CS 6810: Theory of Computing **Fall 2021**

Lecture 25: Nov 23, 2021

Lecturer: Eshan Chattopadhyay *Scribe: Mark Schachner*

1 Computing with defective randomness

We take a brief look at the theory of extractors and imperfect randomness.

The motivation for this topic comes from the desire to simulate randomness in real-world applications, where "true" randomness is difficult to come by. Many of the complexity classes we have discussed– BPP, RP, interactive and probabilistically checkable proofs, to name a few– have made crucial use of an unlimited stream of random, uniformly distributed bits. So in order to implement some of the models of computation we have considered requires some good approximation of this. Other crucial applications of randomness are in cryptography, distributed computing, etc. This leads to the notion of *extractors*, i.e., functions which take in a sequence of biased bits and return a sequence of bits whose distribution is sufficiently close to uniform.

Example 1.1. [von Neumann's extractor] An important first example of an extractor is due to von Neumann in the '50s: suppose we have a stream of biased bits, i.e, a sequence $x_1, \ldots, x_n$ such that for each i we have $Pr[x_i = 1] = p$ for some $p \in [0,1]$. Crucially, we do not know the value of p, but in order for this procedure to work we will require p be bounded away from 0 and 1. von Neumann observed that, given access to the x_i, we could nonetheless output a single bit which was nearly uniformly distributed. His procedure was as follows:

- Collect the biased bits into pairs $(x_1, x_2), (x_3, x_4), \ldots, (x_{n-1}, x_n)$. (If n is odd we just throw out the last bit.)
- Take the first pair (x_1, x_2). If $x_1 \neq x_2$ then output x_1; otherwise, proceed to the next pair.
- Repeat until either a pair where $x_i \neq x_{i+1}$ has been reached (in which case x_i is returned), or all the bits have been exhausted. In this last case we simply return 0.

We can calculate the distribution of the resulting output as follows: if some pair $x_{2i-1} \neq x_{2i}$ is reached, then $(x_{2i-1}, x_{2i}) = (0,1)$ with probability $(1-p)p$, and $(x_{2i-1}, x_{2i}) = (1,0)$ with probability $p(1-p)$. These are equal, so as long as we find such a pair we are guaranteed to output 0 and 1 with equal probability. This leaves the case where no such pair is found, which occurs with probability

$$(p^2 + (1-p)^2)^{n/2} = (1 - 2p + 2p^2)^{n/2} \leq (1-p)^{n/2} \leq e^{-pn/2}.$$

Thus, the output bit is $2^{-\Omega(n)}$-close to a uniform bit (for constant $p \in (0,1)$).

2 Min-entropy, extractors, and some bad news

In this section, we make many of the notions in the above exposition precise. In particular, we will want to define extractors over certain classes of distributions, so we will need to make a few definitions in order to state such classes.

We begin with a minor notation convention which we will use repeatedly.

Convention 2.1. We denote natural numbers by n, m, k. We denote by the capitalized versions of these letters the exponential of the corresponding lowercase value: $N = 2^n, M = 2^m$, etc.

Definition 2.2. Let $n \in \mathbb{N}$, and let X be a distribution on $\{0,1\}^n$, i.e., a map $\{0,1\}^n \to [0,1]$ such that $\sum_{i \in [N]} X(i) = 1$. The support $\text{supp}(X)$ is then the set of values which are possible to be sampled from X. We define the *min-entropy* of X by

$$H_\infty(X) = \min_{x \in \text{supp}(X)} \log\left(\frac{1}{Pr[X = x]}\right).$$

Intuitively, $H_\infty(X)$ is larger if X is closer to a uniform distribution: for instance, if $H_\infty(X) \geq k$, then for all $x \in \{0,1\}^n$ we have $Pr[X = x] \leq 2^{-k}$. Since the sum of X across all values must be 1, this leads to the observation that $H_\infty(X)$ is always at most n. We enshrine this in the following definition:

Definition 2.3. Fix $n, k \in \mathbb{N}, 0 \leq k \leq n$. An (n,k)-*source* is then a distribution X on $\{0,1\}^n$ with $H_\infty(X) \geq k$.

The last ingredient in the definition of extractors is a formalization of the notion of distance between distributions, hinted at in the previous section.

Definition 2.4. Let X, Y be distributions over a finite set Ω. The *statistical distance* between X and Y is

$$\begin{aligned}\Delta(X,Y) &= \frac{1}{2}|X - Y|_1 \\ &= \frac{1}{2}\sum_{\omega \in \Omega} |Pr[X = \omega] - Pr[Y = \omega]|.\end{aligned}$$

Equivalently,

$$\Delta(X,Y) = \max_{T \subseteq \Omega} |Pr[X \in T] - Pr[Y \in T]|.$$

If $\Delta(X,Y) < \varepsilon$ for some $\varepsilon > 0$ we write $X \approx_\varepsilon Y$.

