Notes for Lecture 19

1 Testing linearity of Boolean functions

As part of our preparation for an analysis of Håstad's proof system (next lecture), we need to consider the problem of testing whether a given string is a code word for some particular code. First, we will see a procedure that, given a string which may or may not be a code word for the Hadamard code, will decide if it is a code word or if it is far from being a code word. This is equivalent to a test that decides whether a Boolean function is linear or far from linear. That is, given $f : \{0,1\}^k \to \{0,1\}$,

- if f is linear, i.e. $f(x_1, \ldots, x_n) = \bigoplus_{i \in S} x_i$ for some $S \subseteq \{1, \ldots, n\}$, the test accepts with probability 1;
- if the test accepts with probability at least $1/2 + \delta$, then f has agreement at least $1/2 + \epsilon$ with a linear function.

Here is a possible test:

1. pick $x, y \in \{0,1\}^k$
2. accept if and only if $f(x) \oplus f(y) = f(x \oplus y)$.

We need to show that this test meets the requirements above. In order to analyze the test, we will introduce some concepts from Fourier analysis of Boolean functions.

1.1 Fourier analysis of Boolean functions

We are interested in linear functions, that is Boolean functions $l : \{0,1\}^k \to \{0,1\}$ such that there is some $a \in \{0,1\}^k$ for which $l(x) = a \cdot x$ for every x. There is a linear function l_S for every set $S \subseteq \{1, \ldots, k\}$ and it is defined as $l_S(x_1, \ldots, x_k) = \bigoplus_{i \in S} x_i$.

If $f : \{0,1\}^k \to \{0,1\}$ is a Boolean function, we denote by $F : \{-1,1\}^k \to \{-1,1\}$ the equivalent Boolean function in the representation where we use 1 instead of 0, -1 instead of 1, and multiplication $(\cdot)$ instead of addition $(\oplus)$. That is,

$$f(x_1, \ldots, x_k) = \frac{1}{2} - \frac{1}{2} F(1 - 2x_1, \ldots, 1 - 2x_k)$$

and

$$F(x_1, \ldots, x_k) = 1 - 2f\left(\frac{1}{2} - \frac{1}{2}x_1, \ldots, \frac{1}{2} - \frac{1}{2}x_k\right)$$

Note that if l_S is a linear function, then $L_S(x_1, \ldots, x_k) = \prod_{i \in S} x_i$ is the equivalent function in this correspondence.

Consider now the set of functions $F : \{-1,1\}^k \to \mathbb{R}$: this is a vector space of dimension 2^k over the reals, and it contains all boolean functions. Define the following scaled dot product over such functions:

$$F \cdot G = \frac{1}{2^k} \sum_{x \in \{-1,1\}^k} F(x)G(x)$$

Notice that if $F : \{-1,1\}^k \to \{-1,1\}$, i.e. if F is a Boolean function in this space, then $F \cdot F = 1$. Also notice that if $F, G : \{-1,1\}^k \to \{-1,1\}$ then

$$\mathbf{Pr}[f(x) = g(x)] = \frac{1}{2} + \frac{1}{2} F \cdot G$$

How do the linear Boolean functions behave with respect to this dot product? Consider two linear functions L_S and L_T, $S \neq T$:

$$\begin{aligned}
L_S \cdot L_T &= \frac{1}{2^k} \sum_{x \in \{-1,1\}^k} \left(\prod_{i \in S} x_i \right) \left(\prod_{j \in T} x_j \right) \\
&= \frac{1}{2^k} \sum_{x \in \{-1,1\}^k} \prod_{i \in S \Delta T} x_i \\
&= \mathbf{E}_{x \in \{-1,1\}^k} \prod_{i \in S \Delta T} x_i = \prod_{i \in S \Delta T} \mathbf{E}_x x_i = 0,
\end{aligned}$$

where $S \Delta T = (S \setminus T) \cup (T \setminus S)$ is the symmetric difference of S and T. To get the third line, we think of the sum as an expectation over $x \in \{-1,1\}^k$, where x is chosen randomly by setting the ith entry independently to ± 1, and therefore it is the expectation of a product of independent variables.

So all functions L_S are orthogonal to each other, that is, $L_S \cdot L_T = 0$ if $S \neq T$. Then all such functions must be linear independent, and since there are 2^k of them in a space of dimension 2^k they form an orthonormal basis. So we can write each function $F : \{-1,1\}^k \to \mathbb{R}$ as

$$F(x) = \sum_{S \subseteq \{1,\ldots,k\}} \hat{F}_S L_S(x).$$

This is known as the Fourier expansion of F. The Fourier coefficient $\hat{F}_S$ can be computed as $\hat{F}_S = F \cdot L_S$. Why would we write F like this? We'll see that the Fourier coefficients have a useful interpretation.

