# Notes for Lecture 12

Today we see an alternative way of proving strong average-case hardness results, assuming we have as a starting point a function that is already somewhat hard on average. Part of these notes were written by Sanjeev Arora.

## 1 The XOR Lemma

**Definition 1 ($\delta$-hardness)** *Let $f$ be a boolean function on $n$-bit inputs, $\mathcal{D}$ a distribution on $n$ bit strings. If $\delta < 1/2$ and $g \in [n, \frac{2^n}{n}]$ then we say that $f$* is $\delta$-hard on distribution $\mathcal{D}$ for circuits of size $g$ *if for any boolean circuit $C$ with at most $g$ gates:*

$$\Pr_{x \in \mathcal{D}}[f(x) = C(x)] < 1 - \delta. \tag{1}$$

We want to transform $f$ into a function $f'$ that is very hard on average and that can be computed efficiently given $f$.

For a parameter $k$, consider the function $f^k : \{0,1\}^{nk} \to \{0,1\}$ defined as

$$f^k(x_1, \ldots, x_k) = f(x_1) \oplus f(x_2) \oplus \cdots \oplus f(x_k) \ .$$

Yao's "XOR-Lemma" states that the fraction on inputs on which $f^k$ can be computed by an efficient circuit tends to $1/2$ for large $k$. To see why this makes sense, let us first compute how successful we can be in computing $f^k$ if we just try to compute $f()$ on each input $x_i$ and the we xor the results. In each attempt we will be successful with probability at most $\delta$ (let us say exactly $\delta$ for simplicity). Then the probability of computing $f^k$ correctly is the same as the probability that we make an even number of errors. This is the same as defining independent 0/1 random variables $Z_1, \ldots, Z_k$, where for each $i$ $\mathbf{Pr}[Z_i = 0] = \delta$ and asking what is $\mathbf{Pr}[Z_1 \oplus \cdots \oplus Z_k = 0]$. We can think of each $Z_i$ as being sampled in the following equivalent way: with probability $1 - 2\delta$ we set $Z_i = 0$, and with probability $2\delta$ we flip an unbiased coin and let $Z_i$ be equal to the outcome of the coin. Then there is a probability $(1 - 2\delta)^k$ that when we sample $Z_1, \ldots, Z_k$ we never flip a coin, we then all $Z_i$ are zero and their xor is also zero; with the remaining $1 - (1 - 2\delta)^k$ probability we flip at least one coin, and the xor of the $Z_i$ contains the value of that coin xor-ed with other independent values, and so it is unbiased. Overall, we have

$$\mathbf{Pr}[Z_1 \oplus \cdots \oplus Z_k = 0] = (1 - 2\delta)^k + \frac{1}{2}(1 - (1 - 2\delta)^k) = \frac{1}{2} + \frac{1}{2}(1 - 2\delta)^k$$

that actually tends to $1/2$ for large $k$.

What happens when we consider arbitrary circuits for $f^k$? The following result by Impagliazzo allows us to get a proof of the XOR Lemma that follows quite closely the outline of the above probabilistic reasoning.

**Definition 2** *A set $H \subseteq \{0,1\}^n$ is a $\epsilon$-hard core set for a boolean function $f$ with respect to size $S$ if for any boolean circuit $C$ with at most $S$ gates:*

$$\Pr_{x \in S}[f(x) = C(x)] < \frac{1}{2} + \epsilon \tag{2}$$

**Lemma 1 (Impagliazzo)** *Suppose boolean function $f$ is $\delta$-hard for size $S$ on $n$-bit strings. Then for every $\epsilon > 0$ there is a set $S$ of size $\geq \delta 2^n$ such that $f$ is $\epsilon$-hard core on $S$ for circuits of size $S\epsilon^2\delta^2/4$.*

This is an extremely strong result: from the $\delta$-hardness of $f$ we only know that for every circuit $C$ there is a subset $H_C$ of $\delta 2^n$ inputs such that $C$ fails on $H_C$. Impagliazzo's result states that there is a single set $H$ of $\delta 2^n$ inputs such that every circuit $C$ fails on about half the elements of $H$. (Notice the different order of quantifiers.)

From Impagliazzo's result, it is easy to prove the XOR Lemma.

**Theorem 2 (Yao's XOR Lemma)** *Let $f$ be a boolean function on $n$-bit inputs that is $\delta$-hard for circuits of size $S$. Then the function $f^{\oplus k}$ on $nk$-bit inputs defined as*

$$f^{\oplus k}(x_1, x_2, \ldots, x_k) = f(x_1) \oplus f(x_2) \oplus \cdots \oplus f(x_k)$$

*is $(\frac{1}{2} - \epsilon - (1-\delta)^k)$-hard for circuits of size $S' = S\epsilon^2\delta^2/4$.*

PROOF: The proof is by contradiction. Assume some circuit $C$ of size at most $S'$ satisfies $\Pr[C(x_1, x_2, \ldots, x_k) = f^{\oplus k}(x_1, x_2, \ldots, x_k)] \geq 1/2 + \epsilon + (1-\delta)^k$.