We can think of the latter definition of Δ above in two ways: the subsets T can be thought of as "tests" on the sample space Ω, and the statistical distance is then the maximum disagreement between X and Y across any test, or X, Y can be thought of as probability measures $2^\Omega \to [0,1]$, in which case Δ is the metric induced by the supremum norm.
Now, we are at last ready to define extractors.

Definition 2.5. Let $\varepsilon > 0, n, m \in \mathbb{N}$, let $\mathcal{U}_m$ denote the uniform distribution on $\{0,1\}^m$, and let $\mathcal{X}$ be a set of distributions. *An ε-extractor for $\mathcal{X}$* is a function $\text{Ext} : \{0,1\}^n \to \{0,1\}^m$ such that for all $X \in \mathcal{X}$ we have $\text{Ext}(X) \approx_\varepsilon \mathcal{U}_m$.

von Neumann's example above had $m = 1$ (for the class of independent biased sources). Some important remarks about this definition:

Remark 2.6.

(i) Like our other models of randomness, the extractor is allowed random access to all of the bits in its input. So the only question of efficiency is the relationship between the input dimension n and the output dimension m.

(ii) *Unlike* our other dealings with randomness, however, we do not allow multiple accesses or repeated trials on the distribution. Rather, the extractor is thought of as a simple function which runs once on the instance it is given.

Given our definitions, a natural question thus arises: for what values of n, k, ε do there exist ε-extractors for the class of (n, k)-sources? We conclude this section with an unfortunate answer to this question.

Proposition 2.7. *For all $\varepsilon < 1/2, n \in \mathbb{N}, k \leq n-1$, no ε-extractor exists for the class of (n, k)-sources.*

Proof. Suppose for some $\varepsilon > 0, m \in \mathbb{N}$ that $f : \{0,1\}^n \to \{0,1\}^m$ is an ε-extractor for (n, k)-sources. By taking only the first of f's outputs, we can suppose WLOG $m = 1$. Then, since $|f^{-1}(0)| + |f^{-1}(1)| = 2^n$, we must have either $|f^{-1}(0)| \geq 2^{n-1}$ or $|f^{-1}(1)| \geq 2^{n-1}$; WLOG suppose the former. Now, let X_0 be the uniform distribution on $f^{-1}(0)$. Then

$$H_\infty(X_0) = \log\left(\frac{1}{1/|f^{-1}(0)|}\right) = \log(|f^{-1}(0)|) \geq \log(2^{n-1}) = n - 1 \geq k,$$

so X_0 is an (n, k)-source. But observe that $f(X_0) = 0$, so that

$$\varepsilon \geq \Delta(f(X_0), \mathcal{U}_1) = 1/2.$$

By contrapositive, if $\varepsilon < 1/2$ then no such f exists. □

Given this result, if we want to find good extractors, we will have to supply them with additional data.

3 A recovery: seeded extractors

Seeded extractors were introduced by Nisan & Zuckerman in 1996 [NZ96]. Essentially, seeded extractors allow us to recover some of the properties of extractors we would like to have, at the expense of allowing for a small random "seed". They are defined as follows:

Definition 3.1. Let $n, d, m \in \mathbb{N}, \varepsilon > 0$. A (k, ε)-*seeded extractor* is a map $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that for all (n, k)-sources X we have $\text{Ext}(X, \mathcal{U}_d) \approx_\varepsilon \mathcal{U}_m$.

One would ideally like the seed length $d << n$.

Unlike the case above, good seeded extractors do exist. The random construction, in which we simply choose $\text{Ext}(x, y)$ uniformly (independently) for each (x, y) shows the existence of seeded extractors with the following parameters:

$$m = k + d - 2\log(1/\varepsilon) - \mathcal{O}(1),$$
$$d = \log n + 2\log(1/\varepsilon) + \mathcal{O}(1).$$

A lot of research ahs focused on explicit constructions in the '90s and '00s, and now we have seeded extractors with $m = \Omega(k), d = \mathcal{O}(\log(n/\varepsilon))$.

An example of an explicit construction is given as follows. Use the seed to pick a hash function h (from a family of 2-universal hash functions), then evaluate h on the sample from the weak source. This gives excellent seeded exrractors in terms of output length, but suffers from the defect that hash functions require lots of randomness to choose, so the seed must be taken very large.

In the following section, we consider ways to use seeded extractors to simulate BPP.

