E0 249: Approximation Algorithms **Spring 2015**

Lecture 15, Session 1: Apr 17, 2015

Instructor: Arnab Bhattacharyya *Scribes: Cressida Hamlet & Marilyn George*

In previous lectures, we have started to see different approaches to proving the hardness of approximation. One of the approaches we listed was reduction from Probabilistically Checkable Proofs (PCPs). We briefly introduced PCPs in the previous class, and we will begin with a formal definition of a PCP.

1 Probabilistically Checkable Proofs

The verifiers that we have seen so far receive (1) an input x and (2) a proof Π, and output either Accept or Reject. The following properties of a verifier $V(x, \Pi)$ were defined:

- $\mathsf{Completeness}$: $\forall$ YES-instances, $\exists \Pi$ such that $V(x, \Pi)$ accepts.
- $\mathsf{Soundness}$: $\forall$ NO-instances, $\forall \Pi, V(x, \Pi)$ always rejects.

We now define a verifier of randomness r and query complexity q, which receives (1) an input x (2) proof Π (3) string R of length r. The verifier queries the string Π at q locations, and then accepts or rejects accordingly. The corresponding properties of the verifier $V(x, \Pi, R)$ are defined as follows:

- $\mathsf{Completeness, c}$: $\forall$ YES-instances, $\exists \Pi$ such that
$$\Pr_{R \in \{0,1\}^r}[V(x, \Pi, R) = \mathsf{Accept}] \geq \mathsf{c}$$
- $\mathsf{Soundness, s}$: $\forall$ NO-instances, $\forall \Pi$,
$$\Pr_{R \in \{0,1\}^r}[V(x, \Pi, R) = \mathsf{Accept}] \leq \mathsf{s}$$

Then the verifier we have seen earlier is equivalent to $V(x, \Pi, R)$ with $r = 0$.

Definition 1 $\mathsf{PCP}_{c,s}[r(n), q(n)]$ is the set of problems for which on inputs of length n, $\exists$ a poly(n) time verifier of randomness $r(n)$ and query complexity $q(n)$ $\diamondsuit$

From the above, we can see that

$$\mathsf{NP} = \mathsf{PCP}_{1,0}[0, \text{poly}(n)]$$

i.e. all problems in NP have a polynomial time deterministic verifier which can accept all YES-instances given the correct proof, and reject all NO-instances.

We will now state the **PCP Theorem**.

Theorem 1 *[PCP Theorem]* $\mathsf{NP} \subseteq \mathsf{PCP}_{1,\frac{1}{2}}[\mathcal{O}(\log n), \mathcal{O}(1)]$

This theorem is closely connected to the hardness of approximation, as we see in the following claim:

Claim: $\forall$ s, $\exists$ s$'$ such that [s$'$, 1]-Gap-MAX3SAT is NP-Hard $\Leftrightarrow$ NP $\subseteq \mathsf{PCP}_{1,s}[\mathcal{O}(\log n), 3]$

1. $[s', 1]$-Gap-MAX3SAT is NP-Hard $\Rightarrow$ NP $\subseteq \mathsf{PCP}_{1,s}[\mathcal{O}(\log n), 3]$.

   Let $A \in$ NP. By our LHS, we have that $\exists$ a reduction from A to [s$'$, 1]-Gap-MAX3SAT. Now, we construct a verifier V such that it randomly picks one clause and checks if the clause is satisfied.

   (a) If the original instance of A was a YES-instance, $\exists \Pi$ such that all clauses are satisfied, therefore any clause the verifier picks will be satisfied and hence $c = 1$.

   (b) If the original instance of A was a NO-instance, at most some s fraction of the clauses would be satisfied, and this would lead to soundness $\leq s$.

   Since the verifier has $\mathcal{O}(\log n)$ randomness and query complexity of 3, it can pick any clause from the input, and query the 3 variables of the clause to check satisfaction.

2. NP $\subseteq \mathsf{PCP}_{1,s}[\mathcal{O}(\log n), \mathcal{O}(1)] \Rightarrow [s', 1]$-Gap-MAX3SAT is NP-Hard, for some $s' < 1$.

