CS265/CME309: Randomized Algorithms and Probabilistic Analysis
Lecture #19: A taste of pseudorandomness: extractors and expanders

Mary Wootters

November 15, 2020

1 Pseudorandomness

A *pseudorandom object,* informally, is a deterministic (or not-completely-random) object that behaves like a random one. Pseudorandom objects are useful for, among other things, *derandomization.* For this whole class, we've seen lots of examples of randomized algorithms. But there are many reasons that we might prefer a deterministic algorithm. First, maybe we would rather not leave things to chance. Second, maybe our algorithm has good guarantees when it gets to use uniformly random bits, but where do those bits come from? Unless we've got something quantum going on, we probably don't have access to truly uniformly random bits. Thus, it's often of interest to *derandomize* randomized algorithms. That is, we would like to adapt our randomized algorithms to use less randomness.[1]

In this lecture, we'll see two pseudorandom objects that can be used to "derandomize" randomized algorithms, extractors and expanders. One could teach a whole course about these objects, and we only have one lecture. So we're just going to scratch the surface, to give you a flavor of what this area is like. (Check out CS352 for more!)

2 Extractors

A *randomness extractor*, informally, is an object that takes a "weak" random source and turns it into a "better" random source. In order to define it, we need to quantify what we mean by "weakly" random and what we mean by "better"ly random.

We will measure "weakly random" using min-entropy:

Definition 1. *Given a random variable X with support $\mathcal{X}$, the* min-entropy *of X is given by*

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

[1] Notice that if we can get an algorithm that uses very little randomness—say only b random bits—and succeeds with probability at least $1/2$, then we essentially have gotten a deterministic algorithm. This is because we can iterate over all 2^b choices of the random bits, and then take majority vote.

We say that X is a k-source *if $H_\infty(X) \geq k$, aka if $\Pr[X = x] \leq 2^{-k}$ for all $x \in \mathcal{X}$.*

Notice that if $H_\infty(X) = \log(|\mathcal{X}|)$, then X is the uniform distribution, while if $H_\infty(X)$ is very small, then X is in some sense "far" from uniform, in that there's some x that has way more than $1/|\mathcal{X}|$ probability of occuring. Some examples of k-sources include:

- A random variable $X \in \{0,1\}^n$ that is uniform on k of the n bits, and fixed on the remaining $n - k$.
- A random variable X that is uniform on some set $S \subseteq \{0,1\}^n$ with $|S| = 2^k$.
- Any convex combination of k-sources is a k-source. (Why?)

We will measure "better"-ly random using total variation distance, which we have seen before:

Definition 2. *The* total variation distance *between two random variables X and Y with support $\mathcal{X}$ is*

$$\|X - Y\| = \frac{1}{2} \sum_{x \in \mathcal{X}} |\Pr[X = x] - \Pr[Y = x]|.$$

Now we are ready to define a (seeded) randomness extractor.

Definition 3. *We say that a function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ϵ)-extractor if for all k-sources X on $\{0,1\}^n$,*

$$\|\mathsf{Ext}(X, U_d) - U_m\| \leq \epsilon,$$

where U_d refers to the uniform distribution on $\{0,1\}^d$.

That is, an extractor takes in a *source* $X \in \{0,1\}^n$, which is not uniformly random but rather which is weakly random (it is a k-source); and a *seed* $Y \in \{0,1\}^d$ which consists of d uniformly random bits. It outputs random variable $Z \in \{0,1\}^m$, which we hope is as close to uniform as possible. The goals are:

- Minimize d. This means that we don't have to use too much "pure" randomness as a seed.
- Minimize ϵ. This means that what we get out is as close to uniform as we can get.
- Get m as close to $k + d$ as possible. Intuitively, there are about $k + d$ bit of "randomness" coming into Ext, k from the source and d from the seed, and we'd like to be able to "extract" all of that out.

You might be wondering, why do we allow for a uniformly random seed? Isn't the point that we don't have uniformly random bits and want to get them? In fact, it is *impossible* to have a good extractor (at least for k-sources as defined above) without a seed.

