to work with the ± 1 notation for Boolean values. Accordingly, it is useful to think of the long code as a function, $\mathrm{LONG}(a) = A : \{1, -1\}^m \to \{1, -1\}$ such that $A(x) = x_a$ (also called the dictatorship function).

4.1 Testing a long code

Let $A : \{1, -1\}^m \to \{1, -1\}$ be a purported long code. If indeed it is the long code of some a, then $A(x) = x_a$, $A(y_a) = y_a$ and $A(xy) = x_a y_a = A(x)A(y)$. Thus, one might propose the test $A(x)A(y)A(xy) = 1$. However, there is a problem with this test– the above test is the exactly the BLR test and thus, all linear functions will also pass the test with probability 1 which is not good.

We now consider the same BLR test with a "twist". Pick $\mu \in \{1,-1\}^m$ as follows (each bit is chosen independently)

$$\mu_i = \begin{cases} 1 & \text{with probability } 1-\epsilon/2 \\ -1 & \text{with probability } \epsilon/2 \end{cases}$$

It is easy to see that $\mathbb{E}[\mu_i] = 1-\epsilon$. We now consider the following test.

1. Pick $x, y \in \{1,-1\}^m$ uniformly at random.
2. Pick $\mu \in \{1,-1\}^m$ as above.
3. Check if $A(x)A(y)A(xy\mu) = 1$.

4.1.1 Completeness of the test

Suppose $A = \mathrm{LONG}(a)$. Then the test is equivalent to checking if

$$x_a y_a x_a y_a \mu_a = \mu_a = 1.$$

By the choice of μ the above happens with probability $1 - \frac{\epsilon}{2} > 1 - \epsilon$.

4.1.2 Soundness of the test

As we saw before, in some sense, the "bad" case for the test is when A is actually a linear function. To develop some intuition for the soundness of the test, let us consider the case $A = \chi_S$ such that $| S |$ is large. In this case the test is the same as

$$\chi_S(x)\chi_S(y)\chi_S(xy\mu) = \chi_S(\mu) = 1.$$

In other words, the equality being tested for is

$$\chi_S(\mu) = \prod_{i \in S} \mu_i = 1.$$

So the probability of the test accepting equals

$$\mathbb{E}\left[\frac{1+\chi_S(\mu)}{2}\right] = \frac{1}{2} + \frac{1}{2} \cdot \mathbb{E}[\chi_S(\mu)] \, .$$

Now consider

$$\mathbb{E}[\chi_S(\mu)] = \mathbb{E}\left[\prod_{i \in S} \mu_i\right] = \prod_{i \in S} \mathbb{E}[\mu_i] = (1-\epsilon)^{|S|}.$$

Note that the above expression goes to zero for large $|S|$. In fact, the above is at most ϵ for $|S| \gg \frac{1}{\epsilon}$. In other words the test accepts with probability at most $1/2 + \epsilon$ which gives us the claimed soundness.

Linear functions χ_S for small $|S|$ still pass the test with good probability (and for $|S| \neq 1$ they are $1/2$-far from dictator functions). But this will be okay for us since in this case we can "list-decode" A into the subset $S \subseteq \{1, 2, \ldots, m\}$, which has a small number of elements (and most importantly this number is independent of m).

We will formalize this argument and use it to construct a 3-query PCP in the next lecture.

References

[1] N. Alon, O. Goldreich, J. Hastad, and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures and Algorithms*, 3:289–304, 1992.

[2] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.