CS 880: Advanced Complexity Theory 25/2/2008

Lecture 14: Hardness Amplification

Instructor: Dieter van Melkebeek Scribe: Adeel Pervez

Last time we introduced a strategy for constructing a problem that is very hard in the average case assuming the existence of a problem in E that is hard only in the worst case. This strategy does not work for NP since the currently known techniques run in exponential time. Hardness amplification, however, can be done within NP. Here the goal is to construct a very average-case hard language under a 'mild' average-case hardness assumption. Today we develop some results regarding hardness amplification of general Boolean functions using the XOR Lemma. In the next lecture we will see details of hardness amplification within NP.

1 Hardcore Lemma

Recall that the Hardcore Lemma says that for every hard function there is a set of inputs on which it is very hard, in the sense that any circuit for the function, of upto a certain size, cannot do much better than to output a random bit.

Lemma 1. *If f is ϵ-hard for size s, then there exists $H \subseteq \{-1,1\}^n$ with $\mu(H) \geq \epsilon$ such that for all $\delta > 0$ and all circuits C of size at most $\Omega(\frac{\delta^2}{\log(1/\delta\epsilon)})s$,*

$$\Pr_{x \in H}[C(x) = f(x)] \leq \frac{1}{2} + \delta.$$

This lemma turns out to be important for the XOR Lemma which we study next.

2 XOR Amplification Lemma

Given a hard function f, the XOR Amplification Lemma allows us to construct a new function that is much harder than f.

The intuition behind the lemma is this: Consider an information-theoretic view of the Hardcore Lemma; namely, given an input in the hardcore set we do not know how to compute f on that input. Then if we take serveral copies of f and consider their aggregate with respect to some function like XOR, then it will be hard to compute as soon as the input of one of the copies of f belongs to the hard set. Thus, the new function has a larger set, in terms of relative size, on which it is hard to compute.

Let us quantify this intuition. Let $f^{\otimes k}$ denote k independent copies of f, i.e.,

$$f^{\otimes k} : (x_1, x_2, \ldots, x_k) \to (f(x_1), f(x_2), \ldots, f(x_k)).$$

For an input of $f^{\otimes k}$ to be outside the hardcore set, all x_i's ($1 \leq i \leq k$) have to be outside the hardcore set. Thus the size of $f^{\otimes k}$'s hardcore complement goes down exponentially with k. More precisely, if H and H' are f's and $f^{\otimes k}$'s hardcore sets then $\mu(H') \approx 1 - (1 - \mu(H))^k$, where $\mu(H)$ denotes the relative size of H. The function $f^{\otimes k}$ is not Boolean. Using an aggregator like XOR

(or product in $\{-1,1\}$) gives us a Boolean function. The resulting function is $g(x) = \prod_{i=1}^{k} f(x_i)$, which we denote by $g = \cdot \circ f^{\otimes k}$.

The above intuition interprets the notion of a hardcore in an information-theroretic sense, whereas the set H of Lemma 1 is only hardcore in a computational sense. We next analyze how the information-theoretic intuition translates into the computational setting.

Lemma 2. *If f is ϵ-hard for circuits of size at most s, then for every $\delta > 0$, $g = \cdot \circ f^{\otimes k}$ is ϵ'-hard for circuits of size at most s', where $\epsilon' = \frac{1}{2} - (1-\epsilon)^k - \delta$ and $s' = \Omega(\frac{\delta^2}{\log 1/(\epsilon\delta)})s$.*

Proof. We prove the lemma by considering the contrapositive. Suppose that g is not too hard. More precisely, suppose there exists a circuit C' of size at most s' that can compute g correctly on a fraction greater than $\frac{1}{2} + \delta'$, where $\delta' \geq \delta + (1-\epsilon)^k$:

$$\Pr_{x_1,x_2,\ldots,x_k}[C'(x_1,x_2,\ldots,x_k) = \prod_{1\leq i\leq k} f(x_i)] \geq \frac{1}{2} + \delta'. \tag{1}$$

From this we construct a circuit C that violates the conclusion of Lemma 1, thus showing that f is not ϵ-hard for circuits of size s.

