Limits of Approximation Algorithms 25 Feb, 2010 (TIFR)

Lec. 5: 2-query PCPs and BLR-Linearity Testing

Lecturer: Prahladh Harsha *Scribe: Girish Varma*

In today's lecture, we will see PCPs that require only 2 queries (however over a large alphabet). This will lead us to the gap problem Label-Cover, which is the starting point of almost all tight hardness results.

In the second half od today's lecture, we will see some of the first local testing results, linearity testing, which eventually then led to the PCP Theorem.

5.1 2-query PCPs

In the previous lectures, we saw the following form of the PCP Theorem.

Theorem 5.1.1 (PCP Theorem)**.**

$$\exists \text{ constant } Q,\ NP \subseteq PCP_{1,1/2}[O(\log n), Q]$$

or equivalently, $\exists \alpha > 1$ *such that it is NP-Hard to approximate MAX3SAT to factor* α*.*

Today we will ask the question how small can the query complexity Q be made? For this, we first generalize our notion of PCPs to include proofs over non-binary alphabets.

Definition 5.1.2. *$PCP^{\Sigma}_{c,s}[r, Q]$ is the class of languages that have restricted verifiers that uses r random bits, Q queries to the proof π, which is a string over Σ, with*

Completeness $\forall x \in L,\ \exists \pi,\ \Pr[V^{\pi}(x) \text{ accepts }] \geq c.$

Soundness *ie.* $\forall x \notin L,\ \forall \pi,\ \Pr[v^{\pi}(x) \text{ accepts }] \leq s.$

When $\Sigma = \{0, 1\}$, we will not mention it.

Remark 5.1.3.

- *If the proof is over a binary alphabet and we have perfect completeness (i.e., $c = 1$) then $Q \geq 3$ unless $P \neq NP$. Suppose we had a 2-query PCP over binary alphabet for a language L with $c = 1$, then $L \leq_p$ 2SAT: for each input x and random coin toss R, the action of the verifier is a 2-ary Boolean function that can be encoded as a 2CNF formula φ_R. Let $\varphi = \bigwedge_R \varphi_R$. Clearly $x \in L$ iff $\varphi \in$ 2SAT. Since 2SAT $\in P$, we have $PCP^{\{0,1\}}_{1,s}[O(\log n), 2] \subseteq P$.*
- *$PCP^{\Sigma}_{c,s}[r, q] \subseteq PCP_{c,s}[r, q \log |\Sigma|]$. This is done by encoding the the alphabet Σ in the binary alphabet $\{0, 1\}$ using $\log |\Sigma|$ bits for each symbol in Σ. The query complexity thus increases to $q \cdot \log |\Sigma|$.*

We will now show that we can reduce the query complexity to 2 if we allow for a large alphabet.

Theorem 5.1.4. $PCP^{\Sigma}_{c,1-\varepsilon}[r, q] \subseteq PCP^{\Sigma^q}_{c,1-\varepsilon/q}[r + \log q, 2]$.

Proof. Suppose $L \in PCP^{\Sigma}_{c,s}[r,q]$, then L has a $(r, q, poly(n), poly(n))$-restricted verifier V. On input x and a proof $\pi : [m] \to \Sigma$, V

1. randomly tosses some coins R
2. generates queries $i_1, i_2, \ldots i_q$
3. reads $\pi_{i_1}, \pi_{i_2}, \ldots \pi_{i_q}$
4. Accepts if the predicate $P_{x,R}(\pi_{i_1}, \pi_{i_2}, \ldots, \pi_{i_q})$ is satisified.

Now consider the following proof system with a verifier V'. The proof will consist of a pair $\pi_1 : [m]^q \to \Sigma^q$ and $\pi_2 : [m] \to \Sigma$. On input x and the proof π_1, π_2, V'

1. randomly tosses some coins R'. We will view R' as (R, q)
2. generates queries $i_1, i_2, \ldots i_q$
3. randomly picks $j \in [q]$
4. reads $\pi_1(i_1, i_2, \ldots, i_q)$ and $\pi_2(i_j)$
5. Computes the predicate

   $$P_{x,R'}(\pi_1(i_1, i_2, \ldots, i_q)) = P_{x,R}(\pi_1(i_1, i_2, \ldots, i_q)_1, \pi_1(i_1, i_2, \ldots, i_q)_2, \ldots, \pi_1(i_1, i_2, \ldots, i_q)_q).$$

6. checks if $\pi_2(i_j)$ is same as jth symbol of $\pi_1(i_1, i_2, \ldots, i_q)$
7. Accepts if the predicate in step 5 and the check in 6 are true.

