Harmonic Analysis and Linearity testing.

April 16, 2006

This material is described in Chapter 19.6 in the book.

Linearity testing Essentially the only thing left to prove the **PCP** theorem is to prove the following theorem:

Recall that a *linear function* from $\{0,1\}^n$ to $\{0,1\}$ is a function L satisfying $L(x \oplus y) = L(x) \oplus L(y)$ or equivalently, $L(x) = \langle x, \alpha \rangle \pmod 2 = \sum_i x_i \alpha_i \pmod 2$ for some $\alpha \in \{0,1\}^n$ (that is, L is the Hadamard encoding of α). Another equivalent way to describe L is $L(x) = \sum_{i \in \alpha} x_i \pmod 2$ for $\alpha \subseteq [n]$. (Where $[n] = \{1, \ldots, n\}$.)

Theorem 1. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function, and assume $p = \Pr_{x,y}[f(x) \oplus f(y) = f(x \oplus y)] \geq \frac{1}{2}$. Then, there exists a linear function $L : \{0,1\}^n \to \{0,1\}$ such that $\Pr[f(x) = L(x)] \geq p$.*

Moving to ± 1 It will be convenient to do the following transformation:

$$0 \mapsto +1$$
$$1 \mapsto -1$$
$$\oplus \mapsto \cdot$$

For $x, y \in \{\pm 1\}^n$ we'll denote by $x \oplus y$ the componentwise multiplication of x and y. That is, $(x \oplus y)_i = x_i y_i$. We'll now call a function $L : \{\pm 1\}^n \to \{\pm 1\}$ *linear* if $L(x \oplus y) = L(x)L(y)$ for all x, y. Equivalently, L is linear if $L(x) = \Pi_{i \in \alpha} x_i$ for some $\alpha \subseteq [n]$. We denote the linear function $x \mapsto \Pi_{i \in \alpha} x_i$ by $\chi_\alpha(cdot)$.

For functions $f, g : \{\pm 1\}^n \to \{\pm 1\}$, we define $\langle f, g \rangle = \mathbb{E}[f(x)g(x)] = 2^{-n} \sum_{x \in \{\pm 1\}} f(x)g(x)$. Note that

$$\langle f, g \rangle = \Pr[f(x) = g(x)] - \Pr[f(x) \neq g(x)] = 2 \Pr[f(x) = g(x)] - 1$$

or in other words $\Pr[f(x) = g(x)] = \frac{1}{2} + \frac{1}{2}\langle f, g \rangle$ and so the larger $\langle f, g \rangle$, the closer (in Hamming distance) f and g are.

We can generalize the definition of $\langle f, g \rangle$ to functions $f, g : \{\pm 1\} \to \mathbb{R}$ and then we get that $\langle f, g \rangle$ satisfies the standard properties of an inner product: $\langle f, f \rangle > 0$ for every non-zero f, $\langle f, g \rangle = \langle g, f \rangle$ and $\langle \alpha f + \beta f', g \rangle = \alpha \langle f, g \rangle + \beta \langle f', g \rangle$.

Rephrasing the theorem Another way to phrase Theorem 1 is the following

Theorem 2. *For a function $f : \{\pm 1\}^n \to \{\pm 1\}$ and $\alpha \subseteq [n]$, denote $\hat{f}(\alpha) = \langle f, \chi_\alpha \rangle$. If $\hat{f}(\alpha) \leq \epsilon$ for every α then*

$$2\Pr[f(x)f(y) = f(x \oplus y)] - 1 = \mathbb{E}[f(x)f(y)f(x \oplus y)] \leq \epsilon$$

Some linear algebra Let's look at a function $f : \{\pm 1\}^n \to \{\pm 1\}$ (or more generally $f : \{\pm 1\}^n \to \mathbb{R}$) as a 2^n-dimensional real vector. We can think of f as equal to $\sum_{z \in \{\pm 1\}^n} f(z)e_z$ where e_z is the standard basis vector: $e_z(x) = 0$ for $x \neq z$ and $e_z(z) = 1$. That is, for every x, $f(x) = \sum_z f(z)e_z(x)$.

