Lecture 29

In this lecture, we shall look at *randomness extractors* and see a beautiful connection between them and pseudo-random generators, due to which a PRG such as the Nisan-Wigderson generator may be used to construct extractors. Then we shall see how those ideas may be used to get worst-case to average case reductions.

1 Extractors and Error-Correcting Codes

We begin be recalling the definition of extractors. To define them we need a notion called the minimum entropy of a distribution, which is a lower bound on the bits of information contained in a distribution, it is the fewest bits required to specify the probability of a particular element in the distribution.

DEFINITION 1 (**Min Entropy**) *The* Min Entropy *of a distribution* $X = \min_a \log\left(\frac{1}{\mathbf{Pr}[X=a]}\right)$.

The min entropy is a measure of the amount of randomness of a distribution, a distribution with low min entropy is one where the weight is heavily concentrated at a particular element and hence one which contains little randomness.

An extractor is a procedure that when given an arbitrary distribution with high enough min-entropy, and a few completely random bits, will return a distribution which is as close to the uniform as is desired. The notion of closeness of distribution is the well known variation distance, we write it here as:

DEFINITION 2 (**Closeness of Distributions**) *Distributions* X, Y *over some set* S *are* $\epsilon-$*close if for every function* $T : S \to B$*, we have:* $|\mathbf{Pr}_{x \in X}[T(x) = 1] - \mathbf{Pr}_{y \in Y}[T(y) = 1]| \leq \epsilon$.

Finally, we are ready to state the definition of Extractors:

DEFINITION 3 (**Extractor**) *A function* $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ *is a* $(k, \epsilon)-$*extractor if for every distribution* X *(over* $\{0,1\}^n$*) of min-entropy* $\geq k$, $Ext(X, S)$ *is* $\epsilon-$*close to being uniform (over* $\{0,1\}^m$*), where* S *is uniform in* $\{0,1\}^t$.

Basic Idea

To connect extractors with PRGs we will use *Error-Correcting Codes*. The basic idea of the connection which we shall see in detail soon is that given a particular element from the distribution, we shall encode that element and then think of the encoded string as a function to be supplied as a seed to the NW generator.

To show that the resulting distribution is close to uniform, we shall fix a(n arbitrary) distinguishing circuit T and argue that the probability of *bad* seeds, which are seeds which cause the generator to give outcomes that are predicted by T is small.

This happens as from the analysis of the NW generator and the property of the ECCs, which ensure that the functions are really very sparsely encoded, it follows that if a seed is bad, then it may be represented by a few number of bits and thus there are few of them. The min-entropy is then used to bound the weight in the distribution of such seeds.

Thus, every distinguishing circuit does well only for a negligibly small probability, and so no circuit can distinguish the output of the extractor from uniform.

We now proceed to the details of Error Correcting codes and see what "sparseness" means, and then to the construction of the extractors and the analysis.

DEFINITION 4 (**Error-Correcting Code**) *A function* $C : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ *is an* Error-Correcting Code *with minimum distance* α *if for every* $x_1,\ x_2 \in \{0,1\}^n$*, the hamming distance between* $C(x_1),\ C(x_2)$ *is at least* $\alpha\bar{n}$*.*

The range points of the function C are called the codewords. If we draw balls of radius $\alpha/2$ around the codewords, we get disjoint sets.

There is a well known result in coding theory, that there is no code with $\alpha \geq \frac{1}{2}$, so if more than one fourth of the message bits are corrupted then there is no way that the original word can be uniquely (correctly) reconstructed. However, there is a bound on the maximum number of codewords that are within a certain distance of a particular codeword:

THEOREM 1
Suppose $C : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ *has minimum distance* $\frac{1}{2} - \delta^2$*. Then for every* $a \in \{0,1\}^{\bar{n}}$ *there are at most* $1/(3\delta^2)$ *strings* $x \in \{0,1\}^n$ *such that* a *and* $C(x)$ *differ in* $\leq (\frac{1}{2} - \delta)\bar{n}$ *co-ordinates.*

We will also use the result that for every n, δ there is a $C : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ computable in poly$(n, \frac{1}{\delta})$ time with minimum distance $\geq \frac{1}{2} - \delta^2$.

Nisan-Wigderson Revisited

Let us quickly review the important points from the analysis of the NW generator :

1. We assumed there was a function $f : \{0,1\}^l \to \{0,1\}$ satisfying certain complexity requirements.

2. We (randomly) selected $S_1, \ldots, S_m \subseteq \{1, \ldots, t\}$ where $t = O(\frac{l}{\gamma})$ where $m = 2^{\gamma l}$. Also, for each $i \in \{1,...,m\}$ we had $|S_i| = l$ and for each pair $i, j \in \{1,...,m\}$ we had $|S_i \cap S_j| \leq \log m$.

