Lecture 23: Hardness Amplification

Lecturer: Kristoffer Arnsfelt Hansen Scribe: Allan Rasmussen

Recap from Previous Two Lectures

In the previous lecture (Nisan-Wigderson PRG) we used the Nisan-Wigderson construction to construct a pseudorandom generator (PRG) from a language which is hard in the avarage-case. Recall that the **avarage-case hardness** of a language is measured by the quantity

$$\mathrm{H}_L(n) = \max\{S : \text{no circuit of size } S \text{ correctly computes } x \mapsto L(x) \text{ at a } 1/2 + 1/S \text{ fraction of the inputs } \{0,1\}^n\}. \quad (1)$$

If the average-case hardness $\mathrm{H}_L(n)$ for the language L has linear exponential growth, then the PRG constructed has linear exponential stretch. This is expressed by the following theorem.

Theorem 1. *Assume there exists a language $L \in \mathbf{E} = \mathbf{DTIME}(2^{\mathcal{O}(n)})$ such that*

$$\mathrm{H}_L(n) = 2^{\Omega(n)}. \quad (2)$$

Then there exists a PRG $G\colon \{0,1\}^{\mathcal{O}(\log m)} \to \{0,1\}^m$ (stretch $S(l) = 2^{\Omega(l)}$).

We also showed the (much easier) result that the existence of such a PRG with linear exponential stretch implies the existence of a language that require circuits of linear exponential size. Recall that a PRG may run in linear exponential time (the PRG G in the theorem may thus run in poly(m) time) and is characterized by that for any l and any circuit C of size at most $S(l)$,

$$\Big| \Pr[C(G(U_l)) = 1] - \Pr[C(U_{S(l)}) = 1] \Big| < 1/10. \quad (3)$$

Here U_l is the random variable uniformly distributed on $\{0,1\}^l$. In the previous but one lecture (Derandomization using PRGs) we showed that the existence of a PRG with linear exponential stretch implies that $\mathbf{BPP} = \mathbf{P}$.

Theorem 2. *If there exists a PRG G of stretch $S(l) = 2^{\Omega(l)}$ then* $\mathbf{BPP} = \mathbf{P}$.

Outline of proof (just because it's so simple). For the probabilistic TM M associated with a language $L \in$ **BPP** and input x of length n, we run for each seed $s \in \{0,1\}^{\mathcal{O}(\log(n))}$ $M(x)$ with random source $G(s)$. By (3) M cannot tell the distribution $G(U_{\mathcal{O}(\log(n))})$ from U_n, so about 2/3 of the answers will be right. By counting the answers we can thus (with certainty) tell whether $x \in L$. The running time is polynomial since the number of seeds is poly(n) (because of the exponential stretch). $\square$

$$
\begin{tikzcd}
\boxed{\exists\ L \in \mathbf{E} \text{ which requires circuits of size } 2^{\Omega(n)}} \arrow[d, Rightarrow, "\text{This lecture}"] & \\
\boxed{\exists\ L \in \mathbf{E} \text{ with } \mathrm{H}_L(n) = 2^{\Omega(n)}} \arrow[d, Rightarrow, "\text{Lecture: Nisan-Wigderson PRG}"] & \\
\boxed{\exists \text{ PRG } G \text{ of stretch } S(l) = 2^{\Omega(l)}} \arrow[r, Rightarrow, "\text{Lecture: Derandomization using PRGs}"'] & \boxed{\mathbf{P} = \mathbf{BPP}}
\end{tikzcd}
$$

Figure 1: Overview of lectures

Error-Correcting Codes

The purpose of this lecture is show the topmost implication of the above diagram. That is, we want to show that the existence of a language of worst-case hardness implies the existence of a language of (strong) average-case hardness. Error-correcting codes will be a tool for us and we start our study here.

The normalized **Hamming distance** between two strings $y, y' \in \Gamma^m$ over some alphabet Γ is defined to be the fraction of characters at which y and y' differ:

$$\Delta(y, y') = 1/m \cdot |\{i : y_i \neq y'_i\}|. \tag{4}$$

An **error-correcting code (ECC)** with distance $\delta > 0$ is a function $E\colon \Sigma^n \to \Gamma^m$ such that

$$x \neq x' \Rightarrow \Delta(E(x), E(x')) \geq \delta. \tag{5}$$

An ECC is trivially injective. The point is though, that for any $y \in \Gamma^m$, there can be at most one $x \in \Sigma^n$ such that $\Delta(y, E(x)) < \delta/2$. We can thus (correctly) decode a corrupted encoding as long as at most a $\delta/2$ fraction of the characters in the encoding are corrupted (this is only in principle - we actually want to be able to do this efficiently as well!).