However, the vectors $\{\chi_\alpha\}$ are also a basis for $\mathbb{R}^{2^n}$. In fact, they are an *orthonormal basis* under our definition of inner product:

Lemma 3. *For every α, β,*

$$\langle \chi_\alpha, \chi_\beta \rangle = \delta_{\alpha,\beta} \tag{1}$$

, where

$$\delta_{\alpha,\beta} = \begin{cases} 1 & \alpha = \beta \\ 0 & \alpha \neq \beta \end{cases}$$

Proof. Since χ_α has values in $\{\pm 1\}$, clearly $\langle \chi_\alpha, \chi_\alpha \rangle = 2^{-n} \sum_x (\chi_\alpha(x))^2 = 2^{-n}2^n = 1$.

Suppose $\alpha \neq \beta$, then

$$2^n \langle \chi_\alpha, \chi_\beta \rangle = \sum_x \chi_\alpha(x)\chi_\beta(x) = \chi_{\alpha \oplus \beta}(x)$$

(where $\alpha \oplus \beta$ denotes the symmetric difference in sets and this follows from the fact that for $x_i \in \pm$, $x_i^2 = 1$. Since for every nonzero $\gamma \in \{0,1\}^n$, $\Pr_{r \in \{0,1\}^n}[\langle \gamma, r \rangle = 1] = 1/2$, this sum will have half the values $+1$ and half -1 and so will be zero. □

Fourier transform: Since $\{\chi_\alpha\}$ is a basis, we can express any function $f : \{\pm 1\}^n \to \mathbb{R}$ in this basis, in fact we have the following equation:

$$f = \sum_\alpha \hat{f}(\alpha)\chi_\alpha$$

Indeed, suppose $f = \sum_\alpha f'(\alpha)\chi_\alpha$. Then, for every β we have that $\hat{f}(\beta) = \langle f, \chi_\beta \rangle = \sum_\alpha f'(\alpha)\langle \chi_\alpha, \chi_\beta \rangle = \sum_\alpha f'(\alpha)\delta_{\alpha,\beta} = f'(\alpha)$.

We call the value $\hat{f}(\alpha)$ the α^{th} Fourier coefficient of f.

We also have *Parseval's equality* that the norm of f is invariant under change to orthonormal basis or in other words that

$$\langle f, f \rangle = 2^{-n} \sum_x f(x)^2 = \sum_\alpha \hat{f}(\alpha)^2$$

In particular, if $f(x) \in \{\pm 1\}$ for all x then $\langle f, f \rangle = 1$ and hence $\sum_\alpha \hat{f}(\alpha)^2 = 1$.

Proving Theorem 2 Suppose that $\hat{f}(\alpha) \leq \epsilon$ for every α. We need to prove that $\mathbb{E}[f(x)f(y)f(x \oplus y)] \leq \epsilon$. This follows from the following calculation:

$$\mathbb{E}_{x,y}[f(x)f(y)f(x \oplus y)] = 2^{-2n} \sum_{x,y} f(x)f(y)f(x \oplus y) \underset{\text{basis change}}{=}$$

$$2^{-2n} \sum_{x,y} \left(\sum_{\alpha} \hat{f}(\alpha)\chi_\alpha(x) \right) \left(\sum_{\beta} \hat{f}(\beta)\chi_\beta(y) \right) \left(\sum_{\gamma} \hat{f}(\gamma)\chi_\gamma(x \oplus y) \right) \underset{\text{linearity of } \chi_\gamma}{=}$$

$$2^{-2n} \sum_{x,y} \left(\sum_{\alpha} \hat{f}(\alpha)\chi_\alpha(x) \right) \left(\sum_{\beta} \hat{f}(\beta)\chi_\beta(y) \right) \left(\sum_{\gamma} \hat{f}(\gamma)\chi_\gamma(x)\chi_\gamma(y) \right) \underset{\text{reordering summations}}{=}$$

$$\sum_{\alpha,\beta,\gamma} \hat{f}(\alpha)\hat{f}(\beta)\hat{f}(\gamma) \left(2^{-n} \sum_{x} \chi_\alpha(x)\chi_\gamma(x) \right) \left(2^{-n} \sum_{y} \chi_\beta(y)\chi_\gamma(y) \right) \underset{(1)}{=}$$

$$\sum_{\alpha,\beta,\gamma} \hat{f}(\alpha)\hat{f}(\beta)\hat{f}(\gamma)\delta_{\alpha,\gamma}\delta_{\beta,\gamma} = \sum_{\alpha} \hat{f}(\alpha)^3 \leq \sum_{\alpha} \epsilon \hat{f}(\alpha)^2 \underset{\text{Parseval}}{=} \epsilon \cdot 1$$

□

Learning the Fourier coefficients of a function: Let $f : \{\pm 1\}^n \to \{\pm 1\}$ be some function. Because of Parseval's equality, there can be at $1/\epsilon^2$ coefficients $\hat{f}(\alpha)$ satisfying $\hat{f}(\alpha) \geq \epsilon$. A natural question is the following: given oracle access to $f(\cdot)$, can we *learn* these coefficients?