Observation 4. *Suppose that $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$. Then Ext is* not *an (ϵ, k)-extractor (with $d = 0$) for any $k \leq n - 1$ and any $\epsilon < 1/2$.*

Proof. Suppose that $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$. Then consider the set $S_1 \subset \{0,1\}^n$ defined by

$$S_1 = \{x \in \{0,1\}^n : \mathsf{Ext}(x)_0 = 1\}$$

That is, S_1 is the set of x's so that the first bit of $\mathsf{Ext}(x)$ is equal to 1. We can define S_0 similarly (the set of x's so that $\mathsf{Ext}(x)_0 = 0$), and observe that $\{0,1\}^n = S_0 \cup S_1$. Therefore, at least one of S_0 or S_1 has size at least 2^{n-1}. Suppose WLOG that S_1 does. Then define X to be a random variable that is uniform on a set S' of size 2^k with $S' \subseteq S$. (We can do this since $k \leq n-1$). But then $\mathsf{Ext}(X)_0 = 1$ with probability 1. In particular,

$$\|X - U_m\| \geq |1 - 1/2| = 1/2 > \epsilon.$$

□

So, given that we need a seed, our goal is to make d, the length of the seed, as small as possible, while still getting a decent number of output bits. What's the best we can do? The following theorem asserts the *existence* of an extractor with very good (actually, nearly optimal) parameters:

Theorem 1. *For all $n \in \mathbb{N}$, for all $k \in \{0,\ldots,n\}$, for all $\epsilon > 0$, there is a (k,ϵ)-extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ so that:*

$$m = k + d - 2\log(1/\epsilon) - O(1)$$

and

$$d = \log(n-k) + 2\log(1/\epsilon) + O(1).$$

We leave the proof of this theorem as an exercise to the reader. (Hint: try the probabilistic method! A completely random function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ will do the trick). However, let's quickly try to understand these parameters. Let's imagine that $\epsilon = 0.1$ is some constant for now, and that $k \ll n$. Then we have $m = k + d + O(1)$, and $d = \log(n) + O(1)$. Note that $m = k + d$ should really be the best we could hope for: we have k bits of randomness coming from the source, and d bits of randomness coming from the seed, so intuitively $k + d$ bits of randomness is the best we should hope for coming out, from an information-theoretic perspective. Given that we have $m \approx k + d$, this theorem says that we can take the seed to be of length about $\log n$. In particular, since we can exhaust over all possible $\log n + O(1)$-length seeds in time $O(n)$, if we had such an extractor (and if we could efficiently apply it), we could effectively run a randomized algorithm with only a weak source, instead of a uniformly random source, with only an $O(n)$-time blow-up.

One important thing to note is that Theorem 1 above is *non-constructive.* That is, it's an existence result, but it doesn't tell us how to find this function Ext, much less how to efficiently apply it. Thus, there has been a great deal of work to come up with *explicit* constructions of extractors, and coming up with an optimal one is still an open question. However, there are several very nice constructions that do quite well. For example, the construction of [GUV09] can get seed length $d = O(\log n)$ and $m = 0.99 \cdot k$ (with $\epsilon = 0.01$, say). However, it remains an open problem to get seed length $O(\log n)$ and output length $m = k + O(1)$, even for costant ϵ. Later, after we discuss *expanders,* we'll see a way to get an extractor that has decent parameters (in the sense that $d + k = O(m)$), albeit a quite long seed length.

3 Expander graphs

Another useful type of pseudorandom object is an *expander graph.* There are several different ways to define expander graphs, but intuitively an expander graph is a sparse (aka, constant-degree) graph G that "behaves like the complete graph" in the sense that:

- For any set S of the vertices, there are lots of edges between S and its complement, and there are lots of vertices in the neighborhood of S.
- Random walks on G mix quickly.

As mentioned above, there are several definitions of expander graphs. For today, we're going to go with a definition of a *spectral* expander.

Let $G = (V, E)$ be an undirected, unweighted, regular graph with degree Δ, with N vertices. Let M be the normalized adjacency matrix of G, so M_{ij} is $1/\Delta$ if $\{i, j\} \in E$, and 0 otherwise. Since M is a symmetric matrix, it has an eigenvalue decomposition with real eigenvalues:

$$M = U^T \Lambda U,$$

where U is orthogonal and Λ is diagonal with the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ on the diagonal.

Observation 5. *With the set-up above,* $\lambda_1 = 1$.