By Impagliazzo's Lemma, there is an $\epsilon$-hard core set $H$ of $\delta 2^n$ inputs with respect to circuits of size $S'$. We construct a circuit $C'$ of size $S'$ that predicts $f$ on at least $1/2 + \epsilon$ fraction of inputs of $H$ and thus reach a contradiction.

Partition the set of $nk$-bit strings into $S_0, S_1, S_2, \ldots, S_k$, where $S_t$ consist of those $(x_1, \ldots, x_k)$ in which exactly $t$ of the $x_i$'s lie in $H$. At most $(1-\delta)^k$ fraction of inputs lie in $S_0$, so $C$ must predict $f^{\oplus k}$ with probability at least $1/2 + \epsilon$ on inputs from the other classes. Averaging shows there is a $t \geq 1$ such that $C$ predicts $f^{\oplus k}$ with probability at least $1/2 + \epsilon$ on inputs randomly chosen from $S_t$. Using nonuniformity, assume we know such a $t$.

Let $x$ be a random input from $H$; this is the input to $C'$. Pick $a_1, a_2, \ldots, a_{t-1}$ randomly from $H$ and $b_1, \ldots, b_{k-t-1}$ randomly from the complement of $H$. Randomly permute the $k$ inputs $a_1, a_2, \ldots, a_{t-1}, x, b_1, \ldots, b_{k-t-1}$ to get an $nk$-bit string $z$; note that $z$ is a random input from $S_t$. Apply $C$ on $z$. With probability at least $(1+\epsilon)/2$ we have $C(z) = f^{\oplus k}(z)$.

By averaging, there is a way to fix $a_1, a_2, \ldots, a_{t-1}, b_1, \ldots, b_{k-t-1}$ and their random permutation such that the circuit still outputs $f^{\oplus k}(z)$ with probability $1/2 + \epsilon$. Using nonuniformity again, we hardwire these choices into circuit $C$. Thus the circuit is left with $n$ input wires, into which $x$ is being fed. Noting that

$$f^{\oplus k}(z) = f(x) \oplus f(a_1) \oplus \cdots \oplus f(a_{t-1}) \oplus f(b_1) \oplus \cdots \oplus f(b_{k-t-1}),$$

we again use nonuniformity and assume we know $b = f(a_1) \oplus \cdots \oplus f(a_{t-1}) \oplus f(b_1) \oplus \cdots \oplus f(b_{k-t-1})$. We add a NOT gate at the top of our circuit iff $b = 1$. This gives a circuit $C'$

with $S'$ gates (remember we do not count NOT gates in circuit complexity) which on input $x$ produces $C(z) \oplus b$ as output. Furthermore, for at least $1/2 + \epsilon$ fraction of $x \in H$, the circuit outputs $f^{\oplus k}(z) \oplus b$, which is $f(x)$. Thus $C'$ computes $f$ on $1/2 + \epsilon$ of inputs from $H$, a contradiction. $\square$

## Exercises

1. Prove a version of Impagliazzo's Lemma for average-case algorithms that are errorless but may fail.

   We say that a circuit $C$ is errorless for a function $f : \{0,1\}^n \rightarrow \{0,1\}$ if $C(x) \in \{0, 1, fail\}$ for each $x$, and $C(x) = f(x)$ whenever $C(x) \neq fail$.

   We say that a function $f : \{0,1\}^n \rightarrow \{0,1\}$ is $\delta$-hard for errorles circuits of size $S$ if for every errorless circuit $C$ of size $\leq S$ for $f$ we have $\mathbf{Pr}_x[C(x) = fail] \geq \delta$.

   Prove that if $f : \{0,1\}^n \rightarrow \{0,1\}$ is $\delta$-hard for errorless circuits of size $S$, then there is a subset $H \subseteq \{0,1\}^n$, $|H| \geq \delta 2^n$ such that for every errorless circuit $C$ for $f$ of size $\leq S'$ we have $\mathbf{Pr}_{x \in H}[C(x) = fail] \geq 1 - \epsilon$, where $S' = \Omega(\epsilon S/(\log(1/\delta)))$.

2. Prove a "concatenation lemma" instead of a XOR Lemma. Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be $\delta$-hard for circuits of size $S$. Then argue that the function $f'\{0,1\}^{kn} \rightarrow \{0,1\}^k$ defined as $f'(x_1, \ldots, x_k) = (f(x_1), \cdots, f(x_k))$ cannot be computed on more than a $\epsilon + (1 - \delta)^k$ fraction of inputs by circuits of size $\epsilon^2 \delta^2 S/4$.