CS 880: Pseudorandomness and Derandomization 4/22/2013

Lecture 24: Hardness Amplification

Instructors: Holger Dell and Dieter van Melkebeek Scribe: Xi Wu

DRAFT

In the last two lectures we show how to translate average-case hardness and further worst-case hardness for *circuits*, via Nisan-Wigderson generator, to pseudorandomness. Let's first review the parameters involved. In a PRG $G : \{0,1\}^d \mapsto \{0,1\}^r$, d is a function of r and error ε. Assume a function $g : \{0,1\}^m \mapsto \{0,1\}$ in E of average hardness $\mathsf{H}_g(m) = s$, and an (r,a)-design $S_1, \ldots, S_r$ ($|S_i| = m$ and $|S_i \cap S_j| \leqslant a$). Given a seed $\sigma \in \{0,1\}^d$, Nisan-Wigderson generator outputs

$$g(\sigma|_{S_1}) \circ g(\sigma|_{S_2}) \circ \cdots \circ g(\sigma|_{S_r})$$

By an unpredictability argument, we show that this generator fools circuits of size $s - r2^a$ with error ε satisfying $\varepsilon/r = 1/s$. Now, with a design where $a = O(\log r)$ and $d = O(m^2/a)$, we can bound seed length d using r, ε as follows,

$$m = \mathsf{H}_g^{-1}(s) = \mathsf{H}_g^{-1}(r/\varepsilon) \qquad \text{by average hardness}$$

$$d = O\left(\frac{\mathsf{H}_g^{-1}(r/\varepsilon)^2}{\log r}\right) \qquad \text{by combinatorial design}$$

Let's consider a simple instantiation: suppose $\mathsf{H}_g(m) = 2^m$ and $\varepsilon = 1/r$, then $\mathsf{H}_g^{-1}(r/\varepsilon) = \log r/\varepsilon = O(\log r)$, and the seed length $d = O(\log r)$, which is optimal in r up to constant.

For worst-case hardness, we start with $f : \{0,1\}^n \mapsto \{0,1\}$ of *worst-case hardness* $\mathsf{C}_f(n) = s$. The idea is to encode χ_f using a binary code $\mathrm{Enc} : \{0,1\}^N \mapsto \{0,1\}^M$ [1] so that the resulting binary sequence gives the characteristic sequence of a function g that is *average-case hard*, and then plug g into Nisan-Wigderson generator. The intuition here is that an efficient decoding algorithm, with oracle access to a small circuit $h : \{0,1\}^m \mapsto \{0,1\}$ that computes $\mathrm{Enc}(\chi_f)$ somewhat well on average, enables reconstructing f correctly everywhere.

As discussed in the last lecture, the notion of decoding we need is *locally list-decoding*: given decoding radius $1/2 - \delta$ (note that this can be very close to the limit of decoding radius of binary code). we need that for every n, there is a list of circuits $D_1, D_2, \ldots$ of size s' so that for any $f : \{0,1\}^n \mapsto \{0,1\}$, if $\mathrm{agr}(\mathrm{Enc}(\chi_f), \chi_h) \geqslant 1/2 + \delta$, then there is a circuit D_i so that D_i^h computes f correctly everywhere, and this is a contradiction if the size of D_i^h is less than s. Note that $|D_i^h| = s' \cdot |h|$ where $|h|$ denotes the size of h. Therefore, this argument translates worst-case hardness of size s to $(s/s', \delta)$-average-case hardness (generally, a function f at length n is (s, ε)-average-case hard if any circuit of size s can correctly compute at most $1/2 + \varepsilon$. In the argument above, we are interested in $(s, 1/s)$-average-case hard).

This procedure of amplifying worst-case hardness to average-case hardness is known as *hardness amplification* in literature. In our setting, we are amplifying the hardness against non-uniform complexity class: namely circuits. In this lecture we elaborate the details of this amplification with a focus on how the local list-decoding works.

