PCPs from Linear PCPs

Instructor: Alessandro Chiesa & Igor Shinkar *Scribe: Aditya Mishra*

1 Introduction

In this lecture, we formally introduce Linear PCPs (LPCPs), and then show how one can compile any LPCP into a PCP. This will complete the proof that NP $\subseteq$ PCP[poly(n), $O(1)$] from last lecture.

2 Linear PCPs

We repeat the definition of a PCP in order to compare it with that of a LPCP.

Definition 1 *A PCP for a language L is a probabilistic polynomial time verifier V such that:*

1. Completeness. $\forall x \in L, \exists \pi \in \{0,1\}^l$ *such that* $\Pr[V^{\pi}(x)] = 1 \geq c$
2. Soundness. $\forall x \notin L, \forall \pi \in \{0,1\}^l$, *it holds that* $\Pr[V^{\pi}(x)] = 1 \leq s$

We say that $L \in \mathbf{PCP}_{c,s}[r,q,l]$ if the above holds with V tossing r random coins and making q queries.

We now turn to LPCPs, which are the same as PCPs except that that the verifier has oracle access to a linear function rather than a string.

Definition 2 *A LPCP for a language L is a probabilistic polynomial time verifier V such that:*

1. Completeness. $\forall x \in L, \exists \lambda \in \{0,1\}^l$ *such that* $\Pr[V^{\langle \lambda, \cdot \rangle}(x) = 1] \geq c$
2. Soundness. $\forall x \notin L, \forall \lambda \in \{0,1\}^l$, *it holds that* $\Pr[V^{\langle \lambda, \cdot \rangle}(x) = 1] \leq s$

We say that $L \in \mathbf{LPCP}_{c,s}[r,q,l]$ if the above holds with V tossing r random coins and making q queries.

Note here that, while $\langle \lambda, \cdot \rangle$ is a linear function defined via l bits, the evaluation table of $\langle \lambda, \cdot \rangle$ consists of 2^l bits.

3 Compiling a Linear PCP into a PCP

We describe how any Linear PCP can be compiled into a (standard) PCP.

Idea 3 *Let* $\pi\colon [2^l] \to \{0,1\}$ *be an evaluation table of* $\langle \lambda, \cdot \rangle$. *Let* $V_{\boldsymbol{PCP}} = V_{\boldsymbol{LPCP}}$.

This seems like a good idea at first. However, the prover may write $\tilde{\pi}$ that is not the evaluation table of any linear function. We clearly have no way to check if $\tilde{\pi}$ is the evaluation of a linear function in less than 2^l queries, as there could always be a mistake at the location that we did not query. That said, as we shall see, it will suffice to ensure that $\tilde{\pi}$ is *close* to the evaluation of a linear function, and this can be done with few queries.

Definition 4 *We say that a function* $f\colon \{0,1\}^n \to \{0,1\}$ *is* δ*-far from* LIN *if for all linear functions* $p \in \text{LIN}$, $\Delta(f,p) \geq \delta$. *Likewise, we say that a function* $f\colon \{0,1\}^n \to \{0,1\}$ *is* δ*-close from* LIN *if there exists a linear function* p *such that* $\Delta(f,p) \leq \delta$.

Theorem 5 *There exists* $O(1)$*-query verifier* V_{LIN} *such that:*

1. $\forall \pi \in \text{LIN}, Pr[V_{\text{LIN}}^{\pi} = 1] = 1$
2. $\forall \pi$ *such that* $\Delta(\pi, \text{LIN}) > \frac{1}{10}, Pr[V_{\text{LIN}}^{\pi} = 1] \leq \frac{1}{2}$

We will hold off the proof for Theorem 5 until Section 4.

Now we can define $V_{\mathbf{PCP}}^{\pi}$ as follows:

1. Run V_{LIN}^{π}. If the function is not linear, reject.
2. Run $V_{\mathbf{LPCP}}^{\langle \tilde{\lambda}, \cdot \rangle}$, where $\langle \tilde{\lambda}, \cdot \rangle$ is π treated as a linear function.

