CS286.2 Lecture 3: The linearity test and low-degree extensions

Scribe: Sid Barman

Continuing our discussion from last lecture, we will complete the proof of the following simpler version of PCP theorem:

Theorem 1. NP $\subseteq$ $\mathrm{PCP}_{1,1/2}(r = O(poly(n)), q = O(1))$.

As mentioned in the last lecture, the proof of this theorem constructs a verifier for the NP-complete problem "Quadratic Equations" (**QUADEQ**). Recall that an instance of **QUADEQ**, say φ, is given by m constraints C_j over n boolean variables x_i of the form:

$$C_j : \alpha^{(j)} \cdot x + \beta^{(j)} \cdot (x \otimes x) \equiv \gamma^{(j)} \mod 2.$$

Here, boolean variables $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$, and for every constraint j we are given the following coefficients $\alpha^{(j)} \in \{0,1\}^n$, $\beta^{(j)} \in \{0,1\}^{n^2}$ along with $\gamma^{(j)} \in \{0,1\}$.

In the last lecture we showed that given a proof $\Pi = (\Pi^1, \Pi^2)$—with $\Pi^1 \in \{0,1\}^{2^n}$ and $\Pi^2 \in \{0,1\}^{2^{n^2}}$—that satisfies

$$(\Pi^1)_\alpha = \alpha \cdot x, \text{ and}$$
$$(\Pi^2)_\beta = \beta \cdot (x \otimes x),$$

for some $x \in \{0,1\}^n$, we can check the proof with completeness 1 and soundness $1/2$.

The linearity test

Today, we present the linearity test that verifies that a proof Π is of the required form, i.e., $(\Pi^1)_\alpha = \alpha \cdot x$ and $(\Pi^2)_\beta = \beta \cdot (x \otimes x)$ for some x. Interestingly, we will accomplish this using only four queries.

The linearity test is based on the following theorem by Blum, Luby, and Rubinfeld [BLR93].

Theorem 2 (BLR)**.** *Suppose a function $f : \{\pm 1\}^n \to \{\pm 1\}$ satisfies*

$$\Pr_{\alpha, \alpha' \in_U \{\pm 1\}} \left[f(\alpha) f(\alpha') = f(\alpha\alpha') \right] = 1 - \varepsilon,$$

where the product $\alpha\alpha'$ is taken componentwise and $\in_U$ means that α, α' are chosen uniformly at random. Then there exists a set $S \subseteq [n]$ such that $f(\alpha) = \prod_{i \in S} \alpha_i$ for at least a fraction $1 - \varepsilon$ of all α.

The connection between the theorem and the linearity test is obtained by setting $f(\alpha) = (-1)^{\Pi}_{\alpha}$. Now, if f satisfies $f(\alpha) = \prod_{i \in S} \alpha_i$ then proof Π satisfies the required linear form, i.e., $\Pi_\alpha = \sum_{i \in S} \alpha_i = c_S \cdot \alpha$, where c_s is the indicator vector for set S. To prove the BLR theorem we will need a little bit of Fourier analysis over $\mathbb{F}_2^n$.

Basics of Fourier analysis over $\mathbb{F}_2^n$

In this section we present some basic definitions and results from Fourier analysis over $\mathbb{F} = (\{\pm 1\}, \times)$. These results are used in the next section to prove Theorem 2.

Note that the set $V = \{f : \{\pm 1\}^n \to \mathbb{R}\}$ is a vector space. The canonical basis for this vector space consists of the delta functions δ_x. Specifically, for all $x \in \{\pm 1\}^n$ write

$$\begin{aligned} \delta_x : y \mapsto 1 \quad & \text{if } y = x \\ 0 \quad & \text{otherwise.} \end{aligned}$$

