CSE 533: The PCP Theorem and Hardness of Approximation (Autumn 2005)

Lecture 8: Linearity and Assignment Testing

26 October 2005

Lecturer: Venkat Guruswami *Scribe: Paul Pham & Venkat Guruswami*

1 Recap

In the last class, we covered the Composition Theorem except for the $O(1)$-query assignment tester (AT). Today we will develop some machinery relating to linearity testing to be used in the construction of such an AT. This can be treated as a self-contained lecture on linearity testing and the necessary Fourier analysis.

2 Linearity Testing

We work in an n-dimensional binary vector space $\{0,1\}^n$ with inner product $x \cdot y = \sum_{i=1}^{n} x_i y_i$ defined as usual, where the summation is done modulo 2 (in other words, it is the binary $\oplus$ operator). The binary $\oplus$ operator will denote bitwise XOR on two vectors. The linear functions on $\{0,1\}^n$ are of the form $L(x) = a \cdot x$ for some $a \in \{0,1\}^n$. There are 2^n such functions. If $S \subseteq \{1, 2, \ldots, n\}$ is the set $\{i : S \ni i\}$, then clearly $L(x) = \sum_{i \in S} x_i$ (again, the summation is in the field $\{0,1\}$). We will use such subsets to index linear functions — the linear function L_S corresponding to a subset $S \subset \{1, 2, \ldots, n\}$ is defined as

$$L_S(x) = \sum_{i \in S} x_i$$

Definition 2.1 (linear function)**.** *A function $f : \{0,1\}^n \to \{0,1\}$ is linear if*

$$\forall_{x,y \in \{0,1\}^n} : f(x+y) = f(x) + f(y) \tag{1}$$

(where $x + y$ denotes coordinatewise addition mod 2).

Alternatively, a function f is linear if it is equal to L_S for some $S \subset \{1, 2, \ldots, n\}$.

The goal of *linearity testing* is then: given $f : \{0,1\}^n \to \{0,1\}$ as a table of 2^n values, check whether f is linear using just a few probes into the table f. Clearly, we cannot check that f is exactly linear without looking at the entire table, so we will aim to check whether f is close to a linear function. A natural test is the Blum-Luby-Rubinfeld (BLR) test where we check the above condition (1 for a single triple $(x, y, x+y)$ with x, y chosen randomly and independently. Formally the BLR test proceeds as follows:

1. Pick $x, y \in \{0,1\}^n$ uniformly and independently at random.

2. If $f(x+y) = f(x) + f(y)$ accept, otherwise reject.

We can describe the distance/closeness between two functions as follows.

Definition 2.2. *We say two functions f and g are δ-*far *if they differ in at least a fraction δ of places, $0 \leq \delta \leq 1$, i.e.,*

$$\Pr_{x \in \{0,1\}^n} [f(x) \neq g(x)] \geq \delta$$

We say two functions f and g are δ-close if they differ in at most a fraction δ of places, $0 \leq \delta \leq 1$, i.e.,

$$\Pr_{x \in \{0,1\}^n} [f(x) \neq g(x)] \leq \delta$$

The completeness of the above test is obvious.

Theorem 2.3 (BLR Completeness)**.** *If f is linear, the BLR test accepts with probability 1.*

In the rest of this lecture we will show the following on the efficacy of the test in detecting the non-linearity of f as a function of the distance of f from linearity.

Theorem 2.4 (BLR Soundness)**.** *If f is δ-far from every linear function then the BLR test rejects with probability at least δ.*

3 Notation Change

We now introduce a change in notation from Boolean binary values in $\{0,1\}$ to $\{1,-1\}$ which turns out to be more convenient. The transformation for $a \in \{0,1\}$ is:

$$a \to (-1)^a$$

which maps 0 to 1, 1 to -1. The advantage with this change is that the xor (or summation mod 2) operation becomes multiplication.

$$a + b \to (-1)^{a+b} = (-1)^a(-1)^b$$

We now consider functions $f : \{-1,1\}^n \to \{-1,1\}$ and have a new form for the linear function associated with a subset $S \subset \{1,2,\ldots,n\}$:

$$\chi_S(x) = \prod_{i \in S} x_i \tag{2}$$

and a new linearity test that checks

$$f(x \cdot y) = f(x) \cdot f(y)$$

for randomly chosen $x, y \in \{1,-1\}^n$, where $x \cdot y$ denotes coordinatewise multiplication, i.e., $(x \cdot y)_i = x_i y_i$.

