Lecture 14

Today we will prove the Goldreich-Levin theorem:

THEOREM 1
If $f : \{0,1\}^n \to \{0,1\}^n$ is a one-way permutation then $B(x,r) = \sum_i x_i r_i \pmod 2$ is a hardcore predicate for $f'(x,r) = f(x), r$.

In other words, given only $f(x)$ and r, it is hard to compute $\sum x_i r_i$. Before proving the above theorem, we establish some preliminary results.

1 Linear Boolean Functions on $\{0,1\}^n$

Consider a $2^n \times 2^n$ matrix M where each row and column is associated with an n-bit string: $a \in \{0,1\}^n$ and $x \in \{0,1\}^n$ respectively. Let the matrix element at position (a,x) be given by

$$M_{ax} = a \cdot x. \tag{1}$$

Then each row of this matrix corresponds to a linear function $f_a : \{0,1\}^n \to \{0,1\}$ where $f_a(x) = ax$. In fact, all 2^n possible linear boolean functions on $\{0,1\}^n$ are represented in the rows of this matrix.

LEMMA 2
For all $a \neq b, |\{x : ax \neq bx\}| = \frac{1}{2}2^n$.

PROOF:

$$\begin{aligned} |\{x : ax \neq bx\}| &= |\{x : ax \oplus bx = 1\}| && (2)\\ &= |\{x : (a \oplus b)x = 1\}| && (3)\\ &= \frac{1}{2}2^n && (4) \end{aligned}$$

where the last equality follows from symmetry and the fact that $a \oplus b \neq 0$. □

2 Error Correcting Codes (ECC)

The preceding lemma implies that any two rows of M differ in half of their values. This makes the rows of M well-suited as codewords for an error correcting code.

DEFINITION 1 *$\mathcal{C} \subseteq \{0,1\}^m$ is an error correcting code with minimum distance d if for any $y, z \in \mathcal{C}$, y and z differ in at least d positions.*

Here we have $\mathcal{C} \subseteq \{0,1\}^{2^n}$ with $d = 2^{n-1}$. This code is known as a Reed-Muller code or a Hadamard code.

The utility of such a code is that a sender (Alice) wishes to communicate with a receiver (Bob) over a possibly unreliable channel. To send the value $a \in \{0,1\}^n$, Alice sends to Bob the function $f_a(x) = ax$ (more precisely, she sends the value of f_a at all 2^n values of x). In other words, Alice sends row a of M which is 2^n bits long. Because the channel is unreliable, Bob receives the function $g(x)$ which may not be the same as $f_a(x)$. However, as we shall demonstrate shortly, provided that $g(x)$ and $f_a(x)$ agree on a fraction $\frac{3}{4}+\epsilon$ of the values in the domain, Bob can always recover $f_a(x)$ and therefore a.

Lemma 3
Suppose that in the above scenario f_a and g disagree on at most a fraction $\frac{1}{4}-\epsilon$ of the values in the domain. Then for all $b \neq a$, f_b and g disagree on at least $\frac{1}{4}$ of the values in the domain. Hence, a is uniquely determined and can be recovered.

Proof: Suppose by way of contradiction that there exists $b \neq a$ such that f_b and g disagree on less than $\frac{1}{4}$ of the values in the domain. Then f_a and f_b disagree on less than $\frac{1}{4}-\epsilon+\frac{1}{4} < \frac{1}{2}$ of the values in the domain. But this is impossible for $a \neq b$ by Lemma 2. □

2.1 Efficient Decoding

This leaves unanswered the question of how Bob can actually recover a from the function g. Naively, this can be done in time quadratic in 2^n by trying all functions f_a on all inputs x until the answer is found. The following theorem, however, indicates that a much more efficient algorithm exists.

Theorem 4
Let $g : \{0,1\}^n \to \{0,1\}$ be such that $g(x)$ agrees with $f_a(x)$ in $\frac{3}{4}+\epsilon$ of the x values. Then with oracle access to g, we can compute a in time $O(n \log n/\epsilon^2)$.

Proof: Let e_i be row i of the identity matrix. That is, $e_i = (00\ldots010\ldots00)$ where the 1 occurs in position i. Then for any x

$$
\begin{aligned}
a_i &= ae_i && (5)\\
&= ae_i \oplus ax \oplus ax && (6)\\
&= a(e_i \oplus x) \oplus ax && (7)
\end{aligned}
$$

Now consider the following operation: choose a random $x \in \{0,1\}^n$ and compute $g(e_i+x) \oplus g(x)$. With probability $\frac{3}{4}+\epsilon$, $g(e_i+x) = a(e_i+x)$ and with probability $\frac{3}{4}+\epsilon$, $g(x) = a(x)$. It then follows that both equalities hold with probability at least $\frac{1}{2}+2\epsilon$ and we conclude

$$\mathbf{Pr}_{x\in\{0,1\}^n}[g(e_i+x) \oplus g(x) = a_i] \geq \frac{1}{2} + 2\epsilon. \tag{8}$$