3. Finally the generator was defined as:
$$\mathcal{NW}_{f,S_1,\ldots,S_m}(z) = f(z|_{S_1}) \cdots f(z|_{S_m})$$

4. In order to study the security of this generator, we saw that if $T : \{0,1\}^m \to \{0,1\}$ is such that $|\mathbf{Pr}_z[T(\mathcal{NW}(z)) = 1] - \mathbf{Pr}_r[T(r) = 1]| \geq \epsilon$, then there exists a circuit $\widehat{C}$ of size $O(m^2) + \mathbf{SIZE}(T)$, such that $\mathbf{Pr}_x[\widehat{C}(x) = f(x)] \geq \frac{1}{2} + \frac{\epsilon}{m}$.

We shall now see how this generator may be used to construct good extractors.

2 Construction of Extractors

Suppose we are given a distribution X over $\{0,1\}^n$ of min-entropy $k = n^\alpha$. Also assume we have a code $C : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ of minimum distance $\frac{1}{2} - \frac{\epsilon^2}{m^2}$, where $m = k^{1/3}$. The extractor is as follows:

1. Sample a from X, and encode a with C to get $C(a) \in \{0,1\}^{\bar{n}}$, where $\bar{n} = \text{poly}(n)$.
2. View $C(a)$ as a function $f : \{0,1\}^l \to \{0,1\}$, where $l = \log \bar{n}$.
3. Pick a random seed $z \in \{0,1\}^t$.
4. Output $\mathcal{NW}_{f,S_1,\ldots,S_m}(z)$.

We will now see that the output of the above procedure is $2\epsilon-$close to uniform. Note firstly that $l = \log \bar{n} = \Theta(\log n) = \Theta(\log m)$ where $m = k^{1/3}$, and k was the min-entropy and equal to n^α.

Recall now that the length of the random seed that $\mathcal{NW}$ takes is $t = O(\frac{l}{\gamma})$, where $\gamma = \log m/l$. As $l = \Theta(\log m) = c \log m$, it follows that $t = O(l/c) = O(\log n)$. Thus, as required for the original purpose of derandomizing, the length of the purely random seed that the extractor requires is indeed logarithmic.

Let us now look at the proof of the claim. We shall see that for any distribution X having the required min-entropy, the output of the extractor is close to uniform.

Lemma 2

For any distribution X of min-entropy $\geq k$, and any $T : \{0,1\}^m \to \{0,1\}$ we have $|\mathbf{Pr}[T(\mathcal{NW}_{C(a),S_1,\ldots,S_m}(z)) = 1] - \mathbf{Pr}[T(r) = 1]| \leq 2\epsilon$, where a is sampled from X, and z and r are uniformly from $\{0,1\}^t$ and $\{0,1\}^m$ respectively.

Proof: We call $a \in \{0,1\}^n$ in the support of X *bad* for T if

$$|\mathbf{Pr}[T(\mathcal{NW}_{C(a),S_1,\ldots,S_m}(z)) = 1] - \mathbf{Pr}[T(r) = 1]| > \epsilon$$

where now we only sample z, r.

We know from the entropy assumption on X that all $a \in \{0,1\}^n$ that are in the support of X have probability at most $1/2^k$. We wish to bound the number of *bad* a's, and putting the two together we shall see that the total weight of bad a's in the distribution X is small.

Suppose that a is bad for T. Let $f : \{0,1\}^l \to \{0,1\}$ be the function corresponding to $C(a)$, and we know that T "breaks" $\mathcal{NW}_f$. Thus, from the (point 4) in the analysis of the NW generator, we know that there exists a function $g : \{0,1\}^l \to \{0,1\}$, such that g may be described by T and a circuit of size $O(m^2)$ and $\mathbf{Pr}[f(x) = g(x)] \geq \frac{1}{2} + \frac{\epsilon}{m}$.

Note now that the function f is "close" to the function g, and moreover these functions are both in the coded space, where the data points are very sparse. Thus, given g, there are in fact very few $b \in \{0,1\}^n$ such that $C(b)$ is close to g, and from the small set of b's that are close (which includes a), we need only to index into the correct one to completely specify a.

Formally, we have that in a code of minimum distance $1/2 - \delta^2$, in a ball of radius $\frac{1}{2} - \delta$ there are at most $1/(3\delta^2)$ other words, where here we have $\delta = \epsilon/m$. Thus, to index into the element of this set of words that corresponds to a we need $\log(\frac{m^2}{3\delta^2})$ bits.