If F is Boolean, then

$$\begin{aligned}
\hat{F}_S &= F \cdot L_S \\
&= \mathbf{E}_x F(x) L_S(x) \\
&= \mathbf{Pr}_x[F(x) = L_S(x)] - \mathbf{Pr}_x[F(x) \neq L_S(x)] - 1 \\
&= 2\mathbf{Pr}_x[F(x) = L_S(x)] - 1 \\
&= 2\mathbf{Pr}_x[f(x) = l_S(x)] - 1.
\end{aligned}$$

Equivalently, for every S, we have $\mathbf{Pr}_x[f(x) = l_S(x)] = \frac{1}{2} + \frac{1}{2}\hat{F}_S$. The Fourier coefficients range between 1 and -1, and in a sense, they represent the bias that a Boolean function f has with respect to each of the linear functions. The coefficient $\hat{F}_S$ is 1 when f is exactly the corresponding linear function l_S, and it is -1 when f is exactly the complement of the linear function.

We will see that if the linearity test for a Boolean function f given at the beginning of the lecture accepts with high probability, then there is a large Fourier coefficient $\hat{F}_S$, which implies that there is a linear function l_S with nontrivial agreement with f. First, we need a couple of results about Fourier coefficients.

Lemma 1 (Parseval's Equality) *For every $F : \{-1,1\}^k \to \mathbb{R}$, $\mathbf{E}_x[F^2(x)] = \sum_S \hat{F}_S^2$.*

One way to interpret this is that the set of values of F and the Fourier coefficients in the expansion of F have the same l_2 norm up to a constant.

Corollary 2 *For every $F : \{-1,1\}^k \to \{-1,1\}$, $\sum_S \hat{F}_S^2 = 1$.*

Lemma 3 *Let $f : \{-1,1\}^k \to \{-1,1\}$. There are at most $1/(4\epsilon^2)$ linear functions l_S such that f and l_S have agreement at least $1/2 + \epsilon$.*

We have used the result in Lemma 3 before when we did worst case to average case reductions. We can easily prove it now using what we know about Fourier coefficients.

Proof: [Lemma 3] Suppose l_s and f have agreement at least $1/2+\epsilon$. Then $\hat{F}_S \geq 2\epsilon$, which implies that $\hat{F}_S^2 \geq 4\epsilon^2$, but from Corollary 2, we know that $\sum_S \hat{F}_S^2 = 1$. So there can be at most $1/(4\epsilon^2)$ functions l_S with that much agreement with f. $\square$

Proof: [Lemma 1 (Parseval's Equality)]

$$\begin{aligned}
\mathbf{E}_x[F^2(x)] &= \mathbf{E}_x\left[\left(\sum_S \hat{F}_S L_S(x)\right)\left(\sum_T \hat{F}_T L_T(x)\right)\right] \\
&= \sum_S \sum_T \hat{F}_S \hat{F}_T \mathbf{E}_x[L_S(x) L_T(x)] \\
&= \sum_S \sum_T \hat{F}_S \hat{F}_T L_S \cdot L_T \\
&= \sum_S \hat{F}_S^2.
\end{aligned}$$

$\square$

1.2 Using Fourier analysis to test linearity

Now consider the test to check if $f : \{0,1\}^k \to \{0,1\}$ is linear: pick at random $x, y \in \{0,1\}^k$ and accept iff $f(x) \oplus f(y) = f(x \oplus y)$. If f is linear then the test accepts with probability one.

Lemma 4 *If the test accepts f with probability p, then there is a linear function l_S that agrees with f on at least a p fraction of inputs.*

PROOF: First observe that

$$\mathbf{Pr}[\text{ test accepts } f\ ] = \frac{1}{2} + \frac{1}{2}\mathbf{E}_{x,y}[F(x)F(y)F(xy)].$$

This follows from the same type of argument we have made before, based on the fact that $F(x)F(y)F(xy)$ is 1 when the test accepts and -1 when it rejects:

$$\begin{aligned}
\mathbf{E}_{x,y}[F(x)F(y)F(xy)] &= \mathbf{Pr}_{x,y}[\text{ test accepts } f\ ] - \mathbf{Pr}_{x,y}[\text{ test rejects } f\ ] \\
&= 2\mathbf{Pr}_{x,y}[\text{ test accepts } f\ ] - 1.
\end{aligned}$$

Now, we need to prove that when $\mathbf{E}_{x,y}[F(x)F(y)F(xy)]$ is bounded away from zero, there is a large Fourier coefficient. To do this we just use the Fourier expansion of each function and simplify.