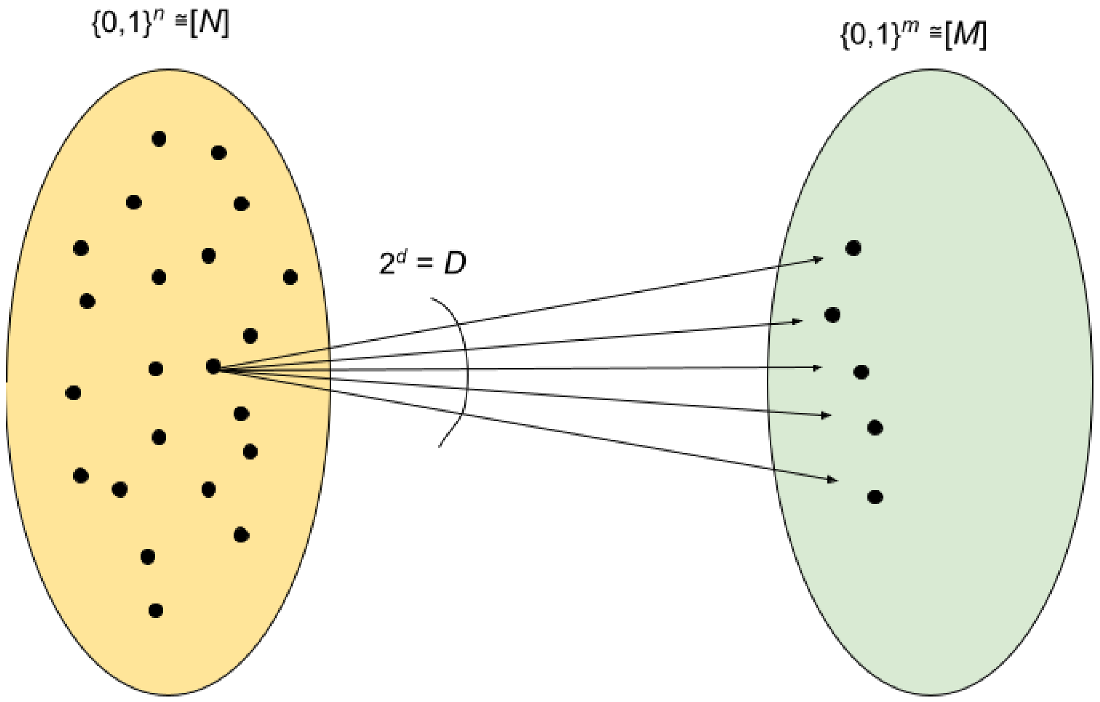
4 Simulating BPP with seeded extractors as samplers.

A useful alternate perspective on seeded extractors is as samplers, defined as follows:

Definition 4.1. Let $n, m, d, k \in \mathbb{N}, \varepsilon, \delta > 0$. A map $\text{Samp} : [N] \to [M]^D$ is a (k, ε, δ)-*sampler* if for all $f : [M] \to \{0,1\}$ and for all (n, k)-sources X,

$$Pr_{(y_1,\ldots,y_D)=\text{Samp}(x), x \leftarrow X}\left[\left|\frac{1}{D}\sum_{i\in[D]} f(y_i) - \mu_f\right| > \varepsilon\right] < \delta.$$

The correspondence between seeded extractors and samplers was discovered by Zuckerman [Zuc97]. To see this correspondence, it is useful to view a seeded extractor as an (N, M)-bipartite graph which is left D-regular, as depicted below. Using this view, output of a sampler on $x \in [N]$ is simply the neighbors of x in the bipartite graph. It was shown by Zuckerman that a (k, ϵ)-extractor is a (k', ϵ, δ)-sampler where $\delta = 2^{k-k'}$.



With this correspondence in hand, we can simulate BPP using weak sources, as follows: Let $L \in \text{BPP}$, so that there exists a polytime PTM $A(x, r)$ satisfying for all $x \in \{0,1\}^*$

$$Pr[A(x, r) = L(x)] \geq 2/3.$$

This is to say that at least $2/3$ of all $r \in \{0,1\}^m$ are "good" for A (i.e., $A(x, r) = L(x)$). Now suppose that we only have access to a weak source (n, k)-source Y. We will construct a new (polytime) randomized algorithm B that solves L using access to Y.

Let $\text{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a $(k/2, \epsilon)$-seeded extractor; we can use it to get an $(k, \epsilon, 2^{-\Omega(k)})$-sampler $S : \{0,1\}^n \to [M]^D$. We denote by S_i the ith (output) sample of S. Now, given an input x and a sample y from an (n,k)-source Y, the algorithm B runs A for D times to compute:

$$\begin{aligned} b_1 &= A(x, \mathrm{Ext}(y, S_1)), \\ b_2 &= A(x, \mathrm{Ext}(y, S_2)), \\ &\vdots \\ b_D &= A(x, \mathrm{Ext}(y, S_D). \end{aligned}$$

Finally, B will simply output the majority b of the b_i's.

It follows by the sampler property of seeded extractors (discussed above) that with probability at least $(1 - 2^{-\Omega(k)})$ over $y \sim Y$, at least $2/3 - \epsilon$ of the b_i will be "good" for A. Thus, taking $\epsilon = 1/10$, it follows that with probability $1 - 2^{-\Omega(k)}$ (over y), we have $B(x, y) = L(x)$. B runs in polynomial since we can use a seeded extractor with seed length $O(\log n)$.

References

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures & Algorithms*, 11(4):345–367, 1997.