[1] As usual, capital letter denotes powering: $N = 2^n$, etc.

1 Hardness Amplification

In this section, we prove the following quantitative version of the above discussion

Theorem 1. *There exists* $\mathrm{Enc}_{n,s} : \{0,1\}^N \mapsto \{0,1\}^M$ *where* m *is a function of* n *and* s *such that*

(1) $\mathrm{Enc}_{n,s}$ *is computable in time* $\mathrm{poly}(N, s)$.

(2) For every n, s *there exists a list of circuits* $D_1, D_2, \ldots$ *of size* $\mathrm{poly}(n, s)$ *such that for any* $h : \{0,1\}^m \mapsto \{0,1\}$ *and* $f : \{0,1\}^n \mapsto \{0,1\}$ *such that if*

$$\delta_H\left(\mathrm{Enc}(\chi_f), \chi_h\right) \leqslant \frac{1}{2} - \frac{1}{s}$$

then there exists i *such that* D_i^h *computes* f *correctly everywhere.*

1.1 High Level Ideas

Our main construction is to concatenate an outer code with distance $1 - O(1/s)$ and is locally list decodable up to distance $1 - O(1/s)$ with Hadamard code as inner code. Let's focus on the outer code. A natural first try is Reed-Solomon code. Consider Reed-Solomon code of degree d over $\mathbb{F}_q$. We identify every information word in $\mathbb{F}_q^{d+1}$ as the values of univariate polynomial P of degree at most d in $d+1$ distinct positions. The encoding of P is its values at all positions. However, Reed-Solomon is not locally list decodable. The rough reason is that, given a received word of size $2^{O(n)}$, we can only afford to look at poly-log(n) positions, however these few positions give little information for local decoding. For example, suppose that we want to decode at $x \in \{0,1\}^n$, looking at any d positions leave x to be equally likely to be any symbol in $\mathbb{F}_q$, which is useless.

Instead we use *low-degree extension* and Reed-Muller codes (how it overcomes the local list decoding issue will come out later). Precisely we identify a d-variate polynomial P of degree $\ell - 1$ in each variable by its evaluation at each point of a ℓ^d (which we set to be the length of the information word, that is $N = \ell^d$) cube. We claim that the values on this cube uniquely determines the polynomial give the inductive proof. Note that first we extend an information word of length ℓ^d to q^d, and second, the total degree is bounded by $d(\ell - 1)$ so the relative minimum distance, by Schwartz-Zippel Lemma is at least $1 - \frac{d\ell}{q}$, and we set $d\ell/q = \Theta(1/s)$.

The way we decode is where the improvement comes. Precisely, in order to decode at x, we randomly restrict the received word to a line L that passes through x. We do this by uniformly pick a point y, form line $L(t) : \mathbb{F}_q \mapsto \mathbb{F}_q^d$ by $x + ty$. It is important to observe that restricting to L gives a univariate polynomial in t of degree at most $d\ell$.

Now consider decoding at radius $1 - O(d\ell/q)$. Because each point on the line is uniformly distributed in the cube $\mathbb{F}_q^d$, the expected fraction of points that agree with $P|_L$ is $\Omega(d\ell/q)$. Because points on a line are pairwise-independent explain this, so we can use Chebyshev bound to show that actually this holds with high probability. Therefore decoding restricting to this line suffices! But now this is exactly list decoding Reed-Solomon of degree at most ℓd, and we only need to query about $d\ell$ position, which could be much smaller than ℓ^d.

