U.C. Berkeley
CS294: Pseudorandomness and Combinatorial Constructions
Professor Luca Trevisan

Handout N19
November 3, 2005
Scribe: Anand Kulkarni

Notes for Lecture 19

1 Improved extractors

Last time we used the Nisan-Widgerson generator and list-decodable codes to build a (k, ϵ) randomness extractor

$$Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m \tag{1}$$

with entropy equal to $k = O(m^2) + O(log(\frac{1}{\epsilon}))$ and needing $d = O(log(\frac{1}{\epsilon}))$ bits of extra randomness., provided $k = n^{\Omega(1)}$. For instance, we could put $\epsilon = \frac{1}{n}$, $k = \sqrt{n}$, $d = O(\log(n))$, and $m = \Omega(n^{\frac{1}{4}})$.

However, if we had $k = (\log(n))^2$, or $k = 2^{\sqrt{(log(n))}}$, then this construction would require $d = O(\frac{(\log(\frac{n}{\epsilon})^2}{\log(m)})$ bits of extra randomness. Today we will give an improved extractor which does not have a $\frac{n}{\epsilon}$ dependence. This will make use of a new function called a randomness condenser. Before beginning the improved extractor, we will briefly review the generator.

1.1 Review of Nisan-Widgerson generator

We first recall some facts about the Nisan-Widgerson generator. The generator takes a function that is hard on average, with input length l; ie, $f : \{0,1\}^l \to \{0,1\}$. We construct a family of sets $S_1, \ldots S_m$ such that $\forall i$, $\|S_i\| = l$. Let $a \geq |S_i \cap S_j\|$ and pick $S_i \subseteq 1, \ldots, d$ from the universe. We showed that such a construction is possible with $a = \log m, l = c \log m$, and $d = e^2 c^2 \log m$. The generator then uses f as an oracle, outputting

$$NW^f(z) = f(z|_{S_1}) \ldots f(z|_{S_m}) \tag{2}$$

where $f(z|_{S_i})$ denotes indicates the bitwise intersection of z and S_i.

1.2 Analysis of the generator

Suppose $D : \{0,1\}^m \to \{0,1\}$ is a linear distinguisher. Then,

$$\mathbf{Pr}[D(NW^f(U_d)) = 1] - \Pr[D(U_m) = 1] \geq \epsilon. \tag{3}$$

Define an algorithm A which takes as input x, b.

```
A(x)
  pick i ∈ 1, ..., m
  pick r_1 ... r_{i-1} ∈ {0,1}
  pick z ∈ {0,1}^m conditioned on z‖_{S_i} = x
  output D(r_1 ... r_{i-1} b f(z‖_{S_{i+1}} ... f(z‖_{S_m}))
```

The probability that this algorithm will return a true value when run on $f(x)$ compared to a random string r is

$$\mathbf{Pr}[A(x, f(x)) = 1] - \mathbf{Pr}[A(x, r) = 1] = \mathbb{E}_i(\mathbf{Pr}[D(H_{i-1}) = 1] - \mathbf{Pr}[D(H_i) = 1] \geq \frac{\epsilon}{m}, \quad (4)$$

using the fact that

$$\mathbf{Pr}[D(NW^f(U_d)) = 1] - \mathbf{Pr}[D(U_m) = 1] = \sum_{i=1}^{m} \mathbf{Pr}[D(H_{i-1}) = 1] - \mathbf{Pr}[D(H_i) = 1]. \quad (5)$$

One of $A(x,0), A(x,1), \overline{A}(x,0)$, or $\overline{A}(x,1)$ computes x correctly on at least a $\frac{1}{2} + \frac{\epsilon}{m}$ fraction of the inputs. Once we fix z on $z\|_{[d]-S_i}$, the complexity of computing all evaluations on z will be $O((m-i)z^2) = O(m^2)$.

We concluded that if D is a distinguisher with probability $\mathbf{Pr}[D(NW^f(U_d)) = 1] - \mathbf{Pr}[D(U_m) = 1]$ of distinguishing the Nisan-Widgerson generator from a uniform distribution, then A can be constructed as a circuit constructed using at most $\log m + (i-1) + (m-i)z^2 \leq \log m + mz^2 \leq \log m + mz^2$ bits of information, that agrees with f on $\geq \frac{1}{2} + \frac{\epsilon}{m}$.

1.3 Review of the extractor

In the last lecture we saw how to build an extractor by first choosing an error-correcting code with good list-decoding properties, ie, where the number of possible messages that agree with a string was at most L. We know that this is possible with an encoding length of $\overline{n} = poly(n, \frac{2m}{\epsilon})$ and list length $L = poly(\frac{m}{\epsilon})$, using a code $C : \{0,1\}^n \to \{0,1\}^n$ such that $\forall y \in \{0,1\}^{\overline{n}}, \|x_i C(x)$ agrees with on $\geq \frac{1}{2} + \frac{\epsilon}{m}\| \leq L$.