The expression $f(x)f(y)f(xy)$ equals 1 if the test accepts for that choice of x, y, and equals -1 if the test rejects. The quantity

$$\left(\frac{1 - f(x)f(y)f(xy)}{2}\right)$$

is thus an indicator for whether the test accepts. It follows that the probability that the BLR test rejects using this new notation can be expressed as:

$$\Pr[\text{ BLR test rejects}] = \mathop{\mathrm{E}}_{x,y\in\{1,-1\}^n}\left[\frac{1 - f(x)f(y)f(xy)}{2}\right] \tag{3}$$

In order to calculate the value of this expectation, we will need some background in discrete Fourier analysis.

4 Fourier Analysis on Discrete Binary Hypercube

Consider the vector space $\mathbb{G}$ consisting of all n-bit functions from $\{-1,1\}^n$ to the real numbers:

$$\mathbb{G} = \{g \mid g : \{-1,1\}^n \to \mathbb{R}\}$$

$\mathbb{G}$ has dimension 2^n, and a natural basis for it are the indicator functions $e_a(x) = \not\Vdash(x = a)$ for $a \in \{1,-1\}^n$. The coordinates of $g \in \mathbb{G}$ in this basis is simply the table of values $g(a)$ for $a \in \{1,-1\}^n$.

We will now describe an alternate basis for $\mathbb{G}$. We begin with an inner product on this space defined as follows:

$$\langle f, g\rangle = \frac{1}{2^n}\sum_{x\in\{-1,1\}^n} f(x)g(x) \tag{4}$$

The linear functions $\chi_S(x)$ for various subsets S form an orthonormal basis with respect to the above inner product. Since $|\chi_S(x)| = 1$ for every S, x, clearly $\langle\chi_S, \chi_S\rangle = 1$ for all $S \subseteq \{1,2,\ldots,n\}$. The following shows that different linear functions are orthogonal w.r.t the inner product (4).

Lemma 4.1.

$$S \neq T \to \langle\chi_S, \chi_T\rangle = 0$$

Proof:

$$\begin{aligned}
\langle\chi_S, \chi_T\rangle &= \frac{1}{2^n}\sum_{x\in\{-1,1\}^n} \chi_S(x)\chi_T(x) \\
&= \frac{1}{2^n}\sum_{x\in\{-1,1\}^n}\left(\prod_{i\in S} x_i\right)\left(\prod_{i\in T} x_i\right) \\
&= \frac{1}{2^n}\sum_{x\in\{-1,1\}^n}\left(\prod_{i\in S\triangle T} x_i\right) \\
&= 0
\end{aligned}$$

$S \triangle T$ denotes the symmetric difference of S and T. This is not empty because we have specified $S \neq T$. The last step follows because we can always pair any x with an $\tilde{x}$ such that $x_j = -\tilde{x}_j$ for a fixed $j \in S \triangle T$. $\square$

Thus we have shown that the $\{\chi_S\}$ form an orthonormal basis for $\mathbb{G}$. We can therefore any function f in this basis as follows (in what follows we use $[n]$ to denote the set $\{1, 2, \dots, n\}$):

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

where the coefficients $\hat{f}(S)$ w.r.t this basis are called the *Fourier coefficients* of f, and are given by

$$\hat{f}(S) = \langle f, \chi_S \rangle = \frac{1}{2^n} \sum_{x \in \{1,-1\}^n} f(x) \chi_S(x) \; .$$

Fact 4.2 (Fourier Coefficients of a Linear Function)**.**

$$f \textit{ linear} \iff \Big(\exists S \subseteq [n] : \hat{f}(S) = 1 \Big) \wedge \Big(\forall T \subseteq [n], T \neq S : \hat{f}(T) = 0 \Big)$$

The following describes how the Fourier coefficients are useful in understanding of a function to the various linear functions.