Binary Codes

When the alphabets are $\{0,1\}$ the ECCs are referred to a being **binary**. Our first example of a binary ECC will be the **Walsh-Hadamard** encoding. The encoding function $\mathrm{WH}\colon \{0,1\}^n \to \{0,1\}^{2^n}$ maps x to $\mathrm{WH}(x)$, where

$$\mathrm{WH}(x)_\xi = x \cdot \xi \bmod 2. \tag{6}$$

Here for $y \in \{0,1\}^{2^n}$ and $\xi \in \{0,1\}^n$ we denote the ξ'th character (thinking of ξ as a binary number) of y by y_ξ. Analyzing the distance of the Walsh-Hadamards code, we first notice that the code is **linear** ($\mathrm{WH}(x + x') = \mathrm{WH}(x) + \mathrm{WH}(x')$), since

$$\mathrm{WH}(x + x')_\xi = (x + x') \cdot \xi \bmod 2 = x \cdot \xi + x' \cdot \xi \bmod 2 = \mathrm{WH}(x)_\xi + \mathrm{WH}(x')_\xi \bmod 2. \tag{7}$$

Keep in mind that we may add two strings from $\{0,1\}^n$ (and $\{0,1\}^{2^n}$) by adding each character/bit separately modulo 2 (that is, we take the XOR of the two strings). Assume now that $x \neq x'$. Then $\Delta(\mathrm{WH}(x), \mathrm{WH}(x'))$ is by definition equal to the fraction of characters at which $\mathrm{WH}(x)$ and $\mathrm{WH}(x')$ differ — or equivalently the fraction of non-zero characters in $\mathrm{WH}(x) - \mathrm{WH}(x')$. By linearity we have

$$\mathrm{WH}(x) - \mathrm{WH}(x') = \mathrm{WH}(x - x'). \tag{8}$$

Writing $x'' = x - x'$ we thus have

$$\begin{aligned} \Delta(\mathrm{WH}(x), \mathrm{WH}(x')) &= 1/2^n \cdot |\{\xi \in \{0,1\}^n : \mathrm{WH}(x'')_\xi \neq 0\}| \\ &= 1/2^n \cdot |\{\xi \in \{0,1\}^n : x'' \cdot \xi \bmod 2 \neq 0\}| \\ &= 1/2. \end{aligned} \tag{9}$$

The last equation follows since $x'' \neq 0$. So we may choose $\delta = 1/2$. This distance is actually the best you can hope for dealing with a binary code (see Arora&Barak). But Walsh-Hadamard is not very effective, since the encoded words has exponential size. The following proposition indicates that we should search for better ECCs than Walsh-Hadamard (even though the proof of the proposition is by a non-constructive, probabilistic argument).

Proposition 3 (Gilbert-Varshamov Bound)**.** *For every $0 < \delta < 1/2$ and n large enough there exists an ECC $E\colon \{0,1\}^n \to \{0,1\}^{n/1-H(\delta)}$ with distance δ, where $H(\delta) = \delta \log 1/\delta + (1-\delta)\log 1/(1-\delta)$ is the Shannon entropy function.*

The encoding stated in the proposition only gives a linear increase in size (even though the increase factor tends to infinity for δ close to $1/2$).

Explicit Codes

We are mainly interested in ECCs where the encoding/decoding can be done efficiently by an algorithm, since we will implement encoding/decoding in circuits. This is why the above proposition is of little help to us. Our approach is to construct explicit codes where Walsh-Hadamard is our first example. Two additional examples follow.

Let $\mathbb{F} = \{\alpha_0, \alpha_1, \dots\}$ be a finite field and $n \leq m \leq |\mathbb{F}|$. The **Reed-Solomon** code $\mathrm{RS}\colon \mathbb{F}^n \to \mathbb{F}^m$ maps $a = (a_0, \dots, a_{n-1})$ to $\mathrm{RS}(a)$, where (for $j = 0, \dots, m-1$)

$$\mathrm{RS}(a)_j = A(\alpha_j), \quad A(x) = \sum_{i=0}^{n-1} a_i x^i. \tag{10}$$

The Reed-Solomon code is linear like Walsh-Hadamard, so for $a \neq a'$, if we write $a'' = a - a'$ and $A(x) = \sum_{i=0}^{n-1} a''_i x^i$, we have

$$\begin{aligned} \Delta(\mathrm{RS}(a), \mathrm{RS}(a')) &= 1/m \cdot |\{j = 0, \dots, m-1 : \mathrm{RS}(a'')_j \neq 0\}| \\ &= 1/m \cdot |\{j = 0, \dots, m-1 : A(\alpha_j) \neq 0\}|. \end{aligned} \tag{11}$$

Since $a'' \neq 0$ the polynomial A is non-zero. Since the degree of A is less than $n-1$, it has at most n roots, so

$$|\{j = 0, \dots, m-1 : A(\alpha_j) \neq 0\}| \geq m - n. \tag{12}$$

So we may choose $\delta = 1 - n/m$.

