CS880: Approximations Algorithms

Scribe: Chi Man Liu **Lecturer:** Shuchi Chawla
Topic: Inapproximability: Vertex Cover and Set Cover **Date:** 5/3/2007

In the previous lecture we introduced probabilistically checkable proofs (PCPs) and saw how they could be used to obtain a tight inapproximability result for MAX-3SAT. We also introduced Hastad's 3-bit PCP, a very useful tool for proving inapproximability results. In this lecture we apply Hastad's 3-bit PCP to show that approximating Vertex Cover within any constant factor less than 7/6 is NP-hard. We also show that Set Cover cannot be approximated within a factor of $\beta \cdot \log n$ for some constant β, unless NP $\subseteq$ DTIME($n^{\log \log n}$).

29.1 Vertex Cover

Last time we introduced Hastad's 3-bit PCP and out of it we obtained the following result.

Theorem 29.1.1 *For all constants $\epsilon > 0$ there is a reduction from* 3SAT *(or any* NP*-complete problem) to* MAX-3LIN *that maps satisfiable formulas to* MAX-3LIN *instances where a $1-\epsilon$ fraction of the constraints can be satisfied simultaneously, and maps unsatisfiable formulas to* MAX-3LIN *instances where no assignment satisfies more than a $1/2+\epsilon$ fraction of the constraints.*

We used Theorem 29.1.1 to show that it is NP-hard to approximate MAX-3SAT within any constant factor greater than 7/8. In the following, we show that approximating Vertex Cover within any constant factor less than 7/6 is NP-hard using a similar technique.

Theorem 29.1.2 *For all constants $\epsilon > 0$, Vertex Cover is* NP*-hard to approximate within a factor of $7/6 - \epsilon$.*

Proof: Our goal is to construct a gap-preserving reduction from MAX-3LIN to Vertex Cover. Given a MAX-3LIN instance with m constraints, we want to construct a graph G. For each linear constraint $x_p \oplus x_q \oplus x_r = b$, we add to G a clique with 4 vertices, where each vertex represents a setting of (x_p, x_q, x_r) satisfying the constraint. Then, for any two vertices, if their corresponding settings of variables conflict with each other, we add an edge between them. We now show that this polynomial-time reduction is gap-preserving. Suppose that the MAX-3LIN instance has an assignment satisfying at least a $1-\epsilon$ fraction of the constraints. For each satisfied constraint, pick the vertex in G that corresponds to its setting of variables in this assignment. These vertices form an independent set since they all come from different cliques, and all settings of variables agree. This independent set has size at least $(1-\epsilon)m$. There are $4m$ vertices in G, hence G has a vertex cover of size at most $(3+\epsilon)m$. Now suppose that no assignment satisfies more than a $1/2+\epsilon$ fraction of constraints. We claim that any independent set of G has at most $(1/2+\epsilon)m$ vertices. From this it follows that G has minimum vertex cover of size at least $(7/2-\epsilon)m$, and we have achieved a gap of $7/6-\epsilon$. It still remains to prove the claim. Let S be an independent set of G with k vertices. The vertices in S correspond to distinct linear constraints as they must lie in different cliques. Moreover, these vertices represent settings of variables that have no conflict. Thus, we can augment the settings to a total assignment that satisfies those k constraints. We finish the argument by setting $k = (1/2 + \epsilon)m$. ■

29.2 Set Cover

In this section, we show that Set Cover is unlikely to have a sublogarithmic approximation. Formally, we have the following theorem.

Theorem 29.2.1 *For some constant β, approximating Set Cover within a factor of $\beta \cdot \log n$ is* NP*-hard unless* $\text{NP} \subseteq \text{DTIME}(n^{\log\log n})$.