Lemma 4.3. *For every $S \subseteq [n]$,*

$$\hat{f}(S) = 1 - 2\mathrm{dist}\,(f, \chi_S) = 1 - 2 \Pr_{x \in \{1,-1\}^n} \Big[f(x) \neq \chi_S(x) \Big] \; . \tag{5}$$

Proof:

$$\begin{aligned} 2^n \hat{f}(S) &= \sum_x f(x) \chi_S(x) \\ &= \sum_{x : f(x) = \chi_S(x)} 1 + \sum_{x : f(x) \neq \chi_S(x)} (-1) \\ &= 2^n - 2 |\{x \mid f(x) \neq \chi_S(x)\}| \\ &= 2^n (1 - 2 \Pr_x \Big[f(x) \neq \chi_S(x) \Big]) \end{aligned}$$

It follows that $\hat{f}(S) = 1 - 2\mathrm{dist}\,(f, \chi_S)$ as claimed. $\square$

In particular, the above implies that any two distinct linear functions differ in exactly $1/2$ of the points. This implies that in the *Hadamard code* which we define below the encodings of two distinct messages differ in exactly a fraction $1/2$ of locations.

Definition 4.4 (Hadamard code)**.** *The Hadamard encoding of a string $a \in \{0,1\}^n$, denoted $\mathrm{Had}(a) \in \{0,1\}^{2^n}$, is defined as follows. Its locations are indexed by strings $x \in \{0,1\}^n$, and $\mathrm{Had}(a)_{|x} = a \cdot x$.*

Parseval's identity

We now state a simple identity that the Fourier coefficients of a Boolean function must obey.

Lemma 4.5. *For any* $f : \{-1,1\}^n \to \{-1,1\}$,

$$\sum_{S\subseteq[n]} \hat{f}(S)^2 = 1 \; .$$

Proof:

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

On the other hand

$$\begin{aligned}
\langle f, f\rangle &= \langle \sum_{S\subseteq[n]} \hat{f}(S)\chi_S, \sum_{T\subseteq[n]} \hat{f}(T)\chi_T\rangle \\
&= \sum_{S\subseteq[n]} \hat{f}(S)^2\langle\chi_S,\chi_S\rangle \text{ (since } \langle\chi_S,\chi_T\rangle = 0 \text{ for } S \neq T) \\
&= \sum_{S\subseteq[n]} \hat{f}(S)^2 \; . \quad \square
\end{aligned}$$

5 Proof of BLR Soundness

We now set out to prove Theorem 2.4. By Equation (3) we need to analyze the expectation:

$$\begin{aligned}
& \mathop{\mathrm{E}}_{x,y}\left[f(x)f(y)f(xy)\right] \\
= \;& \mathop{\mathrm{E}}_{x,y}\left[\left(\sum_S \hat{f}(S)\chi_S(x)\right)\left(\sum_T \hat{f}(T)\chi_T(x)\right)\left(\sum_U \hat{f}(U)\chi_U(xy)\right)\right] \\
= \;& \mathop{\mathrm{E}}_{x,y}\left[\sum_{S,T,U}\left(\hat{f}(S)\hat{f}(T)\hat{f}(U)\chi_S(x)\chi_T(y)\chi_U(xy)\right)\right] \\
= \;& \sum_{S,T,U} \hat{f}(S)\hat{f}(T)\hat{f}(U) \mathop{\mathrm{E}}_{x,y}\left[\chi_S(x)\chi_T(y)\chi_U(xy)\right]
\end{aligned}$$

We claim that the expectation in the last line is 0 unless $S = T = U$. Indeed this expectation equals

$$\begin{aligned}
& \mathop{\mathrm{E}}_{x,y}\left[\left(\prod_{i\in S} x_i\right)\left(\prod_{j\in T} y_j\right)\left(\prod_{k\in U} x_k\right)\left(\prod_{l\in U} y_l\right)\right] \\
= \ & \mathop{\mathrm{E}}_{x,y}\left[\left(\prod_{i\in S\triangle U} x_i\right)\left(\prod_{j\in T\triangle U} y_j\right)\right] \\
= \ & \mathop{\mathrm{E}}_{x}\left[\prod_{i\in S\triangle U} x_i\right]\mathop{\mathrm{E}}_{y}\left[\prod_{j\in T\triangle U} y_j\right]
\end{aligned}$$

If $S \neq U$ or $T \neq U$, then one of the symmetric differences is non-empty, and the expectation is 0, as claimed.