   Suppose $A \in$ NP and x is an instance of A. Let $V^{x,R}(\Pi)$ be equal to $V(x, \Pi, R)$. If we fix x and R, then V acts only on a constant number of locations in Π, since query complexity is $\mathcal{O}(1)$. Then we consider $V^{x,R}(\Pi)$ over all R. Each $V^{x,R}(\Pi)$ is now a clause over all the possible values of R. By writing each $V^{x,R}(\Pi)$ as a 3-SAT formula, blow-up can be seen to be constant; since $V^{x,R}$ is a function on a constant number of inputs.

   Then regardless of the proof Π, for a NO-instance at most some s'-fraction of the clauses will be true ($s' < 1$). This is now a giant 3-SAT instance, with [s$'$, 1]-Gap.

For proving hardness of approximation, we need stronger results than Theorem 1. Hastad's theorem is such a result, where the verifier's tests are very restricted.

Theorem 2 *[Hastad's Theorem] NP $\subseteq PCP_{1-\delta,1/2+\varepsilon}[O(log\ n), 3]$, where each predicate of the verifier is checking whether $x + y + z = 0$ or $x + y + z = 1$, $\forall \varepsilon, \delta > 0$.*

Note: If the theorem holds for $\varepsilon = 0$ or $\delta = 0$, then it means that P=NP. In the $\delta = 0$ case, we just need to find if there is a satisfying assignment for the set of equations and we can use Gaussian elimination to find the existence of a solution, which takes only polynomial time. Therefore, all NP problems can be solved in polynomial time and so P=NP. For the $\varepsilon = 0$ case, we can find an assignment which satisfies atleast half of the equations, so we need only to check if there is an assignment where more than half the equations are satisfiable or not and that can be done in polynomial time, which also leads to the result P=NP.

From this theorem we immediately get the following.

Corollary 3 *$[1/2 + \varepsilon, 1 - \delta]$-Gap-3Lin is NP hard, $\forall \varepsilon, \delta > 0$.*

Theorem 4 *MAX3SAT is NP-hard to approximate to $7/8 + \varepsilon$, $\forall \varepsilon > 0$.*

Proof We can reduce $[1/2+\varepsilon, 1-\delta]$-Gap-3Lin to $[(7/8+\varepsilon)k, k]$-Gap-MAX3SAT for some $k > 0$. Since we know that $[1/2+\varepsilon, 1-\delta]$-Gap-3Lin is NP-hard, $[(7/8+\varepsilon)k, k]$-Gap-MAX3SAT is NP hard which implies that MAX3SAT is NP-hard to approximate to $7/8+\varepsilon$.

Now let's see how we can reduce a Gap-3Lin to a Gap-MAX3SAT problem. For any equation $x_i + x_j + x_k = 1$, there are 4 clauses (those shown below) corresponding to this equation.

$$\begin{gathered} x_i \vee x_j \vee x_k \\ x_i \vee \bar{x_j} \vee \bar{x_k} \\ \bar{x_i} \vee x_j \vee \bar{x_k} \\ \bar{x_i} \vee \bar{x_j} \vee x_k \end{gathered}$$

Similarily for any equation $x_i + x_j + x_k = 0$, the corresponding clauses are the following.

$$\begin{gathered} \bar{x_i} \vee \bar{x_j} \vee \bar{x_k} \\ \bar{x_i} \vee x_j \vee x_k \\ x_i \vee \bar{x_j} \vee x_k \\ x_i \vee x_j \vee \bar{x_k} \end{gathered}$$

In both cases if main equation is satisfied, then all 4 corresponding clauses will be satisfied, but if the main equation is not satisfied, then exactly 3 of the corresponding clauses will be satisfied. Which means if p out of m equations are satisfied, then in 3-Sat, $4p + 3(m-p) = p + 3m = (p/m + 3)m$ clauses are satisfied. So all instances of $[1/2+\varepsilon, 1-\delta]$-Gap-3Lin, where atleast $1-\delta$ of equations satisfiable, then $p/m = 1-\delta$, which means the MAX3SAT instance we created has atleast $(1-\delta+3)m = (4-\delta)m$ satisfiable clauses. Similarly, for instances in Gap-3Lin where atmost $(1/2+\varepsilon)$ of equations are satisfiable, we create a correponding instance in MAX3SAT where atmost $(1/2+\varepsilon+3)m = (7/2+\varepsilon)m$ clauses are satisfiable. So we got a reduction to $[(7/2+\varepsilon)m, (4-\delta)m]$-Gap-MAX3SAT, which we can write as $[(7/8+\varepsilon)k, k]$-Gap-MAX3SAT. ■