We have a natural inner product in this vector space that is defined as follows:

$$\langle f, g \rangle := \frac{1}{2^n} \sum_{x \in \{\pm 1\}^n} f(x) g(x).$$

In addition to the canonical basis, for this vector space of functions we have a *Fourier* basis that consists of the following set of functions $\{\chi_S : x \mapsto \prod_{i \in S} x_i \mid S \subseteq [n]\}$. Note that this basis is orthonormal with respect to the inner product $\langle , \rangle$:

$$\begin{aligned} \langle \chi_S, \chi_T \rangle &= \frac{1}{2^n} \sum_x \chi_S(x) \chi_T(x) \\ &= \frac{1}{2^n} \sum_x \prod_{i \in S} x_i \prod_{i \in T} x_i \\ &= \frac{1}{2^n} \sum_x \chi_{S \Delta T}(x), \qquad (\text{since } x_i \in \{\pm 1\}, x_i^2 = 1). \end{aligned}$$

where $S \Delta T$ denotes the symmetric difference between the sets S and T. For $S \neq T$ we have $\sum_x \chi_{S \Delta T}(x) = 0$; since we are summing over all $x \in \{\pm 1\}^n$. For $S = T$, the inner product $\langle \chi_S, \chi_S \rangle$ is equal to one. Hence, we get orthonormality. In particular, the characters χ_S form a complete basis for our vector space V, and every function $f \in V$, $f : \{\pm 1\} \to \mathbb{R}$ can be decomposed as

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S) \cdot \chi_S(x),$$

where $\hat{f}(S) \in \mathbb{R}$ is called the Fourier coefficient at S. Note that by orthogonality of the Fourier basis we have $\hat{f}(S) = \langle f, \chi_S \rangle$. Next we state a useful property of the Fourier coefficients.

Proposition 3. *For any functions $f, g : \{\pm 1\} \to \mathbb{R}$ we have $\langle f, g \rangle = \sum_S \hat{f}(S) \hat{g}(S)$.*

This proposition implies a useful corollary—called Parseval's identity—for ±1 valued functions. Note that when f is ±1 valued, the inner product $\langle f, f\rangle = 1$. This along with the proposition above, $\langle f, f\rangle = \sum_S \hat{f}(S)^2$, gives us Parseval's identity:

Corollary 4. *For any ±1-valued function f, we have $\sum_S \hat{f}(S)^2 = 1$.*

0.1 Proof of Theorem 2

Using Fourier analytic tools we will now prove Theorem 2. The assumption of the theorem implies that

$$\begin{aligned}
1 - 2\varepsilon &= \frac{1}{2^{2n}} \sum_{\alpha,\alpha'} f(\alpha) f(\alpha') f(\alpha\alpha') \\
&= \frac{1}{2^{2n}} \sum_{\alpha\alpha'} \sum_{S,T,U} \hat{f}(S)\hat{f}(T)\hat{f}(U)\chi_S(\alpha)\chi_T(\alpha')\chi_U(\alpha\alpha') \qquad \text{(via Fourier expansion)} \\
&= \frac{1}{2^{2n}} \sum_{\alpha\alpha'} \sum_{S,T,U} \hat{f}(S)\hat{f}(T)\hat{f}(U)\chi_S(\alpha)\chi_T(\alpha')\chi_U(\alpha)\chi_U(\alpha')
\end{aligned}$$

Here, the last equality follows from the fact that $\chi_U(\alpha\alpha') = \chi_U(\alpha)\chi_U(\alpha')$; this equality is obtained just from the definition of χ_U. Rewriting the right-hand-side of the above equation we obtain:

$$\begin{aligned}
1 - 2\varepsilon &= \sum_{S,T,U} \hat{f}(S)\hat{f}(T)\hat{f}(U) \left(\frac{1}{2^n}\sum_{\alpha} \chi_S(\alpha)\chi_T(\alpha)\right) \left(\frac{1}{2^n}\sum_{\alpha} \chi_T(\alpha')\chi_U(\alpha')\right) \\
&= \sum_{S,T,U} \hat{f}(S)\hat{f}(T)\hat{f}(U) \langle \chi_S, \chi_T\rangle \, \langle \chi_T, \chi_U\rangle
\end{aligned}$$

The orthogonality of χs implies that only the summands where $S = T = U$ are non-zero, in fact are equal to one. The remaining summands, where either $S \neq T$ or $T \neq U$, are equal to zero. Therefore, the following inequality holds:

$$\begin{aligned}
1 - 2\varepsilon &\leq \sum_{U} \hat{f}(U)^3 \\
&\leq \max_W \hat{f}(W) \, \sum_U \hat{f}(U)^2 \\
&= \max_W \hat{f}(W) \qquad \text{(by Parseval's identity)}.
\end{aligned}$$

Write $S_0 = \arg\max_S \hat{f}(S)$. By the inequality above, $\hat{f}(S_0) \geq 1 - 2\varepsilon$.