To prove this theorem, we consider the 2-Prover 1-Round system for 3SAT in the previous lecture. Given a 3CNF with m clauses, the verifier selects a clause and one of the variables in that clause uniformly at random, queries that clause and that variable to provers 1 and 2 respectively, and accepts if and only if the two responses agree on the value of the chosen variable. This system uses $O(\log m)$ random bits, has completeness 1 and soundness $1 - \epsilon'$ for some cosntant $\epsilon' > 0$. We take k parallel repetitions to reduce the soundness to α^k for some constant $\alpha < 1$, while increasing the amount of randomness to $O(k \log m)$. We will determine the value of k later. Now, a query to prover 1 is a k-tuple of clauses, while a query to prover 2 is a k-tuple of variables. The numbers of different queries to provers 1 and 2 are m^k and n^k respectively.

We introduce the Label Cover problem in the next section. It will soon be clear that finding optimal provers (ones that maximize the accepting probability) in 2P1R systems can be reduced to solving Label Cover instances.

29.2.1 Label Cover

Consider a bipartite graph $G = (Q_1, Q_2; E)$ with $|Q_1| = m^k$ and $|Q_2| = n^k$. Each vertex in Q_1 (resp. Q_2) represents a possible query to prover 1 (resp. prover 2). There is an edge between $u \in Q_1$ and $v \in Q_2$ if and only if the verifier can ask prover 1 query u and prover 2 query v simultaneously for some sequence of random bits. To each vertex $w \in Q_1 \cup Q_2$ we associate a *label set* L_w containing all correct answers to the query w. For every edge $e = (u, v)$, let R_e be the relation containing all pairs (a, b) ($a \in L_u$, $b \in L_v$) such that (a, b) is an accepting answer pair to query pair (u, v). Our goal is to pick an answer from each label set so that the probability of acceptance is maximized. In other words, we want to assign to each vertex w a label $a_w \in L_w$ such that the following quantity is maximized:

$$\frac{|\{e \mid e = (u, v) \in E, (a_u, a_v) \in R_e\}|}{|E|}.$$

We can generalize the above problem in the following way. Let L be a set of labels. For each vertex w, the label set L_w is a nonempty subset of L. For each edge e, the relation R_e is a subset of $L \times L$. This generalized problem is known as *Label Cover*.

The 2P1R system for 3SAT discussed above has completeness 1 and soundness α^k. Hence, G either has an assignment satisfying all edges, or has no assignment satisfying more than an α^k fraction of the edges. Note that $|L| = 2^{O(k)}$ and G has $O(n^k)$ vertices, where n is the number of literals in the formula. Moreover, we can assume that G satisfies certain properties that we will make use of in the reduction to Set Cover. This is formalized in the following lemma.

Lemma 29.2.2 *There exists a constant $\alpha < 1$, such that for every integer $k > 0$, there exists an infinite family of instances of the Label Cover problem indexed by n, such that the following holds:*

1. *The bipartite Label Cover graph is regular and has an equal number of vertices on both sides. Moreover, each side has $O(n^k)$ vertices.*
2. *For each edge $e = (u, v)$, the relation R_e is a projection from L_u to L_v. That is, for any $a \in L_u$, there exists a unique $b \in L_v$ satisfying $(a, b) \in R_e$.*
3. *Either there is an assignment satisfying all the edges (YES instances), or no assignment satisfies more than an α^k fraction of the edges (NO instances).*
4. *It is not possible to distinguish between YES and NO instances in polynomial time, unless* $\text{NP} \subseteq \text{DTIME}(n^k)$.

Proof: We form a Label Cover instance based on 3SAT(5) instead of a MAX-3SAT instance. (3SAT(5) is a special case of 3SAT where each variable in the formula occurs in exactly 5 clauses.) We use the fact that 3SAT(d) has gap instances just like 3SAT, for any $d \geq 5$. Following the above construction, we can reduce any 3SAT(5) instance with n variables to a Label Cover instance G satisfying property (3). For property (1), observe that $|Q_1| = (5n/3)^k$ and $|Q_2| = n^k$. Moreover, every vertex in Q_1 has degree 3^k while every vertex in Q_2 has degree 5^k. We create 3^k copies of Q_1 and 5^k copies of Q_2, and add edges appropriately. This gives us a (15^k)-regular graph with $(5n)^k$ vertices on each side which is essentially equivalent to G. Finally, property (2) follows from that any correct answer given by prover 1 induces a unique accepting answer for prover 2, namely the setting of variables that agrees with prover 1's answer.

