Notes for Lecture 27

These notes follow quite literally the notes that I wrote for the PCMI school on computational complexity in Summer 2000. There is some redundancy between these notes and the content of past lectures.

See [Nis96, Gol99] to find references and proper credits about the material in this lecture.

1 Use of Weak Random Sources

Suppose that we have a probabilistic algorithm $A(\cdot,\cdot)$ that on inputs of length n runs in time $T(n)$ and uses $m(n)$ random bits. Instead of a perfect source of randomness, we assume that we have a source that produces an output containing some "impredictability," but that can still be very far from uniform. A very general way of modeling this source is to assume that on input 1^N it outputs a string in $\{0,1\}^N$, and that the output string has "some randomness" (a notion that we will formalize and quantify in a moment). Typically, a good way to quantify the amount of randomness, or impredictability, in a distribution, is to compute its (Shannon) entropy. For a random variable X whose range is $\{0,1\}^N$, its entropy is defined as $\mathcal{H}(X) = \sum_{a\in\{0,1\}^N} \mathbf{Pr}[X=a]\log(1/\mathbf{Pr}[X=a])$,

Shannon entropy is a very good measure in cases where one is allowed to take multiple samples from the distribution, but in our setting this is not the best measure. Consider for example a distribution X such that $X = (0,0,\cdots,0)$ with probability $1-1/\sqrt{N}$, and it is uniform with probability $1/\sqrt{N}$. Then its Shannon entropy is about $\sqrt{N}$, which is quite high, yet it is almost always a useless string of zeroes. It is a good intuition to think that the amount of randomness contained in the outcome a of a random variable X is $\log 1/\mathbf{Pr}[X=a]$. If X has Shannon entropy k, then *on average*, when we sample from X we get a value of "randomness" k, however it can be the case that with very high probability we get almost zero randomness, and with low probability we get high randomness. We would rather have a measure of randomness that guarantees to have almost always, or, even better, always, high randomness. This motivates the definition of *min-entropy*: a random variable X has min-entropy at least k if for every a in the range of X we have $\mathbf{Pr}[X=a] \le 1/2^k$. That is, the min-entropy of X is $\min_a\{\log 1/\mathbf{Pr}[X=a]\}$.

Definition 1 *A random variable with range $\{0,1\}^N$ having min-entropy at least k will be called a (N,k)-source.*

Given one access to a (N,k) source, we would like to be able to simulate any probabilistic algorithm that uses m random bits, where m is close to k. If the simulation is "black box" and takes time T, one can argue that $m \le k + O(\log T)$. We will not define formally what a black-box simulation is, but we will develop simulations that are black box, so it will come as no surprise that our simulations will work only for m smaller than k, in fact only for m smaller than $k^{1/3}$. (This is partly due to oversimplifications in the analysis; one could get $k^{.99}$ with almost the same proof.)

2 Extractors

An extractor is a function that transforms a (N, k) source into an almost uniform distribution. The transformation is done by using a (typically very small) number of additional random bits.

Formally, we have the following definition.

Definition 2 *For two random variables X and Y with range $\{0,1\}^m$, their variational distance is defined as $||X - Y|| = \max_{S \subseteq \{0,1\}^m} \{|\mathbf{Pr}[X \in S] - \mathbf{Pr}[Y \in S]|\}$. We say that two random variables are ϵ-close if their variational distance is at most ϵ.*

Definition 3 *A function $Ext : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is a (k, ϵ) extractor if for any (N, k) source X we have that $Ext(X, U_l)$ is ϵ-close to uniform, where U_l is the uniform distribution over $\{0,1\}^l$.*

Equivalently, if $Ext : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is a (k, ϵ) extractor, then for every distribution X ranging over $\{0,1\}^N$ of min-entropy k, and for every $S \subseteq \{0,1\}^m$, we have

$$|\mathbf{Pr}_{a \in X, z \in \{0,1\}^t}[Ext(a, z) \in S] - \mathbf{Pr}_{r \in \{0,1\}^m}[r \in S]| \leq \epsilon$$

3 Applications

See [Nis96] for an extensive survey. Here we present only one application. Another notable application is the construction of expanders.

Suppose that $A(\cdot, \cdot)$ is a probabilistic algorithm that on an input of length n uses $m(n)$ random bits, and suppose that for every x we have $\mathbf{Pr}_r[A(x, r) = \text{ right answer }] \geq 3/4$. Suppose $Ext : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ is a $(k, 1/4)$-extractor.

Consider the following algorithm A': on input $x \in \{0,1\}^n$ and weakly random $a \in \{0,1\}^N$, A' computes $A(x, Ext(a, z))$ for every $z \in \{0,1\}^t$, and then it outputs the answer that appears the majority of such 2^t times. We want to argue that A' is correct with high probability if a is sampled from a weak random source of entropy slightly higher than k. Let us fix the input x. Consider the set B of strings $a \in \{0,1\}^N$ for which the algorithm A' makes a mistake:

$$B = \{a : \mathbf{Pr}_{z \in \{0,1\}^t}[A(x, Ext(a, z)) = \text{ right answer }] < 1/2\}$$

Consider the random variable X that is uniformly distributed over B (clearly, X has min-entropy $\log B$). Then we have

$$\mathbf{Pr}_{a \in X, z \in \{0,1\}^t}[A(x, Ext(a, z)) = \text{ right answer }] < 1/2$$

and so

$$|\mathbf{Pr}_{a \in X, z \in \{0,1\}^t}[A(x, Ext(a, z)) = \text{ right answer }] - \mathbf{Pr}_r[A(x, r) = \text{ right answer }]| > 1/4$$

and then it follows form the property of Ext that X must have min-entropy less than k, that is $|B| \leq 2^k$.