1.2 List-Decoding of Reed-Solomon Code

In this section we give a list-decoding algorithm for Reed-Solomon code which is first discovered by Guruswami and Sudan. We identify each information word $f \in \mathbb{F}_q^{d+1}$ as evaluations of a polynomial

of degree at most d at $d+1$ points. The encoding is the evaluation of f at every point. On input $r \in \mathbb{F}_q^q$, we first find a nonzero bivariate polynomial $Q(Y, X)$ of degree d_Y in Y and degree d_X in X, so that for every $y \in \mathbb{F}_q$ (so we have in total q equations)

$$Q(y, r(y)) = 0$$

There are $(d_Y + 1)(d_X + 1)$ coefficients in Q, so nonzero Q exists provided that

$$(d_Y + 1)(d_X + 1) > q \tag{1}$$

Now we argue that for any polynomial g of degree at most d, if $\mathrm{agr}(\mathrm{Enc}(f), g) \geqslant \varepsilon$, then $Q(Y, g(Y)) \equiv 0$. This is true provided that the number of inputs we vanish (εq) is larger than the degree of $Q(Y, g(Y))$, so gives condition

$$\varepsilon q > d_Y + d \cdot d_X \tag{2}$$

Therefore now if we consider univariate polynomial $Q^*(X)$ by viewing Q as polynomial in $(\mathbb{F}_q[Y])[X]$, then the g's we are seeking for are all in the factorization of Q^*. The size of the list is bounded by the degree of Q^*, which is at most d_X

Now we set parameters, we are interested in maximizing the decoding radius $1 - \varepsilon$, while minimizing the list size d_X. This gives that ε, d_Y, d_X as function of d, q. Set $d_Y = \lceil \sqrt{dq} \rceil$ and $d_X = \lceil \sqrt{q/d} \rceil$, we have that (1) is satisfied, and to satisfy (2),

$$\varepsilon q > 2\sqrt{qd}$$

Therefore set $\varepsilon \approx 2\sqrt{d/q}$. Note that the minimum distance of Reed-Solomon code is $\delta = d/q$, where $R = d/q$ is the rate of the code. therefore our decoding radius is $1 - O(\sqrt{R})$. Going beyond $1 - \sqrt{R}$ for $R < 1/16$ is a long-standing open problem (which is known as the Guruswami-Sudan radius), which is finally resolved by Parvaresh-Vardy codes (as we saw in previous lectures).

1.3 Local List-Decoding of Reed-MüDer codes

Let's delve into the parameters. elaborate the details.

Setting parameters we have

$$d = \Theta\left(\frac{\log N}{\log \log N}\right)$$
$$\ell = \Theta(d \cdot s)$$
$$q = \Theta((ds)^2)$$

2 Generalization to Other Models

Our argument for general circuits (both average-case hardness and worst-case hardness arguments) carry through for branching programs. We leave this as an exercise to read to verify. One immediate instantiation of this hardness-randomness tradeoff is that if there is a function $f_m \in \mathrm{DSPACE}(O(m))$ requires branching programs of linear exponential size, then BPL = L. In fact we get that BPL with two-way access to the random tape equals L.

3 On Fooling Constant-depth Circuits

Our argument for general circuits also applies to constant-depth circuits. However, in this case, we only know that the average-case hardness argument could carry through because we do not know whether the list decoding algorithm is computable by small depth circuits. Fortunately, average-case hardness for these circuits are known, and so we have unconditional pseudorandom generator for constant-depth circuits.

Theorem 2. *If f can be computed by a circuit of depth d and size s, for every $\Delta > 0$, there exists a multivariate polynomial $P : \{0,1\}^m \mapsto \mathbb{R}$ of total degree at most Δ such that*

$$\mathbb{E}_{x \leftarrow U_n}\left[\left|f(x) - P(x)\right|^2\right] \leqslant 2 \cdot s \cdot 2^{-\Delta^{1/2d}/20}$$

Theorem 3. *Any k-wise uniform distribution with $k = (\log s/\varepsilon)^{O(d^2)}$ is ε-pseudorandom for circuit of depth d and size s.*

The first important observation towards proving this result is the following, for any polynomial $P : \{0,1\}^n \mapsto \mathbb{R}$ of degree at most k,

$$\mathbb{E}[P(D)] = \mathbb{E}[P(U)]$$

if D is poly-logarithmic-wise independent.