Suppose that this problem can be solved in time poly(n^k). Then, by our reduction, 3SAT(5) can also be solved in time poly(n^k). Since 3SAT(5) is NP-hard, we have $\text{NP} \subseteq \text{DTIME}(n^k)$. ■

29.2.2 Reduction from Label Cover to Set Cover

Given a 3SAT(5) instance with n variables, we can reduce it to a regular Label Cover gap instance G with $O(n^k)$ vertices. We are going to show how to reduce regular Label Cover instances with edge relations being projections to Set Cover instances. Our strategy is to associate each edge $e = (u, v)$ in G with a Set Cover instance I_e such that picking an accepting pair $(a, b) \in L_u \times L_v$ would correspond to covering I_e with just a few sets, and vice versa. In other words, we would like to have a Set Cover instance in which the "coordinated" solutions have very few sets, whereas "uncoordinated" solutions use a much larger number of sets. For our purpose, we want this gap to be at least logarithmic.

Let U be the universe of elements and t be some integer parameter. For $i = 1, \dots, t$, we construct a set S_i by picking each element of U independently with probability 1/2. The collection of sets in the instance is $\mathcal{C} = \{S_1, \dots, S_t, \bar{S}_1, \dots, \bar{S}_t\}$, where $\bar{S}$ denotes the complement of S, i.e. $\bar{S} = U \backslash S$. We claim that the instance $I = (U, \mathcal{C})$ has the property we want. Let $\mathcal{S} \subseteq \mathcal{C}$ be a set cover. It is clear that if we pick both S_i and $\bar{S}_i$ for some i, we get a set cover with size 2. Otherwise, the (expected) size of $\mathcal{S}$ would be at least logarithmic in $|U|$: The expected number of elements covered by the first set in $\mathcal{S}$ is $|U|/2$. The second set covers $|U|/4$ of the uncovered elements on average since the sets are constructed by picking elements independently and uniformly at random. This rough argument suggests that covering all elements in U would require $\Theta(\log |U|)$ sets.

We can now use I as a building block to transform our Label Cover instance G to a Set Cover instance $\tilde{I}$. For each edge $e = (u, v)$, we construct an instance $I_e = (U_e, \mathcal{C}_e)$ equivalent to I with $|U_e| = n^k$. We take $t = |L_v| = 2^k$, so that each label $b \in L_v$ gets mapped to a unique set $S_{e,b} \in \mathcal{C}_e$. Then, $\mathcal{C}_e = \{S_{e,b} \mid b \in L_v\} \cup \{\bar{S}_{e,b} \mid b \in L_v\}$ has size 2^{k+1}. Next, we merge all the "edge" instances to get a large instance $\tilde{I} = (\tilde{U}, \tilde{\mathcal{C}})$ for the whole graph. Let $\tilde{U} = \bigcup_{e \in E} U_e$. For each $v \in Q_2$ and $b \in L_v$, let

$$\tilde{S}_{v,b} = \bigcup_{e \text{ incident on } v} S_{e,b}.$$

For each $u \in Q_1$ and $a \in L_u$, let

$$\tilde{S}_{u,a} = \bigcup_{e \text{ incident on } u} \bar{S}_{e,R_e(a)},$$

where $R_e(a)$ denotes the unique $b \in L_v$ satisfying $(a, b) \in R_e$. Let $\tilde{\mathcal{C}}$ be the collection of all these sets. We have $|\tilde{U}| = n^{O(k)}$ and $|\tilde{\mathcal{C}}| = n^{O(k)}$. Suppose that uncoordinated solutions to "edge" instance I_e have size at least $\ell = \Theta(\log |U_e|) = \Theta(k \cdot \log n)$. We will see that our reduction works if we pick $k = O(\log \log n)$. Then, intuitively, if we had a "good" approximation algorithm for Set Cover, we could use it to solve our Label Cover problem (and thus 3SAT(5)) exactly in $n^{O(\log \log n)}$ time. Hence, Theorem 29.2.1 follows. In the following, we give a formal proof for the correctness of our reduction.