The **Reed-Muller** code is based on a finite field $\mathbb{F}$, $d < |\mathbb{F}|$, and a free parameter l. The encoding function RM: $\mathbb{F}^{\binom{d+l}{d}} \to \mathbb{F}^{|\mathbb{F}|^l}$ maps c to RM(c), where

$$\mathrm{RM}(c)_{x_1,\ldots,x_l} = P(x_1,\ldots,x_l), \quad P(x_1,\ldots,x_l) = \sum_{i_1+\cdots+i_l \le d} c_{i_1,\ldots,i_l} x_1^{i_1} \cdots x_l^{i_l}. \tag{13}$$

The analysis of the distance is much like that for Reed-Solomon. The number of roots in $P \neq 0$ is estimated by the Schwartz-Zippels lemma (Lemma A.36 in Arora&Barak). The distance is $\delta = 1 - d/|\mathbb{F}|$.

The three constructed codes either are not binary (RS and RM) or has an exponential blowup in size (WH). Concatenated codes will allow us to remedy this.

Concatenated Codes

Given two ECCs

$$\begin{aligned} E_1 &\colon \Sigma^n \to \Gamma^m, \\ E_2 &\colon \Omega^l \to \Delta^k, \end{aligned} \tag{14}$$

with $|\Gamma| \le |\Omega|^l$, and where Γ is identified with a subset of Ω^l, we may construct the concatenation $E = E_2 \circ E_1 \colon \Sigma^n \to \Delta^{mk}$. The ECC E is defined by

$$E(x) = E_2(E_1(x)_1) \cdots E_2(E_1(x)_m). \qquad \text{(concatenation)} \tag{15}$$

Lemma 4. *Assume that E_1 and E_2 have distances respectively δ_1 and δ_2. The distance of $E = E_2 \circ E_1$ is then $\delta_1\delta_2$.*

Proof. Assume $x \neq x'$. At least a δ_1 fraction of the characters of $E_1(x)$ and $E_1(x')$ differ. Let j be the index of a character that differs. Then at least a δ_2 fraction of the characters of $E_2(E_1(x)_j)$ and $E_2(E_1(x')_j)$ differ. So at least a $\delta_1\delta_2$ fraction of the characters of $E(x)$ and $E(x')$ differ. □

As an example we may take WH: $\mathbb{F} = \{0,1\}^{\log|\mathbb{F}|} \to \{0,1\}^{|\mathbb{F}|}$ and RS: $\mathbb{F}^n = \{0,1\}^{n\log|\mathbb{F}|} \to \mathbb{F}^m$. The concatenation WH $\circ$ RS: $\{0,1\}^{n\log|\mathbb{F}|} \to \{0,1\}^{m|\mathbb{F}|}$ has distance $(1 - n/m)/2$. Notice that fixing the fraction m/n gives a fixed distance (arbitrarily close to 1/2) and only a linear increase in code size. This is a nice binary ECC but the next section explains why we eventually must use a composition of Walsh-Hadamard and Reed-Muller.

Local Decoding

All our three explicit codes may be decoded efficiently. But we need decoders decoding only a single character of the encoded string and running in polylogarithmic time — clearly this timebound restricts them to read only a small part of the codeword!

Definition 5. Let $E\colon \Sigma^n \to \Gamma^m$ be an ECC. A **local decoder** for E error-rate ρ is a probabilistic algorithm running in polylog(m) time, such that when given an index $0 \le i < n$ and random access to a string y, such that $\Delta(y, E(x)) \le \rho$, the algorithm outputs x_i with probability at least 2/3.

Clearly, for a local decoder to exist ρ needs to be smaller than half the distance for the ECC E (else x may not be unique). Recall that the distance δ for WH: $\{0,1\}^n \to \{0,1\}^{2^n}$ is 1/2. Let $\rho < 1/4$ be given. We will now describe a local decoder for WH handling ρ errors.

