CS 830: Randomness and Complexity	04/26/05

Lecture 27: Trevisan's Extractor

instructor: Jin-Yi Cai — Scribe: Scott Diehl, Jeff Kinne

Today we present and extractor due to Trevisan that was later improved by Raz, Reingold, and Vadhan. We will first give recap the error correcting codes and combinatorial designs we will need for the extractor. We then define the extractor and begin proving that it is in fact an extractor. The proof is finished in the next lecture.

1 Recap of Error Correcting Codes

Recall the following result about Hamming balls in sets with large pairwise distance:

Lemma 1.1 *If $S \subseteq \{0,1\}^n$ is such that the pairwise Hamming distance between all points in S is at least d, then for any Hamming ball has a small number of points from S. In other words, for any point $p \in \{0,\}^n$*

$$|B_p(w) \cap S| \leq \frac{1-2\beta}{4\alpha^2 - 2\beta},$$

where $B_p(w)$ is the Hamming ball of radius w around p, $\alpha = 1/2 - w/n$ and $\beta = 1/2 - d/n$.

This was shown by considering the sum $\sum \langle v_i, v_j \rangle$ in two different ways. We can restate Lemma 1.1 in a more usable manner as follows:

Corollary 1.2 *If S has relative pairwise distance at least $1/2 - \gamma/2$, then for all balls B of relative radius $1/2 - \sqrt{\gamma}$, B has at most $\frac{1-\gamma}{3\gamma} < \frac{1}{3\gamma} < \frac{1}{\gamma}$ many points from the "code" S. In other words,*

$$|B \cap S| < \frac{1}{\gamma}.$$

To be able to make use of these results, we need an error correcting code that has large pairwise distance. We will give such a code. Specifically, we consider the concatenation of the Reed-Solomon code with the Hadamard code.

Given $\alpha \in \{0,1\}^n$, the Reed-Solomon code thinks of α as specifying the coefficients of a degree $d \doteq \lfloor n/m \rfloor - 1$ polynomial over a field $\mathbb{F}$ where $|\mathbb{F} = 2^m|$. The output of the Reed-Solomon code is then this polynomial $p_\alpha(x) \in \mathbb{F}[x]$ evaluated at every point in $\mathbb{F}$:

$$\alpha \mapsto (p_\alpha(a_1), p_\alpha(a_2), \ldots, p_\alpha(a_{|\mathbb{F}|})).$$

Since α and α' define different degree at most d polynomials for $\alpha \neq \alpha'$, their encodings under the RS code can agree in at most $|\mathbb{F} - d|$ points. However, this does not mean that the bit-wise distance must be high, since the points on which their encodings differ could only be different by one bit. Therefore, we further encode each point of the output of the RS code with the Hadamard code, resulting our error correcting code $EC_{n,\delta}$:

$$EC_{n,\delta}(\alpha) = (H(p_\alpha(a_1)), H(p_\alpha(a_2)), \ldots, H(p_\alpha(a_{|\mathbb{F}|}))),$$

where H is the Hadamard encoding function and δ is the distance parameter defining m. The Hadamard code maps a bit string of length m to its inner product with all 2^m bit strings of length m, and hence has length 2^m. The important property of H is that the relative distance is $1/2$ for different inputs (even if they differ by only one bit). Therefore, the output of EC is 2^m strings, each of length 2^m, so the total length is 2^{2m}. Two different strings are encoded so that at least $2^m - d$ points differ after applying the Reed-Solomon code, which in turn will differ in at least half of bit positions after applying the Hadamard code to each, so the relative distance of EC is

$$\mathrm{rel}(EC_{n,\delta}) = \frac{(2^m - d)2^m/2}{2^{2m}} \geq \frac{1}{2} - \frac{n}{2^{m+1}}.$$

Letting $m = \lceil \log(n/\delta^2) \rceil$ and $\sigma = \delta^2$, we can see that

$$\overline{n} \doteq |EC_{n,\delta}(\alpha)| = 2^{2m} = O(n^2/\delta^4),$$

and that the relative distance $\mathrm{rel}(EC_{n,\delta}) \geq \frac{1}{2} - \frac{\sigma}{2}$. Therefore, by Corollary 1.2, any ball of relative radius $\frac{1}{2} - \delta$ contains at most $\frac{1}{\delta^2}$ points from the code.

2 Weak Designs

The optimized construction of Trevisan's extractor which we will shortly present makes use of the Nisan-Widgerson generator, but with combinatorial designs that are more general than those used in the derandomiation setting. Intuitively, the requirement on these designs are such that the "average" pairwise intersection is small, rather than requiring all of the pairwise intersections to be small.

More formally, recall the traditional definition of designs:

Definition 2.1 *For $\ell \in \mathbb{N}, \rho \geq 1$, an (ℓ, ρ)-design is a collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ with $S_i \subseteq [d]$ such that:*

- *$\forall i, |S_i| = \ell$, and*
- *$\forall i \neq j, |S_i \cap S_j| \leq \log \rho$.*

We similarly define weak designs as follows:

Definition 2.2 *For $\ell \in \mathbb{N}, \rho \geq 1$, a weak (ℓ, ρ)-design is a collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ with $S_i \subseteq [d]$ such that:*

- *$\forall i, |S_i| = \ell$, and*
- *$\forall i, \sum_{j<i} 2^{|S_i \cap S_j|} \leq \rho(m-1)$.*

Note that if $\mathcal{S}$ is a (ℓ, ρ)-design, then it is also a weak (ℓ, ρ)-design, so that the second definition is indeed a generalization of the first.