We can now prove Theorem 2. Recall that the Fourier coefficient $\hat{f}(S_0) = \frac{1}{2^n}\sum_\alpha f(\alpha)\chi_{S_0}(\alpha)$. But, the inequality $\hat{f}(S_0) \geq 1 - 2\varepsilon$ holds iff $f(\alpha) = \chi_{S_0}(\alpha)$ for at least $(1-\varepsilon)2^n$ α's; both f and χ_{S_0} are ±1 valued. In other words, for at most $\varepsilon 2^n$ many α's for f and χ_{S_0} differ. This completes the proof of the theorem.

PCP **Verifier**

We give the complete PCP verifier for **QUADEQ** that uses the linearity test described above. Recall that the a correct proof $\Pi = (\Pi^1, \Pi^2)$ must consist of $(\Pi^1)_\alpha = \alpha \cdot x$ and $(\Pi^2)_\beta = \beta \cdot (x \otimes x)$ for some satisfying assignment x of the given **QUADEQ** instance φ.

To verify the proof Π, the verifier performs the each of the following with probability $1/4$.

1. Linearity test on Π^1. That is, we use Theorem 2 with $f = (-1)^{\Pi^1}$. This is equivalent to querying $(\Pi^1)_\alpha$, $(\Pi^1)_{\alpha'}$ and $(\Pi^1)_{\alpha+\alpha'}$ for $\alpha, \alpha' \in_u \{0,1\}^n$, and testing whether $(\Pi^1)_\alpha + (\Pi^1)_{\alpha'} = (\Pi^1)_{\alpha+\alpha'}$.

2. Linearity test on Π^2

3. Consistency of tensor product: Choose $\alpha, \alpha' \in \{0,1\}^n$, $\beta \in \{0,1\}^{n^2}$ uniformly at random. Then check $(\Pi^1)_\alpha (\Pi^1)_{\alpha'} = (\Pi^2)_{(\alpha \otimes \alpha')+\beta} + (\Pi^2)_\beta$.

4. As in the previous lecture, construct an equation at random, say with coefficients (α, β, γ), and check $(\Pi^1)_\alpha + (\Pi^2)_\beta = \gamma$.

We will skip the detailed analysis of this verifier. But, we have presented all the required elements to establish the correctness and soundness of this verifier. Overall, if φ is satisfiable then there exists a proof Π that is accepted with probability 1. On the other hand, if φ is not satisfiable then any proof is rejected with probability at least 0.001 (the exact constant is not relevant here). This implies that we have soundness at most 0.999. Repeating the verifier's test a large enough (constant) number of times to amplify the soundness, we obtain Theorem 1: NP $\subseteq$ PCP$_{1,1/2}(O(poly(n)), O(1))$. Note that the length of the proof Π is $2^n + 2^{n^2}$ bits, and the number of random bits required by the verifier is at most $2 + 2n^2$.

Recall that the PCP theorem is a stronger version of Theorem 1 that claims that NP $\subseteq$ PCP$(O(\log n), O(1))$. In order to get to the PCP theorem we need to save on randomness, or equivalently make the proof much shorter. We can accomplish this by extending our framework to low-degree polynomials over $\mathbb{F}_p$.

Low-degree testing

Without going into any amount of detail we give a flavor of the ideas that go into designing a much more randomness-efficient version of the PCP verifier described in the previous section. Abstracting the details of the particular NP-complete problem we chose, the proof $\Pi = (\Pi^1, \Pi^2)$ can be though of as a (very long) encoding of a candidate assignment to formula φ that has the following properties:

1. Local testability: it is possible to verify that Π is close to a proof having the correct format, $\Pi^1 = \alpha \cdot x$ and $\Pi^2 = \beta \cdot (x \cdot x)$ for some x, by making only a constant number of queries to Π;

2. Local decodability: given any coordinate x_i of x that we may be interested in, it is possible to recover x_i from the proof by making a constant number of *uniformly distributed* queries (the fact that the queries are uniformly distributed is important to ensure compatibility with item 1; this was used in test 3. of our verifier):

$$x_i = x \cdot e_i = x \cdot (e_i + \alpha) + x \cdot \alpha = (\Pi^1)_{e_i+\alpha} + (\Pi_1)_\alpha$$

for any $\alpha \in \{0,1\}^n$.

