CSE522/Math581EB Coding Theory **Fall 2001**

Lecture 12: November 20

Lecturer: Chris Umans *Scribe: Özgür Çetin*

12.1 Introduction

How do computers generate samples from a uniform distribution? They probably check the clock or use a function evaluation of the form $x_{n+1} = x_n + b \pmod a$. A more principled approach is to take samples from a physical process e.g. background radiation, and try to purify them since the randomness in nature is correlated, not independent. Extractors are such algorithms that convert a "weakly random" source into an almost uniform distribution. First, let us clarify what we mean to be weakly random and to be almost uniform

The measure that will used to measure randomness is not Shannon's entropy which is rather insensitive to a few large values in the distribution. The amount of randomness will be measured by min-entropy which is defined as follows.

Definition 12.1 *The* **min-entropy** *of distribution X over an alphabet $\mathcal{A}$ is defined to be*

$$H_\infty(X) = \min_{a \in \mathcal{A}} \left(-\log_2 \left(X(a)\right)\right).$$

Note that $H_\infty(X) \geq k$ implies that $X(a) \leq 2^{-k}$. A distribution with min-entropy k can be seen as a generalization of the uniform distribution on some subset of 2^k strings. The min-entropy of a distribution can be interpreted as the number of truly random bits or "hidden randomness" in a weakly random source. The min- and Shannon entropies of the uniform distribution are the same.

A distibution is said to be ϵ-close to another distribution if the statistical distance between the two distributions is less than ϵ.

Definition 12.2 *The* **statatistical distance** *between the distributions X and Y over a common alphabet $\mathcal{A}$ is defined to be*

$$\begin{aligned} SD(X,Y) &\stackrel{\text{def}}{=} \frac{1}{2}\sum_{a \in \mathcal{A}} |X(a) - Y(a)| \\ &= \max_{T:\mathcal{A}\to\{0,1\}} \left|\Pr[T(X)=1] - \Pr[T(Y)=1]\right|. \end{aligned}$$

12.2 Extractors

We have an n-bit source X with at least k-bits of min-entropy and we seek a function EXT producing m-bit strings which are ϵ-close to U_m, the uniform distribution on $\{0,1\}^m$.

If EXT is deterministic, we claim that it is impossible to extract even *one* bit of ramdomness from an n-bit source X with $H_\infty(X) \leq n-1$. We prove this by contradiction. Suppose there exists a deterministic

function $EXT : \{0,1\}^n \to \{0,1\}$ extracting one bit of randomness from every source X with $H_\infty(X) \leq n-1$. Consider the sets $EXT(0)^{-1}$ and $EXT(1)^{-1}$, and take X to be uniform in the larger of the two. Then we obtain a distribution X with $H_\infty(X) \geq n-1$ for which $EXT(X)$ is constant (and therefore far from uniform). So such an extractor can not exist and we can not extract even one bit of randomness with a deterministic extractor.

However, we can use a few "truly" random seed bits independent of the source as the catalyst to generate more random bits. As depicted in figure 1 the general extractor problem is defined as follows.

Definition 12.3 *Function $EXT : \{0,1\}^n \times \{0,1\}^m$ is a (k,ϵ)-**extractor** if for all distributions X with $H_\infty(X) \geq k$, the distribution of $EXT(X, U_t)$ obtained by sampling x from X and u from U_t, the uniform distribution over $\{0,1\}^t$, and evaluating $EXT(x,u)$, is ϵ-close to the uniform distribution over $\{0,1\}^m$.*

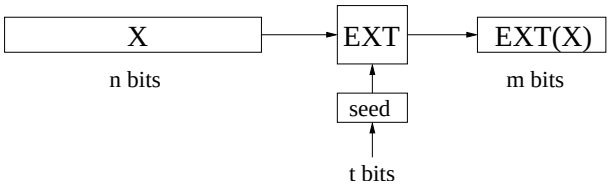

Figure 12.1: An extractor which converts an n-bit weakly random source with k-bits of min-entropy to an almost uniform distribution on $\{0,1\}^m$ using t truly random bits.

We are aiming for $t = O(\log n)$ because in many applications of extractors, we enumerate all seeds and we want this to be possible in polynomial time.

Here is a lemma that suggests how to prove that a function $EXT : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is an extractor. We will see two extractors ([T99] and [SU01]) whose proofs fit into this general framework.

Lemma 12.4 *Suppose there exists a randomized oracle function $R : \{0,1\}^A \to \{0,1\}^n$ such that $\forall x \in \{0,1\}^n$ for which $\exists i \in [m]$ and $\exists f : \{0,1\}^{i-1} \to \{0,1\}$ such that*

$$\Pr_{y \leftarrow Y}\left[f(EXT(x,y)_{1,\ldots,i-1}) = EXT(x,y)_i\right] \geq \frac{1}{2} + \frac{\epsilon}{2m}, \tag{12.1}$$

we have

$$\Pr[\exists z \quad R^f(z) = X] \geq \rho. \tag{12.2}$$

Then EXT is a (k,ϵ) extractor provided that $k > A + \log\left(\frac{1}{\rho}\right) + \log\left(\frac{2m}{\epsilon}\right)$.