We'll now show this is possible. That is, we prove the following theorem:

Theorem 4 (Goldreich-Levin 89)**.** *There's a* $\mathrm{poly}(n, 1/\epsilon, \log(1/\delta))$ *algorithm* A *that with oracle access to any* $f : \{\pm 1\}^n \to \{\pm 1\}$ *such that* $\hat{f}(\alpha) \geq \epsilon$ *for some* α*, with probability* $1 - \delta$*,* A^f *outputs a list of size* $O(1/\epsilon^2)$ *containing all the* α *with* $\hat{f}(\alpha) \geq \epsilon$*.*

Going back from the $\{\pm 1\}$ representation to the $\{0,1\}$ representation, this means that if we have a function $f : \{0,1\}^n \to \{0,1\}$ that has agreement $\frac{1}{2} + \epsilon$ with some linear function $r \mapsto \langle z, r \rangle \pmod 2$ then we can find z with probability $O(\epsilon^2)$. In cryptography this is used to prove that if $f(x)$ is a one-way permutation, then $f(x, r) = f(x) \circ r \circ \langle x, r \rangle \pmod 2$ is a pseudorandom generator. (This is the well known hard-core bit theorem.)

Another way to look at this theorem is that this is *local list decoding* of the Hadamard code: given oracle access to f, find the list of $O(1/\epsilon^2)$ Hadamard codewords that are $\frac{1}{2} + \epsilon$-close to f.

Proof of Theorem 4 For $k \leq n$ and $\alpha \in \{0,1\}^k$, we'll denote by f_α to be the sum of squares of Fourier coefficients of f starting with α. That is, $f_\alpha = \sum_{\beta \in \{0,1\}^{n-k}} \hat{f}(\alpha \circ \beta)^2$. If $f : \{\pm 1\}^n \to \{\pm 1\}$ then by Parseval $f_\lambda = 1$ (where λ is the empty word). Also, for every $\alpha \in \{0,1\}^k$, $f_\alpha = f_{\alpha 0} + f_{\alpha 1}$. Think of a full binary tree with the root labeled by $f_\lambda = 1$ and the two children of a node labeled by f_α are labeled by $f_{\alpha 0}$ and $f_{\alpha 1}$. The values at each level of the tree sum up to 1, and the goal of the algorithm is to find the list leafs that are labeled with values at least ϵ.

The outline of the algorithm will be as follows: we'll have a procedure `Estimate` that given α and oracle access to $f(\cdot)$, will estimate f_α up to $\epsilon/4$ accuracy with probability $1 - \delta\epsilon/n$.

We'll work our way from the root down, and whenever we `Estimate`(α) gives a value smaller than $\epsilon/2$ we will "kill" this node and will not deal with it and its subnodes. Note that, unless the output of `Estimate` is more than $\epsilon/4$-far from the real value (which we'll ensure by the union bound will happen with probability at most δ) at most $4/\epsilon$ nodes will survive at any level. The algorithm will output the $4/\epsilon$ leaves that survive.

The procedure `Estimate`: At the heart of `Estimate` is the following lemma:

Lemma 5. *For every α^0,*

$$f_{\alpha^0} = \mathbb{E}_{x,x' \leftarrow_R \{0,1\}^k, y \leftarrow_R \{0,1\}^{n-k}}[f(x \circ y) f(x' \circ y) \chi_{\alpha^0}(x) \chi_{\alpha^0}(x')]$$

By simple Chernoff bounds, one can estimate this expectation using repeated sampling. Thus, all that is left is to prove the lemma:

Proof. Before proving the lemma, let's have some intuition for why it works. Consider the case that $\alpha^0 = 0^k$ and suppose that $f_{\alpha^0} = 1$. This means that f can be expressed as a sum of functions of the form $\chi_{0^k \circ \beta}$ and hence it does not depend on its first k variables. Thus $f(x \circ y) = f(x' \circ y)$ and we'll get that $\mathbb{E}[f(x \circ y) f(x' \circ y)] = \mathbb{E}[f(z)^2] = 1$. If f_{0^k} is large then that means that in the Fourier representation, the weight of functions not depending on the first k variables is large and hence we expect large correlation between $f(x' \circ y)$ and $f(x \circ y)$. For the case $\alpha^0 \neq 0^k$, we essentially add these factors to translate it to the case $\alpha^0 = 0^k$. Indeed one can verify that if we define $g(x \circ y) = f(x \circ y) \chi_{\alpha^0}(x)$ then $g_{0^k} = f_{\alpha^0}$.