The proof of completeness is trivial. We now prove soundness. Suppose that $x \in L$ and $\tilde{\lambda}$ is a function from $[2^l] \to \{0,1\}$. There are two cases. Suppose that $\tilde{\lambda}$ is $\frac{1}{10}$ far from LIN. This implies that V_{LIN} accepts $\tilde{\lambda}$ as linear with probability at most $\frac{1}{2}$, and $V_{\mathbf{LPCP}}$ by definition accepts with probability at most s. The second case is when $\tilde{\lambda}$ is $\frac{1}{10}$-close from LIN. Let λ be the closest linear function to $\tilde{\lambda}$. Assuming that the distribution of the queries is uniformly random, we see that

$$\Pr[V_{\mathbf{PCP}}^{\tilde{\lambda}}\text{accepts}] \leq \Pr[V_{\mathbf{LPCP}}^{\langle \lambda, \cdot \rangle}\text{accepts}] + \Pr[\exists\text{a query that is noise}]$$
$$\leq s + q \cdot \frac{1}{10}$$

Of course in most cases, the distribution of the queries is not uniformly random. We can use self-correction in order to bring down the upper-bound shown in the last expression, and to address the issue of the bias of the queries. This is explained below.

Idea 6 *For all* $a \in \{0,1\}^l$, *pick random* $r \in \{0,1\}^l$ *and return* $\pi(r) + \pi(r+a)$. *Using the union bound, we see that*

$$\Pr[\langle \lambda, a \rangle \neq \pi(r) + \pi(r+a)] \leq \frac{2}{10}$$

Using Chernoff bounds, we see that doing this process $O(\log q)$ *times will result in an error at most* $O(\frac{1}{q})$. *Of course, we can bring down the error further as we wish by having more queries.*

We have shown that indeed Theorem 1 holds with:

1. $c' = c$
2. $s' = \max\{\frac{1}{2}, s + \epsilon\}$, where ϵ is the error that occurred from the $\log q$ queries
3. $r' = r + \log(q) \cdot l$
4. $q' = q \cdot \log(q)$
5. $l' = 2^l$

4 A Linearity Test

The compiler from LPCP to PCP that we have described assumed the existence of a *linearity test*, as stated in Theorem 5. We now prove this theorem by presenting and analyzing the linearity test of Blum, Luby, and Rubinfeld [BLR93]; we follow lecture notes by Moshkovitz [Mos10].

4.1 Preliminaries

Before we introduce the actual test, we first go over some definitions.

Definition 7 *A function $f : \{0,1\}^n \to \{0,1\}$ is linear if for all $x, y \in \{0,1\}^n$, $f(x+y) = f(x)+f(y)$.*

4.2 The Actual Test

Suppose we are given a (potentially linear) function $f : \{0,1\}^n \to \{0,1\}$. Choose points $x, y \in \{0,1\}^n$ independently and uniformly at random, and test if $f(x) + f(y) = f(x + y)$ over $\mathbb{F}_2$. It is easy to see that this is a 3-query verifier. The proof of completeness is trivial, since if f is linear, then by definition of linearity, this test will pass with probability 1. The soundness theorem is as follows:

Theorem 8 $\Pr[\textit{BLR test rejects } f] \geq \textit{min}\left(\frac{2}{9}, \frac{\Delta(f,\text{LIN})}{2}\right)$

The subsequent section gives a proof of soundness for the BLR test.

4.3 Proof of Soundness

We use the idea of majority correction. If a function f is linear in a binary field, we have that $f(x) = f(y) + f(x + y)$. We can think of each of the 2^n possible values of y as a vote on the value of $f(x)$. Since $f(x)$ is equal to either 0 or 1, we see that either 0 or 1 received the majority of votes from the y values. More formally, we define g_f (which is dependent on f) as follows:

$$g_f(x) = \begin{cases} 1 & \text{if } \Pr_y[f(y) + f(x-y) = 1] \geq \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

We also define $P_x = \Pr_y[g_f(x) = f(y) + f(x - y)]$. Note that by definition of g_f, $P_x \geq \frac{1}{2}$. In order to prove soundness, we first prove some claims.