So, if x is the n-bit output of a weak random source, then $NWExt(x,z) = NW^f(z)$, where $f : \{0,1\}^l \to \{0,1\}$ is a function whose truth table is $C(x)$, with $l = \log(\overline{n}) = O(\log n + \log \frac{1}{\epsilon})$.

In the analysis, we wanted to show that, for sufficiently large entropy k, NWExt is a (k, ϵ) extractor. We proceeded in a manner similar to the construction of the Nisan-Widgerson pseudorandom generator.

Claim 1 *The Nisan-Widgerson Extractor is a (k, ϵ)-extractor*

PROOF: *Observe that if NWExt is not (k, ϵ), then there must exist a distribution X of entropy $\geq k$ and a statistical test D such that $\mathbf{Pr}_{x,z}[D(NWE(x,z)) = 1] - \mathbf{Pr}[D(U_m) = 1] \geq \epsilon$. We show for a contradiction that X must have an entropy $\leq k$.*

Call X "bad" if $\mathbf{Pr}_{x\sim X}[D(NWE(x,z)) = 1] - \mathbf{Pr}[D(U_m) = 1] \geq \frac{\epsilon}{2}$. We will show that the number of bad x is small. We have that $\mathbf{Pr}_{x\sim X}[x \text{ is bad}] \leq \frac{\text{number of bad } x}{2^k}$. Thus, for a contradiction, we need to show that the number of bad x is $\leq 2^{O(m^2)}$, contradicting $k \geq O(m^2) + \log \frac{1}{\epsilon}$.

Fix a bad x. $\overline{x} = C(x)$. Let f be a function whose truth table is $\overline{x}$. Then, $NWE(x,z) = NW^f(z)$, and $\mathbf{Pr}_{x,z}[D(NWE(x,z)) = 1] - \mathbf{Pr}[D(U_m) = 1] \geq \frac{\epsilon}{2}$. So, there is a circuit constructable using $\leq \log m + m2^a + 2$ bits of information, which describes a string y with agreement $\geq \frac{1}{2} + \frac{\epsilon}{2m}$ with $\overline{x}$.

How many "bad" x does this imply? If someone knows the error correcting code and has this circuit C, then he can get all possible x simply by using $\log L$ additional bits, since there are L possible values in his list. Thus, x can be specified entirely using $\log m + m2^a + 2 + \log L$ bits. Hence, the number of bad $x \leq 2^{m2^a} + \log m + 2 + \log \frac{m}{\epsilon} = 2^{m^2 + O(\log m) + O(\log \frac{1}{\epsilon})}$, giving the desired contradiction.
□

1.4 Small-k distributions

The next result handles the case where the minimum entropy is very small.

Suppose that X is a source of minimum entropy k that is uniform over some subset $S \subseteq \{0,1\}^n$ of size 2^k. Let's imagine what would happen if we ran $NWE_{x\sim X, z\sim Z}(x,z) = NW^f(z)$, where f is the function whose truth-table is $C(x)$, but put $m = n^{\frac{1}{4}}$ and $k << n$. We know that the output of the extractor will not be close to the uniform distribution, so this means that there exists a linear distinguisher D such that

$$D(y) = 1 \text{ iff } \exists x \in S \text{ and } \exists z \in \{0,1\}^d : NWE(x,z) = y \tag{6}$$

ie, that $\mathbf{Pr}[D(U_m = 1] \leq \frac{2^{k+d}}{2^m}$ and that $\mathbf{Pr}[D(NWE(X,U_d)) = 1] = 1$. Further, let $B \subset S$ be the set of bad $x \in B$. $x \in B$ iff $\mathbf{Pr}[D(NWE(X,U_d)) = 1] - \mathbf{Pr}[D(U_m) = 1] \geq \frac{1}{4}$. Since $\mathbf{Pr}_{x\in S}[x \in B] \geq \frac{1}{2}$, we have that $\|B\| \geq 2^{k-1}$.

What does it mean if x is bad? Each $x \in B$ can be computed with a circuit that has a small description. Let f be a function whose truth table is $C(x)$. The algorithm we gave before will distinguish $x, f(x)$ from x, r. Suppose instead we define the following condenser algorithm:

```
Condenser (x, z, i)
  for j = 1, ..., m  j ≠ i
    for y ∈ {0,1}^l
        set z‖_{S_i} = y
        output f(z‖s_j)
```

The condenser program takes input from the weak random source and a short truly random string. Its output will be shorter than the input but will have about the same entropy. In particular, the condenser maps n mostly-random bits with entropy k to $\sqrt{n}$ bits with entropy approximately k. We saw that using the Nisan-Widgerson extractor on small strings leads to 'bad' x. Using the condenser, we will have an output string length of $(m-1)z^2 \leq m^2 = \sqrt{n}$, with an entropy $k' \geq (k-1) - \log L - 2$.

There are some more complications to the condenser; for example, we have given a slightly inaccurate representation of entropy here. We will continue with the condenser during the next lecture. For the moment, simply observe that the extra random bits needed here are bounded, since the input is $c \log n$ bits long and the output is $c \log \sqrt{n}$ bits.