If $x \in L$, then there is a proof π that V accepts with probability $\geq c$. We can easily construct a proof $\pi_1(i_1, i_2, \ldots, i_q) = \pi(i_1)\pi(i_2)\ldots\pi(i_q)$ and $\pi_2 = \pi$ that V' will accept with probability $\geq c$.

If $x \notin L$, then for any proof $\pi_2 = \pi$, V rejects with probability at least ε. In other words, at least ε-fraction of the predicates $P_{x,R}(\pi_{i_1}, \ldots, \pi_{i_q})$ are rejected. But π_1 need not be an exact expansion of π_2(ie $\pi_1(i_1, i_2, \ldots, i_q) \neq \pi(i_1)\pi(i_2)\ldots\pi(i_q)$). We must thus have that either $\pi_1(i_1, i_2, \ldots, i_q) = \pi(i_1)\pi(i_2)\ldots\pi(i_q)$ in which case step 5 rejects or at least one of the symbols $\pi_1(i_1, \ldots, i_q)_j$ must disargree with the corresponding symbol $\pi_2(i_j)$ in which case step 6 rejects with probability at least $1/q$. Hence, the overall test rejects with probability at least ε/q, completeing the soundness analysis.

□

Corollary 5.1.5. $\exists$ *constant* Q, $NP \subseteq PCP_{1,1/2}[O(\log n), Q] \subseteq PCP^{\{0,1\}^Q}_{1,1-\frac{1}{2Q}}[O(\log n), 2]$.

Suppose we want to decrease the error probability of soundness s, then we repeat the PCP, say some t times and reject if at least one of them rejects. This gives us that $\text{PCP}^{\Sigma}_{1,s}[O(\log n), 2] \subseteq \text{PCP}^{\Sigma}_{1,s^t}[O(t \log n), 2t]$. But this increases the number of queries. What if we ask all the t queries together? In other words, we still ask only 2 queries but club all the t repetitions together. On the face of it, it is not at all clear that the error drops if one repeats in such a parallel fashion. Surprisingly, Feige and KIlian [FK00] and then Raz [Raz98] showed that the error does drop exponentially in t even in this case.

Theorem 5.1.6 (Parallel Repitition). $\forall s \in (0,1),\ \exists c_s \in (0,s),\ PCP^{\Sigma}_{1,s}[r,2] \subseteq PCP^{\Sigma^t}_{1,c_s^t}[rt,2]$.

Corollary 5.1.7. $\forall \varepsilon > 0,\ \exists$ *an alphabet* Σ *such that*

$$NP \subseteq PCP^{\Sigma}_{1,\varepsilon}[O(\log n \cdot \log(1/\varepsilon)), 2]$$

where such that $|\Sigma| = \text{poly}(1/\varepsilon)$.

5.1.1 LABEL-COVER

We saw that there are PCP proof systems for languages in NP, in which the verifier makes only 2 queries. Furthermore, observe that the check made by the verifier is in the form of a projection from the first answer to the second. It will be more convenient to abstract these 2-query PCPs in terms of a graph problem, which we call Label-Cover.

Definition 5.1.8 (Label-Cover). *An instance I of the* Label-Cover *problem is specified by a quadruple $(G, \Sigma_1, \Sigma_2, F)$ where $G = (L, R, E)$ is a bipartite graph, Σ_1 and Σ_2 are two finite sized alphabets and $\Pi = \{\pi_e : \Sigma_1 \to \Sigma_2 | e \in E\}$, is a set of functions (also called projections), one for each edge $(u,v) \in E$.*

A labeling $A : L \to \Sigma_1, B : R \to \Sigma_2$, is said to satisfy an edge (u,v) iff $\pi_{(u,v)}(A(u)) = B(v)$. The value *of an instance is the maximal fraction of edges satisfied by any such labeling.*