$$\begin{aligned}
\mathbf{E}_{x,y}[F(x)F(y)F(xy)] &= \mathbf{E}_{x,y}\left[\left(\sum_S \hat{F}_S L_S(x)\right)\left(\sum_T \hat{F}_T L_T(y)\right)\left(\sum_U \hat{F}_U L_U(xy)\right)\right] \\
&= \sum_S \sum_T \sum_U \hat{F}_S \hat{F}_T \hat{F}_U \mathbf{E}_{x,y}[L_S(x)L_T(y)L_U(x)L_U(y)] \\
&= \sum_S \sum_T \sum_U \hat{F}_S \hat{F}_T \hat{F}_U \left(\mathbf{E}_x L_S(x)L_U(x)\right)\left(\mathbf{E}_y L_T(y)L_U(y)\right) \\
&= \sum_S \sum_T \sum_U \hat{F}_S \hat{F}_T \hat{F}_U (L_S \cdot L_U)(L_T \cdot L_U) \\
&= \sum_S \hat{F}_S^3 \ \le\ \max_S \hat{F}_S \sum_S \hat{F}_S^2 \ =\ \max_S \hat{F}_S.
\end{aligned}$$

So the largest Fourier coefficient of F is an upper bound on the expectation $\mathbf{E}_x[F(x)F(y)F(xy)]$, and the expectation tells us how likely the test is to accept f. We have shown so far that for every S, $\mathbf{Pr}_x[f(x) = l_S(x)] = \frac{1}{2} + \frac{1}{2}\hat{F}_S$, and that $\mathbf{Pr}[\text{ test accepts } f\ ] \le \frac{1}{2} + \frac{1}{2}\max_S \hat{F}_S$. Therefore if $\mathbf{Pr}[\text{ test accepts } f\ ] \ge 1/2 + \epsilon$, then there exists an S such that $\hat{F}_S \ge 2\epsilon$, i.e. there is an S such that f and l_S have agreement on at least $1/2 + \epsilon$ fraction of inputs. $\Box$

Since not more than 1 in $4\epsilon^2$ functions can have that much agreement with f, if we do list decoding for f then we come up with a small list.

2 Testing code words for the Long Code

So far, we have been thinking of the encoding of a message $m \in \{0,1\}^k$ in the Hadamard code as a function l_S, where $S = \{i :\ m_i = 1\}$. An equivalent way to think of the encoding is as the concatenation of $a_1 \cdot m, a_2 \cdot m, \ldots a_{2^k} \cdot m$ where $a_1, \ldots, a_{2^k}$ is an enumeration of $\{0,1\}^k$. This is how we though of the code in worst case to average case reductions. It is also equivalent to think of the encoding as the concatenation of $l_{S_1}(m), l_{S_2}(m), \ldots, l_{S_{2^k}}(m)$, where $S_1, \ldots, S_{2^k}$ is an enumeration of subsets of B^k, that is, the encoding is the concatenation of the evaluation of all possible linear functions at m. In fact, for any code, it is possible to think of the encoding as the concatenation of the evaluation of a series of functions at the input value.

Let us now consider the error correcting code, called the Long Code, used in Håstad's proof system. One way to think of the encoding of $m \in \{1, \ldots, k\}$ in the Long Code is as a functional $M : (\{1, \ldots, k\} \to \{0, 1\}) \to \{0, 1\}$ that given a function $f : \{1, \ldots, k\} \to \{0, 1\}$ returns $M(f) = f(m)$. Equivalently, the encoding of m can be thought of as the concatenation of $f_1(m), f_2(m), \ldots, f_{2^k}(m)$ where $f_1, \ldots, f_{2^k}$ is an enumeration of *all* possible functions $f : \{1, \ldots, k\} \to \{0, 1\}$. Every error correcting code can be seen as a subset of this code, and no encoding can be longer than the Long Code unless two of the functions being evaluated are exactly the same.

Notice that the Long Code is doubly exponentially long: in order to transmit $\log k$ bits of information (since $m \in \{1, \ldots, k\}$), we need to send an encoding that takes 2^k bits. In Håstad's proof system, the code is only used on inputs of a constant size, so this blow-up does not cause a problem.

Another equivalent way to think of the encoding of m is as the concatenation of $a_1[m], a_2[m], \ldots, a_{2^k}[m]$ where $a_1, \ldots, a_{2^k}$ is an enumeration of $a \in \{0, 1\}^k$, that is, it is a concatenation of the mth entry in each string of length k. Equivalently, the encoding is a function that maps strings to one of their bits: the encoding of m is a function $M : \{0, 1\}^k \to \{0, 1\}$ that given an $x \in \{0, 1\}^k$ returns $M(x) = x[m]$. From this point of view, the Long Code encodes a message $m \in \{1, \ldots, k\}$ as a special linear function $l_{\{m\}}$. There is a loss of expressive power in this representation $l_{\{m\}}$ (note that it requires only $\log k$ bits of information), but it is convenient for our purposes.