Hence we have the following expression for the desired expectation:

$$\begin{aligned}
\mathop{\mathrm{E}}_{x,y}\left[f(x)f(y)f(xy)\right] &= \sum_{S\subseteq[n]} \hat{f}(S)^3 \\
&\leq \max_S \hat{f}(S) \sum_S \hat{f}(S)^2 \\
&= \max_S \hat{f}(S) \text{ (using Parseval's identity)} . \\
&= 1 - 2\min_S \mathrm{dist}\left(f, \chi_S\right)
\end{aligned}$$

where the last step used Lemma 4.3. Together with (3) we have the following conclusion:

$$\Pr\left[\text{ BLR test rejects}\right] \geq \min_S \mathrm{dist}\left(f, \chi_S\right)$$

This is precisely the claim of Theorem 2.4.

6 Self-Correction

Another tool which will be useful in constructing an assignment tester is a self-correction procedure for the Hadamard code. Assume we have $f : \{0,1\}^n \to \{0,1\}$, a function or table of values, that is δ-close to some linear function L. (We now move back to the $\{0,1\}$ notation; the $\{1,-1\}$ notation was used only for analysing the linearity test.)

Remark 6.1. *If $\delta < \frac{1}{4}$ then such a δ-close L is uniquely determined.*

Using f we would like to compute $L(x)$ for any desired x with high accuracy. (Note that simply probing f at x doesn't suffice since x could be one of the points where f and L differ.) Such a procedure is called a self-correction procedure since a small amount of errors in the table f can be corrected using probes only to f to provide access to a noise-free version of the linear function L.

Consider the following procedure:

Procedure Self-Corr(f, x):

1. Select $y \in \{0,1\}^n$ uniformly at random.

2. Return $f(x+y) - f(y)$

Lemma 6.2. *If f is δ-close to a linear function L for some $\delta < 1/4$, then for any $x \in \{0,1\}^n$ the above procedure Self-Corr(f, x) computes $L(x)$ with probability at least $1 - 2\delta$.*

Proof: Since y and $x + y$ are both uniformly distributed in $\{0,1\}^n$, we have

$$\Pr_y [f(x+y) \neq L(x+y)] \leq \delta$$

$$\Pr_y [f(y) \neq L(y)] \leq \delta$$

Thus with probability at least $1 - 2\delta$, $f(x+y) - f(y) = L(x+y) - L(y) = L(x)$, and so Self-Corr$(f, x)$ outputs $L(x)$ correctly. □

7 Constant Query Assignment Tester: Arithmetizing Circuit-SAT

We now have a way to test for linearity and self-correction procedure to gain access to the Hadamard encoding of a string using oracle access to a close-by function. How can we use this for assignment testing? Let's review the assignment testing problem.

Let Φ be a circuit on Boolean variables X and Ψ be a collection of constraints on on $X \cup Y$ where Y are auxiliary Boolean variables produced by an assignment tester. We want the following two properties:

1. if $\Phi(a) = 1$, then $\exists b$ such that $\forall \psi \in \Psi$, $a \cup b$ satisfies ψ

2. If a is δ-far from every a' for which $\Phi a' = 1$ then $\forall b \Pr_{\psi \in \Psi} [(a \cup b)$ violates $\psi] = \Omega(\delta)$.

We can reduce any given circuit over Boolean variables (CIRCUIT-SAT) to a set of quadratic equations over $\mathbb{F}_2$ (QFSAT). Then the existence of solutions for the system of equations implies that the original circuit is satisfiable, and vice versa. How do we do this? We have one $\mathbb{F}_2$-valued variable w_i for each gate of the circuit Φ (including each input gate that each has an input variable connected to it). We only need to provide quadratic equations that enforce proper operation of AND and NOT gates.

- $(w_k = w_i \wedge w_j) \rightarrow (w_k - w_i w_j = 0)$
- $(w_l = \neg w_i) \rightarrow (w_i + w_l - 1 = 0)$

If w_N denotes the variable for the output gate, we also add the equation $w_N - 1 = 0$ to enforce that the circuit outputs 1.

It is easy to check that all these constraints can be satisfied iff Φ is satisfiable.

In the next lecture, we'll describe how to use linearity testing and self-correction codes to give an assignment tester for input an arbitrary circuit Φ.