Let $H \subseteq \{-1,1\}^n$ be nonempty. By conditioning on H, (1) implies

$$[1-(1-\mu(H))^k] \cdot \Pr_{\substack{x_1,x_2,\ldots,x_k \\ \text{at least one } x_i\in H}}[C'(x_1,x_2,\ldots,x_k) = \prod_{1\leq i\leq k} f(x_i)] + (1-\mu(H))^k \geq \frac{1}{2} + \delta',$$

where the probability is uniform over the tuples $(x_1,x_2,\ldots,x_k)$ that contain at least one component from H. Rearranging we get

$$[1-(1-\mu(H))^k] \cdot \Pr_{\substack{x_1,x_2,\ldots,x_k \\ \text{at least one } x_i\in H}}[C'(x_1,x_2,\ldots,x_k) = \prod_{1\leq i\leq k} f(x_i)] \geq \frac{1}{2} + \delta' - (1-\mu(H))^k.$$

We can generate the underlying distribution by picking $i \in [n]$ uniformly at random, then $x_i \in H$, and then $x_j \in \{-1,1\}^n$ for each $j \neq i$. By an averaging argument, we can fix everything except for the x_i in H and still maintain the inequality:

$$(\exists i \in [n])(\exists x_j, j \neq i) \Pr_{x_i\in H}[C'(x_1,\ldots,x_k) = \prod_{1\leq i\leq k} f(x_i)] \geq \frac{1}{2} + \delta' - (1-\mu(H))^k.$$

Hardwiring the choice of i and x_j, $j \neq i$, gives a circuit C of size at most s' such that

$$\Pr_{x\in H}[C(x) = f(x)] \geq \frac{1}{2} + \delta' - (1-\mu(H))^k.$$

This means that H cannot be a δ-hardcore for f and size s' if $\delta \leq \delta' - (1-\mu(H))^k$. Thus, f cannot have a δ-hardcore H of relative size $\mu(H) \geq \epsilon$ for circuits of size s' as long as $\delta \leq \delta' - (1-\epsilon)^k$. By Lemma 1, this means that f is not ϵ-hard for circuits of size s. □

3 Instantiations

In this section we present some instantiations of Lemma 2 obtained by fixing $\epsilon = 1/poly(n)$ and varying s. We present these results in terms of the notion of average-case hardness generally used in the context of derandomization.

Definition 1. *The hardness $H_g(m)$ of a function g at inputs of size m is that largest s' such that no circuit of size s' can satisfy*

$$\Pr_{|x|=m}[C'(x) = g(x)] \geq \frac{1}{2} + \frac{1}{s'}.$$

Assuming f has (worst-case) circuit complexity at least $s(n)$, we obtain the following average-case hardness results for g at length $m = k \cdot n$:

1. For $s(n) = n^{\omega(1)}$, $H_g(m) = n^{\omega(1)} = m^{\omega(1)}$.
2. For $s(n) = 2^{n^{\Omega(1)}}$, $H_g(m) = 2^{n^{\Omega(1)}} = 2^{m^{\Omega(1)}}$.
3. For $s(n) = 2^{\Omega(n)}$, $H_g(m) = 2^{\Omega(n)}$, which is at best $2^{\Omega(\sqrt{m})}$ since k needs to be $\Theta(n)$ for exponential hardness. This is still not good enough for full derandomization since for that we need the hardness to be $2^{\Omega(m)}$. It is possible to realize the latter by picking the $k = \Theta(n)$ inputs from a pseudo-random distribution that can be generated from $\Theta(n)$ truly random bits (rather than the $\Theta(k \cdot n)$ random bits needed for k independent uniform samples). However, we will not study these derandomization techniques in this course.

We point out that the transformation of f into g is very efficient. The techniques based on error-correcting codes which we studied in CS 810, take exponential time as they act on the entire truth-table of f at length n. In contrast, the our transformation runs in time polynomial in n (assuming k is polynomially bounded). Unfortunately, we cannot guarantee that g is in NP whenever f is. This is due to the fact that the XOR function is not monotone. We next consider combining function other than XOR, and in particular monotone ones.