2.1 Pre-Håstad verification

To get some understanding of how the Long Code test works in verification, we'll start by considering the verification model in vogue before Håstad's verifier. Given a function $A : \{0, 1\}^k \to \{0, 1\}$, we want the verifier to check that A is actually a code word for the Long Code as follows: for some predicate P, the verifier should check that there is some $a \in \{1, \ldots, k\}$ satisfying the predicate, $P(a) = 1$, such that given any string x, $A(x) = x[a]$, or in other words, A corresponds to some linear function $l_{\{a\}}$.

So suppose that after performing some test, we find that A has agreement $1 - \delta$ with some $l_{\{a\}}$.

Then define the vector $p \in \{0, 1\}^k$ such that $p[i] = P(i)$ for every i, and do the following test: choose a random $x \in \{0, 1\}^k$ and accept if and only if $A(p \oplus x) \oplus A(x) = 1$.

For a random x, $x \oplus f$ is also random, and so there is at least a $1 - 2\delta$ probability that both $A(x) = x[a]$ and $A(p \oplus x) = p[a] \oplus x[a]$. Thus, if $P(a) = 1$, the test accepts with probability at least $1 - 2\delta$, and if $P(a) = 0$ the test accepts with probability at least 2δ.

2.2 The Long Code test

Given $A : (\{1, \ldots, k\} \to \{0, 1\}) \to \{0, 1\}$,

- pick $f, g : \{1, \ldots, k\} \to \{0, 1\}$ uniformly

- pick $h : \{1, \ldots, k\} \to \{0, 1\}$ such that for every x

$$h(x) = \begin{cases} 0 & \text{with probability } 1 - \epsilon \\ 1 & \text{with probability } \epsilon \end{cases}$$

- accept iff $A(f) \oplus A(g) = A(f \oplus g \oplus h)$

Suppose that $A = l_{\{a\}}$. Then with high probability the test will accept since with high probability $f(a) \oplus g(a) = f(a) \oplus g(a) \oplus h(a)$. If A has no correlation with any linear function then this test is close enough to a test for linearity that the test will fail. If A is linear but *not* a code word for the Long Code , it cannot be defined over a very large set. Suppose $A = l_S$ for some set S. Then the probability that

$$\bigoplus_{x \in S} f(x) \bigoplus_{x \in S} g(x) = \bigoplus_{x \in S} f(x) \bigoplus_{x \in S} g(x) \bigoplus_{x \in S} h(x)$$

gets exponentially close to 0 with the cardinality of S.

Exercises

1. Show that if $f : \{0, 1\}^n \to \{0, 1\}$ is an arbitrary function, then $\max_S |\hat{F}_S| \geq 2^{-n/2}$.

2. Compute the Fourier coefficients of the function $f(x_1, x_2) = x_1 \vee x_2$.

3. Let $f : \{0, 1\}^n \to \{0, 1\}$ and $g : \{0, 1\}^m \to \{0, 1\}$ be arbitrary functions, and let $h : \{0, 1\}^{n+m} \to \{0, 1\}$ be defined as

$$h(x_1, \ldots, x_{n+m}) = f(x_1, \ldots, x_n) \oplus g(x_{n+1}, \ldots, x_{n+m})$$

   Compute the Fourier coefficients of h in terms of the Fourier coefficients of f and g.

4. Let n be even and consider the function

$$f(x_1, \ldots, x_n) = (x_1 \vee x_2) \oplus (x_3 \vee x_4) \oplus \cdots (x_{n-1} \vee x_n)$$

   Prove that $\max_S |\hat{F}_S| = 2^{-n/2}$.

5. Suppose that $f : \{0, 1\}^n \to \{0, 1\}$ is a function that depends only on its firts k variables, that is, suppose that there is a function $g : \{0, 1\}^k \to \{0, 1\}$ such that $f(x_1, \ldots, x_n) = g(x_1, \ldots, x_k)$. Compute the Fourier coefficients of f in terms of the Fourier coefficients of g.

6. Analyse the following linearity test: given oracle access to a function $f : \{0, 1\}^n \to \{0, 1\}$, pick uniformly at random four points $x_1, x_2, x_3, x_4 \in \{0, 1\}^n$ and accept if and only if

$$f(x_1) \oplus f(x_2) \oplus f(x_3) \oplus f(x_4) = f(x_1 \oplus x_2 \oplus x_3 \oplus x_4)$$