For any $\delta \in (0,1)$, the gap problem GAP_ε-LC *is the promise problem of deciding if a given instance has value 1 or at most ε. More precisely, the* YES *and* NO *of* GAP_ε-LC *are given as follows.*

$$\begin{aligned} \textsf{YES} &= \left\{ I : \exists\, (A : L \to \Sigma_1,\ B : R \to \Sigma_2) \text{ such that } \forall (u,v) \in E,\ \pi_{(u,v)}(A(u)) = B(v) \right\} \\ \textsf{NO} &= \left\{ I : \forall\, (A : L \to \Sigma_1,\ B : R \to \Sigma_2),\ \left| \{(u,v) \in E : \pi_{(u,v)}(A(u)) =, B(v)\} \right| \leq \varepsilon |E| \right\} \end{aligned}$$

Thus, an equivalent formulation of Corollary 5.1.7 is the following.

Corollary 5.1.9. $\forall \varepsilon > 0$, *there exist alphabets Σ_1, Σ_2 with $|\Sigma_1|, |\Sigma_2| = \text{poly}(1/\varepsilon)$ such that* GAP_ε-LC *is NP-hard.*

5.2 Linearity Testing

We will now see a randomized test for checking whether a function between two groups is linear (ie a homomorphism).

Definition 5.2.1 (Linear function). *Gven two Abelian groups G, H, the function $f : G \to H$ is said to be linear iff*

$$\forall x, y \in G,\ f(x+y) = f(x) + f(y).$$

(Observe that the first $+$ is performed according to group G while the second is performed according to H.

A naive test for checking for linearity is to randomly pick $x, y \in G$ and check if $f(x+y) = f(x)+f(y)$. Blum, Luby and Rubinfield [BLR93] showed that this check is actually a "good" one.

$\mathsf{BLR\text{-}Test}^f$: " 1. Choose $y, z \in_R G$ independently
2. Query $f(y), f(z),$ and $f(y+z)$
3. Accept if $f(y) + f(z) = f(y+z)$. "

We will prove that this test is "good". Before stating the result we need some definitions.

Definition 5.2.2 (Local consistency)**.** *Local consistency of $f \in H^G$ denoted by $\varepsilon(f)$ is defined as*

$$\varepsilon(f) = \Pr_{x,y}[f(x) + f(y) \neq f(x+y)]$$

It is clear that, if f is linear then the naive check is true with probability 1 (ie. $\varepsilon(f) = 0$).

Definition 5.2.3 (Hamming distance)**.** *For $f, g \in H^G$, the* Hamming distance *$\delta(f, g)$ is defined as*

$$\delta(f, g) = \Pr_{x \in G}[f(x) \neq g(x)]$$

For $f \in H^G$ and $S \subseteq H^G$,

$$\delta(f, S) = \min_{g \in S} \delta(f, g)$$

Definition 5.2.4 (Global Consistency)**.** *Let $L \subset H^G$ be the set of all linear functions (homomorphisms) from G to H. Then the* global consistency *of $f \in H^G$ denoted by $\delta(f)$ is defined as*

$$\delta(f) = \delta(f, L)$$

It is clear that $\delta(f) = 0 \Rightarrow \varepsilon(f) = 0$. To show that the test is "good" is equivalent to proving a result of the following type: $\delta(f)$ is "large" implies $\varepsilon(f)$ is "large". We will first show that a general result of this nature is impossible if $\varepsilon(f)$ is not too small.

Consider the function $f : \mathbb{Z}/3n\mathbb{Z} \to \mathbb{Z}/3n\mathbb{Z}$ defined as follows:

$$f(x) = \begin{cases} 0 & \text{if } x \equiv 0 \pmod 3 \\ 1 & \text{if } x \equiv 1 \pmod 3 \\ 3n-1 & \text{if } x \equiv -1 \pmod 3 \end{cases}$$

It can be shown that though for this f, $\varepsilon(f) = 2/9$, $\delta(f) = 2/3$. Thus, even though $\delta(f)$ is very large, $\varepsilon(f)$ is not all that large. This counterexample was given by Coppersmith. We will now give an analysis of the test (also due to Coppersmith) which shows that this is basically the worst example. and also showed the following

Claim 5.2.5. *Suppose $\varepsilon(f) < 2/9$ then $\delta(f) \leq 2\varepsilon(f)$.*

Proof. Let $\varphi : G \to H$ be defined as

$$\varphi(x) = \text{plurality}_y\{f(x+y) - f(y)\},$$

with ties being broken arbibtrarily. We will show that φ has the following properties, which clearly implies the claim.