We'll now prove the lemma:

$$2^{-n-k} \sum_{x,x',y} f(x \circ y) f(x' \circ y) \chi_{\alpha^0}(x) \chi_{\alpha^0}(x') \underset{\text{basis change}}{=}$$

$$2^{-n-k} \sum_{x,x',y} \left(\sum_{\alpha \circ \beta} \hat{f}(\alpha \circ \beta) \chi_{\alpha \circ \beta}(x \circ y) \right) \left(\sum_{\alpha' \circ \beta'} \hat{f}(\alpha' \circ \beta') \chi_{\alpha' \circ \beta'}(x' \circ y) \right) \chi_{\alpha^0}(x) \chi_{\alpha^0}(x') \underset{\chi_{\alpha \circ \beta}(x \circ y) = \chi_\alpha(x) \chi_\beta(y)}{=}$$

$$2^{-n-k} \sum_{x,x',y} \left(\sum_{\alpha \circ \beta} \hat{f}(\alpha \circ \beta) \chi_\alpha(x) \chi_\beta(y) \right) \left(\sum_{\alpha' \circ \beta'} \hat{f}(\alpha' \circ \beta') \chi_{\alpha'}(x') \chi_{\beta'}(y) \right) \chi_{\alpha^0}(x) \chi_{\alpha^0}(x') \underset{\text{reordering terms}}{=}$$

$$\sum_{\alpha,\beta,\alpha',\beta'} \hat{f}(\alpha\beta) \hat{f}(\alpha'\beta') 2^{-k} \left(\sum_x \chi_{\alpha'}(x) \chi_{\alpha^0}(x) \right) 2^{-k} \left(\sum_{x'} \chi_\alpha(x') \chi_{\alpha^0}(x') \right) 2^{-(n-k)} \left(\sum_y \chi_\beta(y) \chi_{\beta'}(y) \right) \underset{\langle \chi_\alpha, \chi_{\alpha'} \rangle = \delta_{\alpha,\alpha'}}{=}$$

$$\sum_{\beta,\beta'} \hat{f}(\alpha \circ \beta) \hat{f}(\alpha' \circ \beta') \delta_{\alpha,\alpha^0} \delta_{\alpha',\alpha^0} \delta_{\beta,\beta'} = \sum_\beta \hat{f}(\alpha^0 \circ \beta)^2 = f_{\alpha^0}$$

□

Testing the long-code We call a function from $\{\pm 1\}^n$ to $\{\pm 1\}$ that depends on only a single variable (i.e. $f(x) = x_i$ or $f(x) = -x_i$) a *dictatorship*. We call a function that depends on at most a constant number of variables a *junta*.

The *long code* maps a value $i \in [n]$ to the function $x \mapsto x_i$ in $\{\pm 1\}^{\{\pm 1\}^n}$. That is, every codeword of the longcode is a dictatorship.

We'll now present a test T that when given oracle access to a longcode codeword f succeeds with probability $1-\rho$ (for some small $\rho < 0$), and if it succeeds with probability $1/2+\delta$ then there exists α with $|\alpha| \leq k = O(\log(1/\delta)/\rho)$ such that $\hat{f}(\alpha) \leq \delta$. Note that if we managed to get $k = 1$ then we'd get the analogous result to the linearity testing - that if we pass the test with significant advantage over half then there's a codeword with significant correlation. Here we get only that there's a linear junta with significant correlation (rather than a linear dictatorship) but it turns out to be sufficient for applications.