We can call those $x \in \{0,1\}^n$ that satisfy equation 12.1 "predictable", as f can predict the i-th bit of $EXT(x,y)$ from the previous bits. The Lemma says that to prove EXT is an extractor, it is sufficient to supply a "reconstruction" procedure R satisfying equation 12.2 that uses f to "reconstruct" such predictable x using a short advice string z. What does this have to do with codes? We might define $EXT(x,y) = C(x)_y$, where C is a binary code with distance close to 1/2. Then we already know that supplying slightly more than 1/2 of $C(x)$ as advice allows us to reconstruct all of $C(x)$ (for *any* x). However we are aiming much more than that: we'd like to reconstruct x using a much shorter advice string; to help us do this we can take advantage of the predictor f.

Proof: Suppose EXT is not a (k,ϵ)-extractor. Then $\exists X$ such that $H_\infty(X) \geq k$, $\exists T : \{0,1\}^m \to \{0,1\}$

$$\left|\Pr_{x \in X, y \in U_t}\left[T(EXT(x,y)) = 1\right] - \Pr\left[T(U_m) = 1\right]\right| \geq \epsilon$$

which can be interpreted as EXT being distinguishable from the uniform distribution by test T. From Yao's Lemma [Yao82] we know that this implies that $\exists i \in [m]$ and $\exists f : \{0,1\}^{i-1} \to \{0,1\}$ such that

$$\Pr_{x\in X, y\in U_t}\left[f(EXT(x,y)_{1,\ldots,i-1}) = EXT(x,y)_i\right] \geq \frac{1}{2} + \frac{\epsilon}{m}.$$

By an averaging argument we get

$$\Pr_{x\in X}\left[\Pr_{y\in U_t}\left[f(EXT(x,y)_{1,\ldots,i-1}) = EXT(x,y)_i \geq \frac{1}{2} + \frac{\epsilon}{2m}\right]\right] \geq \frac{\epsilon}{2m}.$$

For every "good" x (i.e., x such that $\Pr_{y\in U_t}\left[f(EXT(x,y)_{1,\ldots,i-1}) = EXT(x,y)_i \geq \frac{1}{2} + \frac{\epsilon}{2m}\right]$), we have by assumption

$$\Pr\left[\exists z \quad R^f(z) = x\right] \geq \rho.$$

Picking x according to X we have:

$$\Pr_{x\in X}\left[\exists z \quad R^f(z) = x\right] \geq \rho\left(\frac{\epsilon}{2m}\right).$$

and at this point we can make R deterministic by the fixing the coin tosses of R. There are at most 2^A strings in $\{0,1\}^n$ which can be described using an A-bit advice string. The total weight of these strings sampled according to X is 2^{A-k}. In combination with the above result we get

$$2^{A-k} \geq \rho\left(\frac{\epsilon}{2m}\right)$$

which contradicts with $k > A + \log\left(\frac{1}{\rho}\right) + \log\left(\frac{2m}{\epsilon}\right)$. This completes the proof.

12.3 Trevisan's Extractor

Trevison's construction uses an error correcing code C and a (t, m, l, a)-design to extract m-bit ϵ-uniform strings from an n-bit source using t truly random bits [Tr99].

1. We will use a $\left(\left(\frac{1}{2} - \delta\right)\overline{n}, \frac{1}{\delta^2}\right)$ error correcting code $C : \{0,1\}^n \to \{0,1\}^{\overline{n}}$ with $\overline{n} = \text{poly}(n, \frac{1}{\delta})$. For example we can use Reed-Solomon $\circ$ Hadamard with distance $\left(\frac{1}{2} - \delta^2\right)\overline{n}$. From the Johnson bound we know that the codes with this distance are $\left(\left(\frac{1}{2} - \delta\right)\overline{n}, \frac{1}{\delta^2}\right)$ error-correcting. In the construction below we will set $\delta = \frac{\epsilon}{4m}$.
2. Let $\{S_1, \ldots, S_m\}$ be the (t, m, l, a) design required by the construction of NW [NW94], i.e.
   - $S_i \subseteq [t] \quad \forall i$,
   - $|S_i| = l \quad \forall i$,
   - $|S_i \cap S_j| \leq a \quad \forall i \neq j$.

   Informally, S_i's can be thought of as being "nearly disjoint". We will state the following lemma about some set of valid parameters and explicit construction of such designs without proof.

   Lemma 12.5 *For $l = O(\log n)$, $\forall \epsilon > 0$ there exists a (t, m, l, a) design with $a = \epsilon \log n$, $t = O(\log n)$, and $m \leq n$. Moreover, such a design can be built in poly(n) time.*

   In the construction below we will use $l = \log(\overline{n})$.