4 Hardness Amplification for Balanced Functions

Let g be the following function

$$g = h \circ f^{\otimes k},$$

where f is balanced and h is an arbitrary function. In this section we analyze the hardness of such a function. In particular, we obtain a lower bound for the hardness of g in terms of a characteristic of h. A corollary of the result obtained here allows us to achieve our goal of amplification within NP by picking simple monotone functions within NP for h.

4.1 Analysis

Consider the distribution $x_1 x_2 \ldots x_k f(x_1) f(x_2) \ldots f(x_k)$. If some x_i belongs to the hardcore set, we can view $f(x_i)$ as random under our information-theoretic view of the Hardcore Lemma. To

simplify things a little, consider $xf(x)$. We replace $f(x)$ by the random variable F defined as follows:

$$F = \begin{cases} f(x) & \text{if } x \notin H \\ \text{random bit} & \text{otherwise} \end{cases}$$

For small circuits the distributions $xf(x)$ and xF look exactly the same since H is a δ-hardcore for f. More precisely, the ϵ-hardness of f for circuits of size s implies that f cannot be predicted on a fraction at least $\frac{1}{2} + \delta$ of H by circuits of size s'. By the connection between unpredictability and computational indistinguishability (see CS 810), this implies that for all circuits C of size at most s'

$$|\Pr[C(x, f(x)) = 1] - \Pr[C(x, F) = 1]| \leq \delta\mu(H) \leq \delta,$$

where the probability is over a uniform choice of $x \in \{-1, 1\}^n$ and over the randomness in F. For the distributions $x_1x_2 \ldots x_kf(x_1)f(x_2) \ldots f(x_k)$ and $x_1x_2 \ldots x_kF_1F_2 \ldots F_k$, this becomes

$$|\Pr[C(x_1, x_2, \ldots, x_k, f(x_1), f(x_2), \ldots, f(x_k)) = 1] - \Pr[C(x_1, x_2, \ldots, x_k, F_1, F_2, \ldots, F_k) = 1]| \leq k\delta, \tag{2}$$

where the circuit is now of size at most s' and F_i for $1 \leq i \leq k$ is defined analogously to F above.

Now let C' be a circuit of size at most $s' - size(h)$ computing h. Then,

$$|\Pr[C'(x_1, x_2, \ldots, x_k) = h(f(x_1), f(x_2), \ldots, f(x_k))] - \Pr[C'(x_1, x_2, \ldots, x_k) = h(F_1, F_2, \ldots, F_k)]| \leq k\delta. \tag{3}$$

This is true because otherwise we would be able to construct a circuit C of size at most s' that would violate (2).

We need an upper bound for $\Pr[C'(x_1, x_2, \ldots, x_k) = h(f(x_1), f(x_2), \ldots, f(x_k))]$, but since the two terms on the left-hand side of (3) are no more than $k\delta$ apart, it suffices prove an upper bound for the other term, i.e., for

$$\Pr[C'(x_1, x_2, \ldots, x_k) = h(F_1, F_2, \ldots, F_k)],$$

where the probability is over the uniform choice of $x_1, x_2, \ldots, x_k$ and the randomness in the F_i's for which $x_i \in H$.

Fix an input $x_1, x_2, \ldots, x_k$ and consider the restriction $R = (I, v)$ where $I = \{i \in [k] \,|\, x_i \notin H\}$ and v contains all function values $f(x_i)$ for $i \in I$. Since the F_i's for $i \notin I$ are just independent random bits and C' needs to agree with $h(F_1, F_2, \ldots, F_k)$, the best C' can possibly do on input $x_1, x_2, \ldots, x_k$ is to output the majority value of $h|_R$. Thus,