Proof. We have $M \cdot \mathbf{1} = \mathbf{1}$, so $\mathbf{1}$ (the all-ones vector) is an eigenvector with eigenvalue 1. To see that it can't be any larger, suppose that $Mx = \lambda x$ for some eigenvalue-eigenvector pair (λ, x). Suppose that $x_i = \|x\|_\infty$. Then the equation above reads

$$\lambda |x_i| = \left| \sum_j M_{ij} x_j \right| \leq \|x\|_\infty = |x_i|,$$

where we have used the fact that $\sum_j M_{ij} = 1$ and $M_{ij} \geq 0$ for all i, j. Solving, we have

$$\lambda \leq 1.$$

□

Given that the largest eigenvalue is 1, we can ask "what is the second-largest eigenvalue?" We define this to be the *expansion* of G:

Definition 6. *Let $G = (V, E)$ be as above. Then the parameter $\lambda(G) := \max\{\lambda_2, |\lambda_n|\}$ is called the* expansion parameter *of G.*

As we will see below, if the expansion parameter is small (that is, if there is a big gap between the largest and second-largest eigenvalue), then G is a good expander graph in the sense that random walks mix quickly. Intuitively, this makes sense: the complete graph G has $\lambda(G) = 0$ (since the matrix M is the all-ones matrix, which has rank 1), and of course random walks mix immediately on the complete graph. The smaller $\lambda(G)$ is, the more G "behaves like the complete graph."

Theorem 2. *Let $\{X_t\}$ be a random walk on a graph $G = (V, E)$ as above. The the (unique) stationary distribution of $\{X_t\}$, π, is the uniform distribution, and*

$$\|\pi - P_i^t\| \leq \lambda(G)^t \sqrt{N}$$

for all vertices $i \in V$. (Recall the notation that P_i^t denotes the distribution of X_t given that $X_0 - i$).

In particular, Theorem 2 implies that $\tau_{mix} = O(\log n)$ whenever $\lambda(G) < 0.99$ (or any constant less than 1).

Proof. Let's think about M as the transition matrix for a Markov chain (which a random walk on G is). The statement that $M \cdot \mathbf{1} = \mathbf{1}$ that we saw above precisely means that the stationary distribution π of this walk is uniform. Let σ be any distribution on the vertices of G, and we think of $\sigma \in \mathbb{R}^N$ as a vector as usual (so σ_i is the probability of i under σ). Then we can write $\sigma = \pi + v$ for some vector $v \perp \pi$. (Indeed, we have $v = \sigma - \pi$, and

$$\sum_i v_i = \sum_i (\sigma_i - \pi_i) = \sum_i \sigma_i - \sum_i \pi_i = 1 - 1 = 0$$

so using the fact that π is the uniform distribution, we conclude that $v \perp \pi$).

We want to bound $\|\pi - M^t\sigma\|$, since $M^t\sigma$ is the distribution of X_t, assuming that $X_0 \sim \sigma$. But we have

$$\pi - M^t\sigma = M^t(\pi - \sigma) = M^t v,$$

where we have used that $M\pi = \pi$. Since $v \perp \pi$, and π is the top eigenvector of M, we have

$$\|M^t v\|_2 \leq \lambda(G)^t \|v\|_2.$$

Thus,

$$\begin{aligned}
\|\pi - M^t\sigma\|_{TV} &= \frac{1}{2}\|\pi - M^t\sigma\|_1 \\
&\leq \frac{\sqrt{N}}{2}\|\pi - M^t\sigma\|_2 \\
&\leq \frac{\sqrt{N}}{2}\lambda(G)^t\|v\|_2 \\
&\leq \frac{\sqrt{N}}{2}\lambda(G)^t\|v\|_1 \\
&\leq \sqrt{N}\lambda(G)^t,
\end{aligned}$$

where above we used the definition of total variation distance in the first line; Cauchy-Schwarz in the second line; our previous calculation in the third line; the fact that the ℓ_2 distance is always smaller than the ℓ_1 distance in the fourth line; and the fact that $\|v\|_1 \leq \|\pi\|_1 + \|\sigma\|_1 = 2$ in the final line. □

Thus, as long as G is a good expander—meaning that $\lambda(G) < 0.99$—after about $\log(N)$ steps of a random walk, we will be close to the uniform distribution.

You might be wondering if it is possible to come up with such a graph, and whether we can do so explicitly. The very best we can hope for is $\lambda(G) \approx \frac{2}{\Delta}\sqrt{\Delta - 1}$ (this is called the *Alon-Boppana bound).*A graph that satisfies $\lambda(G) \leq \frac{2}{\Delta}\sqrt{\Delta - 1}$ is called *Ramanujan.* A random constant-degree graph is a good expander with high probability. Moreover, there are explicit constructions of

good expanders, and even explicit constructions of Ramanujan graphs for certain parameter settings (although it is still open to construct, say, an infinite family of degree-7 (non-bipartite) Ramanujan graphs). See [HLW06] for a great survey of expander graphs.