The input is $0 \le i < n$ and random access to $f\colon \{0,1\}^n \to \{0,1\}$ with

$$\Pr_{\xi}[f(\xi) \neq x \cdot \xi \bmod 2] \le \rho \tag{16}$$

The decoder will choose ξ uniformly random in $\{0,1\}^n$ and output $f(\xi) + f(\xi + e^i) \bmod 2$, where $e^i \in \{0,1\}^n$ is the string with the i'th bit being 1 and all the other bits 0. With probalility at least $1 - 2\rho$ both $f(\xi)$ and $f(\xi + e^i)$ are 'correct', in which case

$$f(\xi) + f(\xi + e^i) \bmod 2 = x \cdot \xi + x \cdot (\xi + e^i) \bmod 2 = x \cdot e^i \bmod 2 = x_i. \tag{17}$$

Since $1 - 2\rho > 1/2$ we may by repetition of the algorithm output x_i with probability greather than 2/3.

For appropriately choosen parameters Reed-Muller has a local decoding algorithm (Reed-Solomon does not). Also, concatenating two codes, both having a local decoder, gives an ECC which has a local decoder. This gives at least an indication that a concatenation of Walsh-Hadamard and Reed-Muller may be used as the ECC needed in the next section.

Hardness Amplification

Theorem 6. *Let $E\colon \{0,1\}^N \to \{0,1\}^{N^c}$ be a family of binary ECCs with local decoders handling error-rate ρ. Assume there exists $f \in \mathbf{E}$ requiring circuits of size $S(n)$ (**worst-case hardness**). Then there exists $\hat{f} \in \mathbf{E}$, such that no circuit of size $S(m/c)/m^{\mathcal{O}(1)}$ correctly computes $\hat{f}$ at a $1-\rho$ fraction of the inputs of length m (**mild average-case hardness**).*

Proof. Given n let $N = 2^n$ and $m = \log N^c = cn$. Let f_n denote the mapping $\{0,1\}^n \ni x \mapsto f(x)$. Define $\hat{f}_m = E(f_n)$ where we regard f_n and $\hat{f}_m$ as truth tables. So $\hat{f}_m$ is a function from $\{0,1\}^m$ to $\{0,1\}$. The function $\hat{f}\colon \{0,1\}^* \to \{0,1\}$ will be defined by the sequence $\hat{f}_m$. We can make sure that $\hat{f}$ is defined for all input length by padding inputs which are not of length cn for some n.

First, computing $\hat{f}_m(y)$ for some m and $y \in \{0,1\}^m$ amounts to computing $\hat{f}_m = E(f_n)$. Since E runs in polynomial time and f_n has size 2^n this takes time $2^{\mathcal{O}(n)} = 2^{\mathcal{O}(m)}$. So $\hat{f} \in \mathbf{E}$.

Assume now that C is a circuit of size $T(m)$ computing $\hat{f}_m$ correctly on a $1-\rho$ fraction of the inputs. The function f_n may then, on input x, be computed correctly with probability 2/3 by running the local decoder for E with input index x and random access to $E(f_n) = \hat{f}_m$ through the circuit C. The decoder runs in time $\operatorname{polylog}(N^c) = \operatorname{poly}(m)$. By error reduction and fixing random bits (see proof of $\mathbf{BPP} \subseteq \mathbf{P_{/poly}}$) we get a circuit of size $m^{\mathcal{O}(1)}T(m)$ computing f for all inputs. So

$$S(n) \le m^{\mathcal{O}(1)} T(m) \tag{18}$$

which implies

$$S(m/c)/m^{\mathcal{O}(1)} \le T(m). \tag{19}$$

□

The needed binary ECC exists at least for $\rho = 0.01$ (see Arora&Barak). But the proof cannot supply a language of **strong average-case hardness** as required in Theorem 1. This is because $\rho < 1/4$ for all binary local decoders. Getting to strong average-case hardness is accomplished by using **list decoders** instead of just decoders. A list decoder will on input y output the list of x's with $\Delta(y, E(x)) \leq \rho$. The size of this list is bounded by the following theorem.

Theorem 7 (Johnson bound)**.** *If $E\colon \{0,1\}^n \to \{0,1\}^m$ is a binary ECC with distance $\rho = 1/2 - \epsilon$, then for every $y \in \{0,1\}^m$, and $\delta \geq \sqrt{\epsilon}$, there exist at most $1/(2\delta^2)$ codewords $y_1, \ldots, y_l$ such that $\Delta(y, y_i) \leq 1/2 - \delta$ for every $i = 1, \ldots, l$.*

Using binary list decoders we may handle error-rate ρ for any $\rho < 1/2$. A **local list decoder** works like a local decoder but additionally takes an index to list of possible codewords as input. Hardwiring the right index into the circuit computing f_n we may prove the following theorem.

Theorem 8. *Assume $f \in \mathbf{E}$ requires circuits of size $S(n)$ (worst-case hardness). Then there exists $\hat{f} \in \mathbf{E}$ and c such that $\mathrm{H}_g(m) \geq S(m/c)^{1/c}$ for sufficiently large m (strong average-case hardness).*