$$\begin{aligned} &\Pr[C'(x_1, x_2, \ldots, x_k) = h(F_1, F_2, \ldots, F_k)] \\ &\quad \leq \quad \mathrm{E}_R[\max(\Pr[h|_R = 0], \Pr[h|_R = 1])] \\ &\quad = \quad \mathrm{E}_R\left[\frac{\Pr[h|_R = 0] + \Pr[h|_R = 1]}{2} + \frac{|\Pr[h|_R = 0] - \Pr[h|_R = 1]|}{2}\right] \\ &\quad = \quad \frac{1}{2} + \frac{1}{2}\mathrm{E}_R[\mathrm{Bias}(h|_R)], \end{aligned}$$

where the expectation is over the distribution on R induced by a uniform choice of $x_1, x_2, \ldots, x_k$, and the bias of a function h with range $\{-1, 1\}$ is defined as

$$\mathrm{Bias}(h) = |\Pr[h = 1] - \Pr[h = -1]|.$$

Substituting this in (3) we get

$$\Pr[C'(x_1, x_2, \ldots, x_k) = h(f(x_1), f(x_2), \ldots, f(x_k))] \leq \frac{1}{2} + \frac{1}{2}\left[\mathrm{E}_R[\mathrm{Bias}(h|_R)]\right] + k\delta.$$

If f is balanced on $\overline{H}$ then the distribution of R is that of a random restriction with parameter $\rho = \mu(H)$. By the Hardcore Lemma f can be no more than δ away from balanced on H. By tweaking the parameters of the Hardcore Lemma a bit we can make sure f is perfectly balanced on H. Since we assumed f is balanced on the whole universe of inputs, this implies that f is balanced on $\overline{H}$. The above analysis thus gives the following lemma.

Lemma 3. *If f is balanced and ϵ-hard for circuits of size at most s then $g = h \circ f^{\otimes k}$ is ϵ'-hard for circuits of size at most s', where $\epsilon' = \frac{1}{2} - \frac{1}{2}\mathrm{E}_R[Bias(h|_R)]] - k\delta$ and $s' = \Omega(\frac{\delta^2}{\log(1/(\delta\epsilon))})s - size(h)$ and where R is a random restriction with parameter $\rho \geq \epsilon$.*

For h equal to the XOR, we have that $\mathrm{E}_R[\mathrm{Bias}(h|_R)] = (1-\rho)^k \leq (1-\epsilon)^k$. This instantion of Lemma 3 gives the XOR Amplication Lemma modulo some slight loss in parameters.

The above method does not apply to NP in general since the aggregated function might not be in NP. In particular, it does not apply in case of the XOR as the aggregation function. However, the aggregator function preserves membership in NP if it is monotone and in NP. More precisely, if $h \in \mathrm{NP}$ and monotone and $f \in \mathrm{NP}$ then

$$g = h \circ f^{\otimes k} \in \mathrm{NP}.$$

We leave this as an exercise to the reader.

5 Connection with Noise Sensitivity

We want ϵ' in Lemma 3 to be as close to $\frac{1}{2}$ as possible. Alternatively, we want $\frac{1}{2}\,\mathrm{E}_R[|h|_R|]$ to be small. The connection to Harmonic Analysis comes from the following relation

$$\mathrm{Bias}(h|_R) = |\widehat{h|_R}(\emptyset)|.$$

We also have the following

$$\mathrm{E}_R[|\widehat{h|_R}(\emptyset)|] \geq \mathrm{E}_R[(\widehat{h|_R}(\emptyset))^2]$$

and that

$$\mathrm{E}_R[|\widehat{h|_R}(\emptyset)|] \leq \sqrt{\mathrm{E}_R[(\widehat{h|_R}(\emptyset))^2]}, \tag{4}$$

where we have use the fact that $\mathrm{E}[X^2] \geq \mathrm{E}[X]^2$.

We will see next time that the right-hand side of (4) is related to the noise sensitivity of h. The precise relation is

$$\mathrm{E}_R[(\widehat{h|_R}(\emptyset))^2] = 1 - 2\mathrm{NS}_{\rho/2}(h),$$

where ρ is the restriction parameter. Due to this relationship our task reduces to finding a monotone $h \in \mathrm{NP}$ that has large noise sensitivity. This will be the topic for the next lecture.