The test The test will be the following: pick $x, y \leftarrow_{\mathrm{R}} \{\pm 1\}^n$. Pick z according to the following distribution: $z_i = +1$ with probability $1-\rho$ and $z_i = -1$ with probability ρ. Then, check that $f(x)f(y) = f(x \oplus y \oplus z)$. Note that if $f(x) = x_i$ then with probability $1-\rho$, $z_i = 1$ in which case $f(x)f(y) = x_i y_i = x_i y_i z_i = f(x \oplus y \oplus z_i)$. For soundness, we prove the following lemma:

Lemma 6. *Suppose that* $\mathbb{E}_{x,y,z}[f(x)f(y)f(x \oplus y \oplus z)] \geq \delta$ *then*

$$\sum_{\alpha} \hat{f}(\alpha)^3 (1-2\rho)^{|\alpha|} \geq \delta$$

Proof.

$$\mathbb{E}_{x,y,z}[f(x)f(y)f(x \oplus y \oplus z)] = \sum_{\alpha,\beta,\gamma} \hat{f}(\alpha)\hat{f}(\beta)\hat{f}\mathbb{E}_x[\chi_\alpha(x)\chi_\gamma(x)]\mathbb{E}_y[\chi_\beta(y)\chi_\gamma(y)]\mathbb{E}_z[\chi_\gamma(z)] = \sum_{\gamma} \hat{f}(\gamma)^3 \mathbb{E}_z[\chi_\gamma(z)] = \sum_{\gamma} \hat{f}(\gamma)^3 (1-2\rho)^{|\gamma|}$$

where the last equality follows from the fact that

$$\mathbb{E}_z[\chi_\gamma(z)] = \mathbb{E}_z[\prod_{i \in \gamma} z_i] \underset{\text{independence}}{=} \prod_{i \in \gamma} \mathbb{E}_z[z_i] = \prod_{i \in \gamma} (+1(1-\rho) - 1\rho)$$

□

Additional reading Fourier analysis has turned out to be a very useful tool to approach problems from computer science. I've put some links on the web site to materials on these applications. Another point that we haven't touched is the relation of Fourier analysis to other notions discussed in the course such as eigenvalues and expansion. It turns out that the characters are exactly the eigenvectors of for the adjacency matrix of any Cayley graph on the group $\{\pm 1\}^n$ and that (up to normalization) the Fourier coefficients of the characteristic function the generating set are the eigenvectors.[1]

It also turns out that certain questions about the expansion of the Boolean cube as a graph (i.e., the graph where you connect $x, y \in \{\pm 1\}^n$ if their Hamming distance is one) can be answered using the tools of Fourier analysis (an example for such a question is which sets of half the vertices in this graph expand the least, and what can we say of sets that don't expand too much). The answers to such questions also turn out to be useful in computer science.

[1] If G is a group and $S \subseteq G$ then the Cayley graph $C(G,S)$ is the graph where we connect u, v if there's $s \in S$ such that $u = sv$. S is called the generating set of the graph: if S generates the group then the graph is connected, if $s \in S \iff s^{-1} \in S$ then the graph is undirected, and the degree of the graph is $|G|$.

We only discussed Fourier analysis over the group $\{\pm 1\}^n$ with componentwise multiplication (or equivalently the additive group $\mathsf{GF}(2)^n$, that has componentwise XOR as the operation). However, one can define the Fourier transform for any Abelian group G (i.e., group satisfying $ab = ba$ for all $a, b \in G$). For this we consider instead of functions χ from G to $\{\pm 1\}$, functions from G to $\mathbb{C}$ (or in fact the unit circle in $\mathbb{C}$) that satisfy $\chi(ab) = \chi(a)\chi(b)$. In fact, one can also define a generalization of the Fourier transform for non-Abelian group– this is called representation theory and involves moving from functions outputting a number $\mathbb{C}$ to functions outputting a higher dimensional matrix.

The web page also has links for additional reading on the PCP theorem. We haven't seen any of the non-trivial reductions from the PCP theorem to the hardness of approximating some **NP**-problems. We also haven't seen the stronger forms of **PCP** that are sometimes crucial for such applications: these include lower error **PCP**'s (parallel repetition, amortized free bit complexity) and lower queries **PCP**'s (two and three queries **PCP** with close to optimal error). There are also some open questions that are still open in the construction of **PCP**'s: one is whether there's a construction of a constant queries polynomial-sized **PCP** with polynomial relation between alphabet size and soundness error (the current constructions fail to work once the alphabet size is larger than $2^{(\log n)^{1-\epsilon}}$). Another is the construction of **PCP**'s with certain structural properties: one form of this question is the *unique games conjecture* which has received a lot of attention in recent years.