Let now X be a $(N, k+2)$-source, and let us execute algorithm A' using X. Then

$$\mathbf{Pr}_{a\in X, z\in\{0,1\}^t}[A(x, Ext(a,z)) = \text{ right answer }] = 1 - \mathbf{Pr}_{a\in X}[a \in B] \geq 3/4$$

More generally

Theorem 1 *Suppose A is a probabilistic algorithm running in time $T_A(n)$ and using $m(n)$ random bits on inputs of length n. Suppose we have for every $m(n)$ a construction of a $(k(n), 1/4)$-extractor $Ext_n : \{0,1\}^N \times \{0,1\}^{t(n)} \to \{0,1\}^{m(n)}$ running in $T_E(n)$ time. Then A can be simulated in time $2^t(T_A + T_E)$ using one sample from a $(N, k+2)$ source.*

4 An Extractor from Nisan-Wigderson

This is a simplified presentation of results in [Tre01] (see also [RRV99, ISW00]).

Let $\mathcal{C} : \{0,1\}^N \to \{0,1\}^{\bar{N}}$ be a polynomial time computable error-correcting code such that any ball of radius at most $1/2 - \delta$ contains at most $1/\delta^2$ codewords. Such a code exists with $\bar{n} = \text{poly}(n, 1/\delta)$.

For a string $x \in \{0,1\}^{\bar{N}}$, let $< x >: \{0,1\}^{\log \bar{N}} \to \{0,1\}$ be the function whose truth table is x. Let $l = \log \bar{N}$, and let $\mathcal{S} = (S_1, \ldots, S_m)$ be a $(l, \log m)$ design over $[t]$. Then consider the procedure $ExtNW : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ defined as

$$ExtNW_{\mathcal{C},\mathcal{S}}(x, z) = NW_{<\mathcal{C}(x)>,\mathcal{S}}(z) \ .$$

That is, $ExtNW$ first encodes its first input using an error-correcting code, then views it as a function, and finally applies the Nisan-Wigderson construction to such a function, using the second input as a seed.

Lemma 2 *For sufficiently large m and for $\epsilon > 2^{-m^2}$, $ExtNW_{\mathcal{C},\mathcal{S}}$ is a $(m^3, 2\epsilon)$-extractor.*

PROOF: Fix a random variable X of min-entropy m^3 and a function $D : \{0,1\}^m \to \{0,1\}$; we will argue that

$$|\mathbf{Pr}[D(r) = 1] - \mathbf{Pr}_{a\in X, z\in\{0,1\}^t}[D(ExtNW(a,z)) = 1]| \leq 2\epsilon$$

Let us call a value a bad if it happens that

$$|\mathbf{Pr}[D(r) = 1] - \mathbf{Pr}_{z\in\{0,1\}^t}[D(ExtNW(a,z)) = 1]| > \epsilon$$

and let us call B the set of bad a. When a is bad, it follows that there is a circuit C of size $O(m^2)$ such that either $D(C())$ or its complement agrees with a on a fraction $1/2 + \epsilon/m$ of its entries. Therefore, a is totally specified by D, C, and $2\log(m/\epsilon)$ additional bits (once we have D and C, we know that the encoding of a sits in a given sphere of radius $1/2 - \epsilon/m$, together with at most other $(m/\epsilon)^2$ codewords). Therefore, for a fixed D, the size of B is upper bounded by the number of circuits of size $O(m^2)$, that is $2^{O(m^2 \log m)}$, times $(m/\epsilon)^2$, times 2. The total is $2^{O(m^2 \log m)}$. The probability that an element a taken from X belongs to B is therefore at most $2^{-m^3} \cdot 2^{O(m^2 \log m)} < \epsilon$ for sufficiently large m. We then have

$$
\begin{aligned}
& |\mathbf{Pr}[D(r)=1] - \mathbf{Pr}_{a \in X, z \in \{0,1\}^t}[D(ExtNW(a,z))=1]| \\
\leq \; & \sum_a \mathbf{Pr}[X=a] \left| \mathbf{Pr}[D(r)=1] - \mathbf{Pr}_{z \in \{0,1\}^t}[D(ExtNW(a,z))=1] \right| \\
\leq \; & \mathbf{Pr}[X \in B] + \epsilon \leq 2\epsilon
\end{aligned}
$$

$\square$

Theorem 3 *Fix a constant ϵ; for every N and $k = N^{\Omega(1)}$ there is a polynomial-time computable (k,ϵ)-extractor $Ext : \{0,1\}^N \times \{0,1\}^t \to \{0,1\}^m$ where $m = k^{1/3}$ and $t = O(\log N)$.*

References

[Gol99] O. Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness.* Springer-Verlag, 1999.

[ISW00] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 1–10, 2000.

[Nis96] N. Nisan. Extracting randomness: How and why. In *Proceedings of the 11th IEEE Conference on Computational Complexity*, pages 44–58, 1996.

[RRV99] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, pages 149–158, 1999.

[Tre01] L. Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, 2001.