Claim 9 $\Pr[\textit{BLR rejects f}] \geq \frac{1}{2} \cdot \Delta(f, g)$

Proof: We have that:

$\Pr[\text{rejection}] = \Pr[g(x) \neq f(x)] \cdot \Pr[\text{rejection} | g(x) \neq f(x)] + \Pr[g(x) = f(x)] \cdot \Pr[\text{rejection} | g(x) = f(x)]$

Since we are interested in a lower bound, we ignore the second term. Note that $\Pr[g(x) \neq f(x)] = \Delta(f, g)$ by definition. We see that if $g(x) \neq f(x)$, then $f(x) = (y) + f(x - y)$ for $1 - P_x \leq \frac{1}{2}$ of the possible values for y. Since we are in $\mathbb{F}_2$, addition and subtraction are the same and so the equation $f(x) = f(y) + f(x - y)$ is the same as the BLR test, $f(x + y) = f(x) + f(y)$. □

Claim 10 *If* $\Pr[\textit{BLR rejects f}] < \frac{2}{9}$, *then for all* x *we have* $P_x > \frac{2}{3}$.

Proof: Fix x. We define

$$A_x = \Pr_{y,z}[f(y) + f(x + y) = f(z) + f(x + z)]$$

We can compute A in two different ways. We see that

$$A_x = \Pr_{y,z}[f(y) + f(x + y) = g(x) \wedge f(z) + f(x + z) = g(x)]$$

$$+ \Pr_{y,z}[f(y) + f(x + y) \neq g(x) \wedge f(z) + f(x + z) \neq g(x)]$$

$$= P_x^2 + (1 - P_x)^2$$

We can also use the BLR rejection probability to bound A_x. Since we are working over a binary field, we can rewrite the equation $f(y) + f(x + y) = f(z) + f(x + z)$ as $f(y) + f(z) = f(x + y) + f(x + z)$. We see that by linearity, $\Pr[f(y) + f(z) = f(y + z)] = 1 - \Pr[\text{BLR rejects f}] > \frac{7}{9}$. As y and z are independent and uniformly sampled, we can apply the same reasoning to the case of $x + y$ and $x + z$. Thus we can say that $f(x + y) + f(y + z) = f((x + y) + (x + z)) = f(y + z)$ with probability greater than $\frac{7}{9}$. Thus the probability of both these events happening (which is A_x) is greater than $\frac{5}{9}$. Solving the quadratic:

$$P_x^2 + (1 - P_x)^2 > \frac{5}{9}$$

gives $[0, \frac{1}{3}) \cup (\frac{2}{3}, 1]$ as solutions. As $P_x \geq \frac{1}{2}$, we see that $P_x > \frac{2}{3}$. □

Claim 11 *If* $\Pr[\textit{BLR rejects f}] < \frac{2}{9}$, *then* g_f *is linear.*

Proof: Using the previous claim, we see that $P_x > \frac{2}{3}$. Fix x and y and choose z uniformly and random. Then $g(x) = f(z) + f(x + z)$ with probability larger than $\frac{2}{3}$. Using the same argument, we see that $\Pr[g(y) = f(z) + f(y + z)] > \frac{2}{3}$ and $\Pr[g(x + y) = f(z) + f(x + z + y)] > \frac{2}{3}$. Substituting $(x + z)$ in place of z, we have that $\Pr[g_f(x + y) = f(z + x) + f(z + y)] > \frac{2}{3}$. Thus, there exists a z_0 such that:

$$g_f(x) = f(z_0) + f(x + z_0)$$

$$g_f(y) = f(z_0) + f(y + z_0)$$

$$g_f(x + y) = f(x + y + z_0)$$

all hold. This shows that

$$g_f(x) + g_f(y) = g_f(x + y)$$

So we see that g_f is linear. □

Using the previous claims we now can prove soundness for the BLR test. There are two cases: either $\Pr[\text{rejection}] \geq \frac{2}{9}$, or g is linear and so

$$\Pr[\text{rejection}] \geq \frac{1}{2} \cdot \Delta(f, g) \geq \frac{1}{2}\Delta(f, \text{LIN})$$

This is exactly what the soundness theorem claims.

References

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld, *Self-testing/correcting with applications to numerical problems*, Journal of Computer and System Sciences **47** (1993), no. 3, 549–595.

[Mos10] Dana Moshkovitz, *Pcp and hardness of approximation, lecture 5: Linearity testing*, `http://www.cs.utexas.edu/~danama/courses/approximability/linearity-testing.pdf`, 2010.