Suppose that G has an assignment satisfying all edges. By picking the sets in $\tilde{\mathcal{C}}$ corresponding to this assignment, we are able to cover $\tilde{U}$ with only $|Q_1| + |Q_2| = O(n^k)$ sets. It remains to show that if G does not have any good solution, covering $\tilde{U}$ would require $\Omega(\ell n^k)$ sets.

Claim 29.2.3 *If every assignment to G satisfies at most an α^k fraction of edges, then every set cover of $\tilde{U}$ has size at least $\Omega(\ell n^k)$.*

Proof: We start with a minimum set cover $\mathcal{C}^*$. Let G_1 (resp. G_2) be the set of vertices in Q_1 (resp. Q_2) that have less than $\ell/2$ sets in $\mathcal{C}^*$. Let $B_1 = Q_1 \backslash G_1$ and $B_2 = Q_2 \backslash G_2$. Let $e = (u, v)$ be an edge such that $u \in G_1$ and $v \in G_2$. We pick an assignment to G in the following way. For every $u \in Q_1$, let A_u denote the set of labels a in L_u such that $\tilde{S}_{u,a} \in \mathcal{C}^*$. Pick an answer uniformly at random from A_u. Likewise, for every $v \in Q_2$, pick an answer uniformly at random from A_v. Consider some edge $e = (u, v)$ with $u \in G_1$ and $v \in G_2$. Then, $|A_u| + |A_v| < \ell$, so $\mathcal{C}^*$ induces a coordinated solution to I_e, and so A_u and A_v contain a pair of "corresponding" answers, i.e. there exist $a \in A_u$ such that $R_e(a) \in A_v$. Thus, the probability of this assignment satisfying e is at least $\frac{1}{\ell/2} \cdot \frac{1}{\ell/2} = 4/\ell^2$. The expected number of edges satisfied by the assignment is at least $\#e(G_1, G_2) \cdot 4/\ell^2$, where $\#e(G_1, G_2) = |E \cap (G_1 \times G_2)|$. Using the fact that the expectation never exceeds the maximum achievable value, we have $\#e(G_1, G_2) \cdot 4/\ell^2 \le \alpha^k \cdot |E|$. Therefore,

$$\frac{\#e(G_1, G_2)}{|E|} \quad \le \quad \alpha^k \cdot \frac{\ell^2}{4}.$$

Since $\ell^2 = O(k^2 \log^2 n)$ and $\alpha < 1$, it suffices to pick $k = O(\log \log n)$ to make $\alpha^k \ell^2/4 < 1/2$. Recall that G is a d-regular bipartite graph for some d depending on k. If $|B_1| < |Q_1|/4$ and $|B_2| < |Q_2|/4$, the number of edges with at least one endpoint in $B_1 \cup B_2$ would be less than $d(|Q_1| + |Q_2|)/4$, implying that $\#e(G_1, G_2)/|E| > 1/2$ since $|E| = d(|Q_1| + |Q_2|)/2$. Hence, either $|B_1| \geq |Q_1|/4$ or $|B_2| \geq |Q_2|/4$. Since $|Q_1|, |Q_2| > n^k$, the number of sets in $\mathcal{C}^*$ is at least $\frac{\ell}{2} \cdot \frac{n^k}{4} = \Theta(\ell n^k)$. ■

Let $N = O(n^{\log \log n})$ denote the size of the Set Cover instance. If Set Cover could be approximated to within a factor of $\beta \cdot \log N$ for every $\beta > 0$, we could easily tell whether our Label Cover instance has an assignment satisfying all edges in time poly(N). By Lemma 29.2.2, NP $\subseteq$ DTIME(N). This proves Theorem 29.2.1.