This suggests the following algorithm:

Algorithm $\frac{3}{4}+\epsilon$**:**
for $i := 1$ to n **do**
choose $k = O(\log n/\epsilon^2)$ points $x_1, \ldots, x_k \in \{0,1\}^n$
compute:
$g(x_1 + e_1) \oplus g(x_1)$
$g(x_2 + e_2) \oplus g(x_2)$
$\vdots$
$g(x_k + e_k) \oplus g(x_k)$
assign to a_i the value occurring in the majority of these computations

To analyze this program, note that for a particular value of i, we expect to get the right answer in a fraction $\frac{1}{2} + 2\epsilon$ of the k trials. Then by a Chernoff bound, the probability of estimating a_i incorrectly is $\exp -\Omega(\epsilon^2 k)$. We can then choose $k = O(\log n/\epsilon^2)$ such that this probability is bounded by $\frac{1}{n^2}$ and hence we conclude

$$\mathbf{Pr}[\text{any of the } a_i \text{ is incorrectly estimated}] \leq \frac{1}{n}. \tag{9}$$

We complete the proof by noting that the running time of this program is $O(nk) = O(n \log n/\epsilon^2)$. $\square$

2.2 Dealing With Noisy Channels

All of the above assumes that the received codeword g differs from the sent codeword f_a on less than one quarter of the values in the domain. If g differs from f_a on a faction greater than or equal to $\frac{1}{4}$ of the values in the domain then it is in general impossible to uniquely correct g to obtain f_a. In particular, there may be more than one value b for which the distance between f_b and g is a minimum. In such a situation, the best we can do is to find a list of codewords which disagree with g in no more than a specified number of locations. The size of such a list is bounded by the following theorem which we state without proof:

Theorem 5
For every $g : \{0,1\}^n \to \{0,1\}$ there are at most $\frac{1}{4\epsilon^2}$ values $a \in \{0,1\}^n$ such that $g(x)$ and $f_a(x)$ agree in at least fraction $\frac{1}{2} + \epsilon$ of their values.

We will shortly provide an algorithm to prove the following:

Theorem 6
Let $g : \{0,1\}^n \to \{0,1\}$ and ϵ be given. Then in $O(n \log n/\epsilon^4)$ time, we can construct a list of size $O(1/\epsilon^2)$ containing elements of $\{0,1\}^n$ such that the list contains every $a \in \{0,1\}^n$ such that $f_a(x)$ has agreement $\geq \frac{1}{2} + \epsilon$ with $g(x)$.

In view of theorem 5, we note that the list constructed according to theorem 6 is at most a constant factor larger than optimal.

PROOF: Fix a such that $g(x)$ and $f_a(x)$ have agreement at least $\frac{1}{2}+\epsilon$. Our goal is to reduce this problem to the $\frac{3}{4}+\epsilon$ case. First choose $x_1 \dots x_k \in \{0,1\}^n$ where $k = O(1/\epsilon^2)$. For the moment, let us suppose that we have "magically" obtained the values $ax_1, \dots, ax_k$. Then define $\hat{g}(z)$ as the majority value of:

$$(ax_i) \oplus g(z + x_i) \qquad i = 1, 2, \dots, k \tag{10}$$

Since for each i we obtain az with probability at least $\frac{1}{2}+\epsilon$, by choosing k sufficiently large we can ensure that

$$\mathbf{Pr}_{z,x_1,\dots,x_k}\left[\hat{g}(z) = az\right] \geq \frac{31}{32}. \tag{11}$$

from which it follows that

$$\mathbf{Pr}_{x_1,\dots,x_k}\left[\mathbf{Pr}_z\left[\hat{g}(z) = az\right] \geq 7/8\right] \geq \frac{3}{4}. \tag{12}$$

Consider the following algorithm:

Algorithm First-Attempt:
  choose $x_1, \dots, x_k \in \{0,1\}^n$ where $k = O(1/\epsilon^2)$
  for all $b_1, \dots, b_k \in \{0,1\}$
    define $\hat{g}_{b_1 \dots b_k}(z)$ as majority of: $b_i \oplus g(z + x_i)$
    apply Algorithm $\frac{3}{4}+\epsilon$ to uniquely decode $\hat{g}_{b_1 \dots b_k}$
    add result to list

The idea behind this program is that we don't in fact know the values of the ax_i so we guess all possibilities by considering all choices for the b_i. For each a such that f_a and g agree on more than half of their domain, we will eventually choose $b_i = ax_i$ for all i and then, with high probability, recover a via Algorithm $\frac{3}{4}+\epsilon$. The obvious problem with this algorithm is that its running time is exponential in $k = O(1/\epsilon^2)$ and the resulting list may also be exponentially larger than the $O(1/\epsilon^2)$ bound provided by Theorem 5.