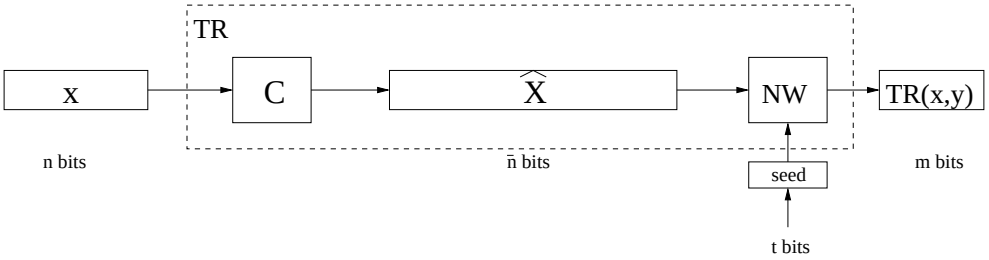

Figure 12.2: Trevisan's Extractor.

Trevisan's extractor is defined as $TR(x,y) = NW_S(\hat{x}, y)$ with $\hat{x} = C(x)$ and $NW : \{0,1\}^{\overline{n}} \times \{0,1\}^t \to \{0,1\}^m$ defined as

$$NW(\hat{x}, y) = \hat{x}(y_{|S_1}) \circ \hat{x}(y_{|S_2}) \circ \cdots \circ \hat{x}(y_{|S_m})$$

in which $\hat{x}(y_{|S})$ is the bit of $\hat{x}$ whose index's binary representation is $y_{|S}$, $y_{|S} \in \{0,1\}^l$ denotes the string $y_{s_1} \circ y_{s_2} \circ y_{s_l}$ and $\cdot \circ \cdot$ denotes the string concatenation.

When we pick a string from a distribution with large min-entropy it is unlikely that it will have a short description. The intuition behind the use of error correcting codes is that when we pick a string from a distribution with large min-entropy and encode it with an error correcting code then with high probability encoding is not even close to a string with short description [Tr99].

Theorem 12.6 *For $a = \Omega(\log n)$, TR is a (k, ϵ) extractor provided $k > m2^a + O(\log(m/\epsilon))$.*

Corollary 12.7 *For all $\gamma > 0$, $\epsilon = 1/m$, $k = n^{\Omega(1)}$, we have a $(k^{1-\gamma}, \epsilon)$ extractor with seed length $t = O(\log n)$.*

Proof: (of Theorem 12.6) Recall that we are given x and $f : \{0,1\}^{i-1} \to \{0,1\}$ such that

$$\Pr_y[f(TR(x,y)_{1\ldots i-1} = TR(x,y)_i)] \geq 1/2 + \epsilon/2m.$$

Taking $\overline{S_i}$ to be $[t] \setminus S_i$, we can write y as $y_{|S_i} \circ y_{|\overline{S_i}}$. We have:

$$\Pr_{y_{|S_i}, y_{|\overline{S_i}}}[f(TR(x,y)_{1\ldots i-1} = TR(x,y)_i)] \geq 1/2 + \epsilon/2m,$$

and an averaging argument shows:

$$\Pr_{y_{|S_i}}[\Pr_{y_{|\overline{S_i}}}[f(TR(x,y)_{1\ldots i-1} = TR(x,y)_i)] \geq 1/2 + \epsilon/4m] \geq \epsilon/4m.$$

For a fixed "good" $y_{|\overline{S_i}}$, notice that $y_{|S_j}$ depends only on $\leq a$ bits that overlap S_i. We have:

$$\Pr_{y_{|S_i}}[f(\hat{x}(y_{|S_1}), \hat{x}(y_{|S_2}), \ldots, \hat{x}(y_{|S_{i-1}})) = \hat{x}(y_{|S_i})] \geq 1/2 + \epsilon/4m. \tag{12.3}$$

We can supply the $i-1$ arguments to f from $i-1$ tables of 2^a values each. If the advice string supplies these tables, then our reconstruction procedure R can use f to compute a "guess" r_j for each $\hat{x}(j)$. Equation 12.3 then implies

$$\Pr_j[r_j = \hat{x}(j)] \geq 1/2 + \epsilon/4m.$$

Thus $r \in \{0,1\}^{\bar{n}}$ agrees with the codeword $\hat{x}$ on a $(1/2 + \epsilon/4m)$ fraction, and the properties of the code imply that there are at most $(4m/\epsilon)^2$ codewords with this agreement with r. We pick a random one from this list and output it. Altogether, we have:

$$\Pr[\exists z \;\; R^f(z) = x] \geq (\epsilon/4m)(\epsilon/4m)^2;$$

that is, our reconstruction procedure R succeeds if we pick a "good" $y_{|\overline{S_i}}$ *and* we output the correct codeword in the list.

Applying the main Lemma above completes the proof. ■

References

[NW94] N. Nisan, A. Wigderson, "Hardness vs. Randomness", *Journal of Computer and System Sciences*, 49:149-167, 1994.

[SU01] R. Shaltiel, C. Umans, "Simple Extractors for All Min-Entropies and a New Pseudo-Random Generator", *Proceedings of FOCS'01.*

[Tr99] L. Trevisan, "Construction of Extractors Using Pseudo-Random Generators", *Proceedings of STOC'99.*