At this point, we are done with the stuff for the pre-lecture videos. The material after this point is meant as a reference for after class.

4 Extractors from Expanders

Expander graphs have many uses in pseudorandomness. One of them is as an extractor, which we defined earlier.

Theorem 3. *Let $\epsilon > 0$. For all n and $k \leq n$, there is an explicit (k, ϵ)-extractor, $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ so that $m = n$ and $d = O(n - k + \log(1/\epsilon))$.*

Before we prove this theorem, let's take a quick look at the parameters. This is certainly not as good as the existential result of Theorem 1, but it's not bad. In particular, the amount of nearly-random bits we get out ($m = n$) is at least big-Omega of the amount of randomness we put in ($k + d = O(n)$). The seed length is quite long compared to Theorem 1 though, and in particular we can't exhaust over all possible seeds like we could for a seed of length $O(\log n)$.

Proof of Theorem 3. Let $N = 2^n$, and let $G = (V, E)$ be an expander graph with N vertices of degree $\Delta = O(1)$ and with $\lambda(G) \leq 1/2$. We're going to use the fact (that we did not prove) that there is an explicit construction of such a graph.

Our extractor Ext will be the following. Let $d = \log(\Delta) \cdot \ell$, for $\ell = n/2 - k/2 + \log(1/\epsilon) + 1$, and treat the seed as ℓ numbers in $\{1, 2, \ldots, \Delta\}$. Call these numbers $a_1, a_2, \ldots, a_\ell$.

Now choose $x \sim X$, where X is the k-source on $\{0,1\}^n$ that is the input for the extractor. Since $x \in \{0,1\}^n$, we can treat x as a vertex in V. Now consider the random walk where we start at x, then step to x's a_1'st neighbor (call it $x^{(1)}$), then step to $x^{(1)}$'s a_2'nd neighbor (call it $x^{(2)}$), and so on. That is, the random numbers $a_1, a_2, \ldots$ tell us how to take our random walk.

Once we have run out of random numbers, we are at $x^{(\ell)} \in V$. Interpreting $x^{(\ell)}$ as an element of $\{0,1\}^n$, we will now output $x^{(\ell)}$. That is, with the notation above, we have

$$\mathsf{Ext}(x; a_1, a_2, \ldots, a_\ell) = x^{(\ell)}.$$

We claim that the distribution of $x^{(\ell)}$ is ϵ-close to uniform. To see this, let $\sigma \in \mathbb{R}^N$ denote the vector corresponding to the distribution of X. Since X is a k-source, $\|\sigma\|_\infty \leq 2^{-k}$. Then the distribution of $x^{(\ell)}$ is distributed like $M^\ell \sigma$, where M is the normalized adjacency matrix for G.

As with the computation we did above in the proof of Theorem 2, we have

$$\begin{aligned} \|M^\ell \sigma - \pi\|_{TV} &\leq \frac{\sqrt{N}}{2} \|M^\ell \sigma - \pi\|_2 \\ &\leq \frac{\sqrt{N}}{2} \lambda(G)^\ell \|\sigma - \pi\|_2 \end{aligned}$$

Now, we can bound

$$\|\sigma - \pi\|_2 \leq \|\sigma\|_2 + \|\pi\|_2 \leq 2^{-k/2} + 2^{-n/2}.$$

This is because $\|\pi\|_2 = 2^{-n/2}$ (since π is the uniform distribution on 2^n things), and

$$\|\sigma\|_2^2 = \sum_i \sigma_i^2 \leq \|\sigma\|_\infty \sum_i \sigma_i = \|\sigma\|_\infty \leq 2^{-k}$$

which implies that

$$\|\sigma\|_2 \leq 2^{-k/2}.$$

Thus,

$$\begin{aligned}
\|M^\ell \sigma - \pi\|_{TV} &\leq \frac{\sqrt{N}}{2} \lambda(G)^\ell 2^{-k/2+1} \\
&\leq 2^{n/2-1} \cdot 2^{-\ell} \cdot 2^{-k/2+1} \\
&= 2^{n/2-1-(n/2-k/2+\log(1/\epsilon)+1)-k/2+1} \\
&= 2^{-\log(1/\epsilon)} \\
&= \epsilon.
\end{aligned}$$

using the definition of ℓ. Therefore, the output of $\mathsf{Ext}(X, U_d)$ is in fact ϵ-close to uniform! □

References

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):1–34, 2009.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.