The proof of the PCP theorem is based on the design of encodings having these two properties that are much more efficient than the linear encoding we used above. These are based on the use of *low-degree polynomials*. To give a flavor of how polynomials come into play, let's fix a large prime p (of size polynomial in n) and set $m = \log n / \log\log n$. Let $\mathbb{F}_p$ be the finite field with p elements. A degree-d polynomial $f : \mathbb{F}_p^m \to \mathbb{R}$ is any function that has an expansion

$$f(x) = \sum_{i_1,\dots,i_m} c_{i_1,\dots,i_m} x_1^{i_1} \cdots x_m^{i_m}.$$

The degree of f is the largest $i_1 + \cdots + i_m$ such that $c_{i_1,\dots,i_m} \neq 0$. Note that we may always assume $i_1, \dots, i_m \in \{0, \dots, p-1\}$, since over $\mathbb{F}_p$ we have $x^p = x$.

Let q be an integer of size approximately $\log n$, so that $|\{0, \dots, q-1\}^m| = q^m > n$ Let ι be an arbitrary injection from $\{1, \dots, n\}$ into $\{0, \dots, q-1\}^m$. Define a function $\tilde{f}$ by $\tilde{f}(\iota(k)) = x_k$ for $k \in \{1, \dots, n\}$. As it turns out, $\tilde{f}$ can be extended to a polynomial f defined over the whole of $\mathbb{F}_p^m$ that has degree at most q in each variable. In particular the total degree is at most $d \leq qm$, which is polylogarithmic in n.

We think of the polynomial f as giving an encoding of x (this is basically the Reed-Muller code). Note that describing f completely requires specifying $p^m = n^{O(\log n)}$ values. This is not quite polynomial (and more work is required in order to bring this down to a polynomial), but it is already much better than the exponential-size proof we obtained from linear encodings.

Now, is f locally testable/decodable? A natural test would be as follows: pick a random line $\ell \subset \mathbb{F}_p^m$, and query $d+1$ values of f along that line. If f is of total degree d, its restriction $f_{|\ell}$ to any line also has degree at most d and these $d+1$ values should be enough to recover a complete description of the univariate polynomial $f_{|\ell}$. We could then query a $(d+2)$-th value of f, and check whether it agrees with the polynomial recovered from the first $d+1$ queries.

The problem with this is that the number of queries, $d+2$, remains large, polylogarithmic in n. To get a code that can be tested with a constant number of queries the idea is to add even more redundancy: in addition to listing all possible values of f, the proof will list all the $f_{|\ell}$, for every possible line ℓ. That is, $\Pi = (\Pi^1, \Pi_2)$ with $\Pi^1 \in (\mathbb{F}_p)^{p^m}$ such that $(\Pi^1)_x = f(x)$, and $\Pi^2 \in (\mathbb{F}_p)^{(p^m)^2}$ with $(\Pi^2)_\ell = f_{|\ell}$. Note the proof still has length $2^{\text{poly}\log n}$. And now we have a simple local test: choose a line ℓ at random, and $x \in \ell$ at random. Check that $(\Pi^1)_x = ((\Pi^2)_{|\ell})(x)$. Local decoding is even simpler: given a point z for which we want to know $f(x)$, choose a random line ℓ such that $z \in \ell$, query $(\Pi^2)_\ell$, and use it to recover $f(z)$.

This is the flavor of the codes that come into the construction of the PCP verifier showing $\mathrm{NP} \in \mathrm{PCP}(O(\log n), O(1))$. The actual details are a little bit (not too much) more complicated; in particular we did not describe the remaining important step of *composition*. But we got most of the ideas!

References

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of computer and system sciences*, 47(3):549–595, 1993.