Moreover, any circuit of size $O(m^2)$ can be represented using $O(m^2 \log m)$ bits. Thus, given T, we can completely specify a using at most $O(m^2 \log m + \frac{m^2}{3\delta^2})$ bits of information. Thus, we have that $\sharp$ *bad* $a \le 2^{O(m^2 \log m)}$. Combining this with the observation about the max probability of elements in the support of X we have $\mathbf{Pr}_a[a \text{ is bad}] \le 2^{O(m^2 \log m)} . 2^{-k} \le \epsilon$ as $m = k^{1/3}$. It follows now that:

$$
\begin{aligned}
&|\mathbf{Pr}[T(\mathcal{NW}_{C(a),S_1,\ldots,S_m}(z)) = 1] - \mathbf{Pr}[T(r) = 1]| \\
&\quad \le |\mathbf{Pr}_{\text{bad } a}[T(\mathcal{NW}_{C(a),S_1,\ldots,S_m}(z)) = 1] - \mathbf{Pr}[T(r) = 1]| \\
&\quad\quad + |\mathbf{Pr}_{\text{good } a}[T(\mathcal{NW}_{C(a),S_1,\ldots,S_m}(z)) = 1] - \mathbf{Pr}[T(r) = 1]|
\end{aligned}
$$

As the max contributions of the bad, good a's is at most 1, ϵ respectively

$$
\begin{aligned}
&\le \epsilon \cdot 1 + (1 - \epsilon) \cdot \epsilon \\
&\le 2\epsilon - \epsilon^2 \\
&< 2\epsilon
\end{aligned}
$$

□

Putting the procedure and the claim together we get the following theorem:

THEOREM 3
For every $k = n^{\Omega(1)}$ there is a poly(n) time computable extractor $Ext : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$, where $t = O(\log n)$, and $m = k^{1/3}$.

Indeed, the above theorem may be generalised to the following existential result:

THEOREM 4
For every n, k, ϵ there is a function : $\{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ that is a $(k, \epsilon)-$extractor with $t = \log(n-k) + 2\log(1/\epsilon) + O(1)$, $m = k + t - 2\log(1/\epsilon) + O(1) = k + \log(n-k)$.

Open Problem: Is it possible to construct extractors where $t = O(\log n)$ and $m = k$.

3 Worst-Case to Average Case Reductions

We will now see how a function with high worst case complexity, may be used to construct functions that have high worst case complexity.

Given $f : \{0,1\}^l \to \{0,1\}$, where the *Circuit Complexity* of f is large say $CC(f) \ge 2^{\epsilon l}$. Let us view f as a string $\langle f \rangle \in \{0,1\}^{2^l}$. Compute $C(\langle f \rangle)$ where C is as before, an Error Correcting Code of minimum distance $1/2 - \delta^2$. View $C(\langle f \rangle)$ as a function $\overline{f} : \{0,1\}^{O(l)} \to \{0,1\}$.

Suppose now there is an algorithm A that computes $\overline{f}$ on a fraction $1/2 + \delta$ of the inputs. Such an algorithm "describes" f upto just $1/(3\delta^2)$ possibilities, and so using it, we can reconstruct f. In particular, given the algorithm A we can precisely define f with little extra overhead. Reasoning again as before, that as the circuit complexity of f was high, it must be that there can be no such succint algorithm A (with a small circuits) as otherwise we would have a small circuit for f. Thus, there is no small circuit that produces the correct result for $\overline{f}$ for even a fraction $1/2 + \delta$ of the inputs and hence $\overline{f}$ has high average case complexity.

Polynomials and Schwartz-Zippel

Indeed we can see now how the properties of polynomials used by the Schwartz-Zippel test for equivalence, can be viewed as a worst case to average case reduction in a manner similar to above, however the reduction is weaker than the one obtained using ECCs.

Say we are given some field $\mathbf{F}$ and polynomials $p : \mathbf{F}^k \to \mathbf{F}$ of degree d and $f : \mathbf{F}^k \to \mathbf{F}$, which has agreement on a fraction $1 - \frac{1}{4d}$ of the points with p. Now, given f we can reconstruct p at *every* point in time $\text{poly}(d, \log \mathbf{F})$.

To see the implications on complexity, say now we have some function $f : \{0,1\}^l \to \{0,1\}$. Pick a field $|\mathbf{F}| = l^3$, pick a subset $H \subseteq F$ such that $|H| = l$. Thus, now $f : H^{\frac{l}{\log l}} \to \{0,1\}$. Define the polynomial $p : \mathbf{F}^{\frac{l}{\log l}} \to \mathbf{F}$ that agrees with f at the points in $H^{\frac{l}{\log l}}$, and of degree $l^2/\log l$.

Note, by doing this, we have effectively encoded f by a bitvector of length 2^l and p a vector of length $2^{O(l)}$.

Suppose there is an algorithm A that computes p in a fraction $1 - 1/l^2$ of the points. Then we know that we can perfectly reconstruct p, in which case, as f has the same values as p on the points in its domain, we have also reconstructed f !

Say f has worst case complexity $= 2^{\epsilon l}$. Then, if S is the size of the circuit for A, then we must have $2^{\epsilon l} < S + \text{poly}(l)$, where the $\text{poly}(l)$ term is the complexity of the reduction. Thus, we get the lower bound $S > 2^{\epsilon l} - \text{poly}(l)$ on the complexity of A.

Thus it is difficult to compute correctly even "most" of the points of p (as opposed to "more than half", the stronger result achieved above) via a reduction from a function which has high worst case complexity.