1. $\delta(f, \varphi) \leq 2\varepsilon(f)$
2. $\forall x, \Pr_y[\varphi(x) = f(x+y) - f(y)] \geq 2/3$. Thus, even though $\varphi(x)$ was defined as the plurality, it is actually a 2/3-majority.
3. φ is linear

1. Proof of "$\delta(f, \varphi) \leq 2\varepsilon(f)$"

   Let $\text{BAD} = \{x \in G : \Pr_y[f(x) \neq f(x+y) - f(y)] \geq 1/2\}$. If $x \notin \text{BAD}$, then $f(x) = \varphi(x)$. Hence, $\delta(f, \varphi) \leq |\text{BAD}|/|G|$. Now

$$\begin{aligned} \varepsilon(f) &= \Pr_{x,y}[f(x) \neq f(x+y) - f(y)] \\ &\geq \Pr[x \in \text{BAD}] \cdot \Pr[f(x) \neq f(x+y) - f(y) | x \in \text{BAD}] \\ &\geq \frac{|\text{BAD}|}{2|G|} \end{aligned}$$

2. Proof of "$\forall x,\ \Pr_y[\varphi(x) = f(x+y) - f(y)] \geq 2/3$"

   Fix any x, consider the following the collision probability

$$\begin{aligned} & \Pr_{y_1,y_2}[f(x+y_1) - f(y_1) = f(x+y_2) - f(y_2)] \\ = & \Pr_{y_1,y_2}[f(x+y_1) + f(y_2) = f(x+y_2) + f(y_1)] \\ \geq & \Pr_{y_1,y_2}[f(x+y_1) + f(y_2) = f(x+y_1+y_2) = f(x+y_2) + f(y_1)] \\ \geq & 1 - 2\varepsilon(f) > 5/9 \end{aligned}$$

   For $h \in H$, let $p_h = \Pr_y[f(x+y) - f(y) = h]$. Clearly, $p_{\max} = \max p_h = \Pr_y[\varphi(x) = f(x+y) - f(y)]$. Since the p_h's are a probability distribution, we have $\sum_{h \in H} p_h = 1$. From the above argument wrt. to the collision probability we have $\sum_{h \in H} p_h^2 > 5/9$. Then the following is true

$$\begin{aligned} p_{\max} = p_{\max} \cdot \sum p_h &\geq \sum_{h \in H} p_h^2 > 5/9 \\ p_{\max}^2 + (1 - p_{\max})^2 &\geq \sum_{h \in H} p_h^2 > 5/9 \\ 2p_{\max}^2 - 2p_{\max} + 4/9 &> 0 \\ p_{max} &> 2/3 \end{aligned}$$

   So for at least 2/3 fraction of the y's $f(x+y) - f(y)$ is the same and hence equal to $\varphi(x)$.

3. φ is linear.
   From 2, we have that

$$\begin{aligned} \varphi(x) &= f(y) - f(y-x) \text{ for all but } < 1/3 \text{ fraction of } y\text{'s} \\ \varphi(z) &= f(y+z) - f(y) \text{ for all but } < 1/3 \text{ fraction of } y\text{'s} \\ \varphi(x+z) &= f(y+z) - f(y-x) \text{ for all but } < 1/3 \text{ fraction of } y\text{'s} \end{aligned}$$

   Therefore $\exists y$ such that all the above 3 equations are satisfied. Hence

$$\forall x, z \in H,\ \varphi(x) + \varphi(z) = \varphi(x+z)$$

□

References

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. *Self-testing/correcting with applications to numerical problems.* J. Computer and System Sciences, 47(3):549–595, December 1993. (Preliminary Version in *22nd STOC*, 1990). doi:10.1016/0022-0000(93)90044-W.

[FK00] Uriel Feige and Joe Kilian. *Two-prover protocols – low error at affordable rates.* SIAM J. Computing, 30(1):324–346, 2000. (Preliminary Version in 26*th STOC*, 1994). doi:10.1137/S0097539797325375.

[Raz98] Ran Raz. *A parallel repetition theorem.* SIAM J. Computing, 27(3):763–803, June 1998. (Preliminary Version in *27th STOC*, 1995). doi:10.1137/S0097539795280895.