To overcome these problems, consider the following similar algorithm:

Algorithm $\frac{1}{2}+\epsilon$**:**
  choose $x_1, \dots, x_t \in \{0,1\}^n$ where $t = O(\log(1/\epsilon))$
  for all $b_1, \dots, b_t \in \{0,1\}$
    define $\hat{g}_{b_1 \dots b_t}(z)$ as majority over all nonempty $S \subseteq \{1, \dots, t\}$ of: $(\oplus_{i \in S} b_i) \oplus g\left(z + \sum_{i \in S} x_i\right)$
    apply Algorithm $\frac{3}{4}+\epsilon$ to uniquely decode $\hat{g}_{b_1 \dots b_k}$
    add result to list

Let us now see why this algorithm works. First we define, for any nonempty $S \subseteq \{1, \dots, t\}$, $x_S \equiv \sum_{i \in S} x_i$. Then since $x_1, \dots, x_t \in \{0,1\}^n$ are random, it follows that for any $S \neq T$, x_S and x_T are independent and uniformly distributed. Now consider any a such that $f_a(x)$ and $g(x)$ agree on $\frac{1}{2}+\epsilon$ of the values in their domain. Then for the choice of $\{b_i\}$ where $b_i = ax_i$ for all i, we have that

$$\bigoplus_{i\in S} b_i = ax_S, \tag{13}$$

and, with probability $\frac{1}{2}+\epsilon$,

$$g\left(z+\sum_{i\in S} x_i\right) = g(z+x_S) \tag{14}$$

$$= a(z+x_s) = az + ax_S, \tag{15}$$

so combining the above results yields

$$\bigoplus_{i\in S} b_i + g\left(z+\sum_{i\in S} x_i\right) = az \tag{16}$$

with probability $\frac{1}{2}+\epsilon$.

Note the following simple lemma whose proof we omit:

LEMMA 7
Let $R_1,\dots,R_k$ be a set of pairwise independent $0-1$ random variables, each of which is 1 with probability at least $\frac{1}{2}+\epsilon$. Then $\mathbf{Pr}[\sum_i R_i \geq k/2] \geq 1 - O(1/\epsilon^2 k)$.

Lemma 7 allows us to upper-bound the probability that the majority operation used to compute $\hat{g}$ gives the wrong answer. Combining this with our earlier observation that the $\{x_S\}$ are pairwise independent, we see that choosing $t = 2\log 1/\epsilon + O(1)$ suffices to ensure that $\hat{g}_{b_1\dots b_t}(z) = az$ with probability $1-c \geq \frac{3}{4}+\epsilon$ for any constant $c>0$. Thus we can use Algorithm $\frac{3}{4}+\epsilon$ to obtain a with high probability. Choosing t as above ensures that the list generated is of length at most $2^t = O(1/\epsilon^2)$ and the running time is then $O(n\log n/\epsilon^4)$ (due to the $O(1/\epsilon^2)$ iterations of Algorithm $\frac{3}{4}+\epsilon$). This completes the proof of Theorem 6. $\Box$

3 Hardcore Predicates

We are now in a position to prove the Goldreich-Levin theorem stated at the beginning of these notes:

THEOREM 8
If $f:\{0,1\}^n \to \{0,1\}^n$ is a one-way permutation then $B(x,r) = \sum_i x_i r_i \pmod 2$ is a hardcore predicate for $f'(x,r) = f(x), r$.

PROOF: Suppose A is such that

$$\mathbf{Pr}_{x,r}[A(f(x),r) = xr] > \frac{1}{2}+\epsilon. \tag{17}$$

from which it follows that

$$\mathbf{Pr}_x \left[\mathbf{Pr}_r[A(f(x), r) = xr] \geq \frac{1}{2} + \frac{\epsilon}{2} \right] \geq \frac{\epsilon}{2}. \tag{18}$$

For a fixed x, define $g(r) \equiv A(f(x), r)$. With probability at least $\epsilon/2$ there is a linear function $f_x(r) = xr$ which agrees with $g(r)$ on at least $\frac{1}{2} + \epsilon$ of the values in the domain. Hence we can use the preceding algorithm on g with parameter $\epsilon/2$ to construct a list of size $O(1/\epsilon^2)$. By applying f to each item in the list, we can find a preimage x' of $f(x)$ (ie, $f(x') = f(x)$). Hence we are able to invert f for a fraction $\epsilon/2$ of its domain, contradicting the assumption that f is one-way. Thus $B(x, r)$ must be hardcore. $\square$