3 The Extractor

We now set out to design an extractor that takes an imperfect random source of n bits with minentropy k as well as d uniform bits, and outputs m bits that are ϵ-close to uniform. The behavior of the extractor will be dictated by a weak (ℓ, ρ)-design $\mathcal{S}$ and $EC_{n,\delta}$ for δ depending on ϵ and m. We will first apply EC to the imperfect random bits to generate $\overline{n}$ bits. We will then view the output of EC as a truth table for a function $\overline{u}$. We will then apply the function $\overline{u}$ to different subsets of the d random bits, choosing which subsets using the design $\mathcal{S}$, and this will be the output of the generator. (Note that this last part of the extractor - choosing different subsets using a design, and then evaluating a function at those points - is just the Nisan-Wigderson generator using $\overline{u}$ as the hard function with seed $y \in_U \{0,1\}^d$).

$$\begin{array}{rcl} Ext_{\mathcal{S}} : \{0,1\}^n \times \{0,1\}^d & \rightarrow & \{0,1\}^m \\ Ext_{\mathcal{S}}(u, y) = NW_{S,\overline{u}}(y) & = & \overline{u}(y|_{S_1}) \cdots \overline{u}(y|_{S_m}) \\ \text{where} & & \overline{u} = EC_{n,\delta}(u). \end{array}$$

We now give a brief outline of the proof that $Ext_{\mathcal{S}}$ is a strong (k, ϵ)-extractor, with a more formal treatment to be given next time. Suppose that $Ext_{\mathcal{S}}$ is not a strong (k, ϵ)-extractor, i.e. that the distribution $(U_d, Ext_{\mathcal{S}(X,U_d)})$ for X having minentropy k has statistical distance at least ϵ from (U_{m+d}). Then by Yao's argument, there exists a next-bit predictor $A : \{0,1\}^d \times \{0,1\}^{i-1} \rightarrow \{0,1\}$ for Ext with error less than $\frac{1}{2} + \frac{\epsilon}{m}$. In other words, when given $y \in_U \{0,1\}^d$ and $\overline{u}(y|_{S_1}), \ldots, \overline{u}|_{S_{i-1}}$, A will provide the correct value of $\overline{u}(y|_{S_i})$ on more than a $\frac{1}{2} + \epsilon/m$ fraction of the points (u, y).

Consider a fixed "bad" u, for which A provies a good approximation to $\overline{u}$,

$$\Pr_{y \in_U \{0,1\}^d}[A(y, \overline{u}(y|_{S_1}), \ldots, \overline{u}(y|_{S_{i-1}})) = \overline{u}(y|_{S_i})] > \frac{1}{2} + \frac{\epsilon}{2m}.$$

Our goal will be to show that there cannot be many such u, and that moreover, since u is selected from a distribution with high minentropy, each such u is not very likely to be picked. Therefore, the joint probability of A's success over u, y will be small.

Since $\overline{u}(y|_{S_i})$ is a function depending only on the bits of y indexed by S_i, we can apply an averaging argument to guarantee the existence of a 0-1 assignment to the bits of y not indexed by S_i such that A's success probability over the remaining bits is preserved. Letting $y(x)$ denote such an assignment with $x \in \{0,1\}^\ell$ be the values given to the bits indexed by S_i, we can write

$$\Pr_{x \in_U \{0,1\}^\ell}[A(y(x), \overline{u}(y(x)|_{S_1}), \ldots, \overline{u}(y(x)|_{S_{i-1}})) = \overline{u}(x)] > \frac{1}{2} + \frac{\epsilon}{2m}.$$

Since the S_j's are from a weak design, the evaluation of $\overline{u}$ at $y(x)|_{S_j}$ should have very little to say about $\overline{u}(y(x)|_{S_i} = x)$ for $j < i$ because the average intersection between these sets is small. Thus the dependence of A on x should be "easy" to describe. In fact, by the observation that $\overline{u}(y(x)|_{S_j})$ is a function on $|S_j \cap S_i|$ bits (and thus can be described by $2^{|S_j \cap S_i|}$ bits), we can describe a circuit $f(x)$ mapping $x \mapsto (y(x), \overline{u}(y(x)|_{S_1}), \ldots, \overline{u}(y(x)|_{S_{i-1}}))$ by $d + \sum_{j<i} 2^{|S_j \cap S_i|} \leq d + \rho(m-1)$ bits (where the extra d bits indicate a bitmask to apply to y to select the proper bits of x).

Suppose have have such an $A(f(x))$ that gives a good approximation for the function $\overline{u}(x)$ that is the output of $EC_{n,\delta}(u, y)$. This means that the characteristic string of $A(f(x))$, of length $2^\ell = \overline{n}$ is close to that of $\overline{u}$. It is also possible that $A(y(x), f(x))$ approximates other $\overline{u}'$, but by the properties of $EC_{n,\delta}$, this number is very few, say, $(2m/\epsilon)^2$. Since there are at most $2^{d+\rho(m-1)}$ possible functions f, A can approximate at most $(2m/\epsilon)^2 \cdot 2^{d+\rho(m-1)}$ such u, giving us the desired upper bound on the number of "bad" u.

In the next lecture, we will complete the last few steps of the proof, and conclude

THEOREM 3.1 *If $\mathcal{S}$ is a weak (ℓ, ρ)-design for $\rho = (k - 3\log(m/\epsilon - 3)/m$, then $Ext_{\mathcal{S}}$ is a strong (k, ϵ) extractor.*