Lecture Notes 12

29 Pseudorandom Generators

Definition: An ensemble $X = \{X_n\}_{n\in\mathbb{N}}$ is *pseudorandom* if X, U are indistinguishable in polynomial time, where $U = \{U_n\}_{n\in\mathbb{N}}$ is the uniform ensemble.

Thus, X is pseudorandom if it "looks" the same to all probabilistic polynomial time algorithms.

Definition: A *pseudorandom generator* is a deterministic polynomial time function G that satisfies two properties:

1. G maps strings of length n to strings of length $\ell(n) > n$. $\ell(n)$ is called the *expansion factor*.
2. $\{G(U_n)\}_{n\in\mathbb{N}}$ is pseudorandom.

We remark that if G is a pseudorandom generator, then $G(U_n)$ is not statistically close to $U_{\ell(n)}$. To see this, let $R_G = \{G(x) \mid x \in \{0,1\}^n\}$ be the range of G. Clearly, $|R_G| \leq 2^n$, and for all $y \notin R_G$, $\Pr[G(U_n) = y] = 0$. On the other hand, for the uniform ensemble, $\Pr[U_{\ell(n)} = y] = \frac{1}{2^\ell}$. Hence, the statistical difference

$$\begin{aligned}
\Delta(\ell(n)) &= \frac{1}{2} \sum_{\alpha\in\{0,1\}^{\ell(n)}} |\Pr[G(U_n) = \alpha] - \Pr[U_{\ell(n)} = \alpha]| \\
&\geq \frac{1}{2} \sum_{\alpha\in R_G} |\Pr[G(U_n) = \alpha] - \Pr[U_{\ell(n)} = \alpha]| \\
&= \frac{1}{2} \sum_{\alpha\in R_G} |0 - \frac{1}{2^\ell}| \\
&= \frac{1}{2} \cdot \frac{2^\ell - 2^n}{2^\ell} \geq \frac{1}{4},
\end{aligned}$$

is not negligible, so $G(U_n)$ and $U_{\ell(n)}$ are not statistically close.

We now describe how to build a pseudorandom number generator G with polynomial expansion factor starting from a generator G_1 with expansion factor $\ell(n) = n + 1$.

Fix a polynomial $p(n)$. For $s \in \{0,1\}^n$, write the length-$(n+1)$ string $G_1(s)$ as $\sigma s'$, where $|\sigma| = 1$ and $|s'| = n$. On input s, iteratively define the sequences $s_0, s_1, s_2, \ldots, s_{p(n)}$ and $\sigma_1, \sigma_2, \ldots, \sigma_{p(n)}$ as follows:

$$\begin{aligned}
s_0 &= s \\
\sigma_i s_i &= G_1(s_{i-1}), \text{ for } i = 0, 1, 2, \ldots, p(n) - 1.
\end{aligned}$$

The output of $G(s)$ is the sequence $\sigma_1\sigma_2 \ldots \sigma_{p(n)}$. $G(s)$ is easily computed in polynomial time by a simple iterative program that calls G_1 a total of $p(n)$ times.

Theorem 1 *If G_1 is pseudorandom, then so is G.*

Proof is by a hybrid argument. We let hybrid H_n^k consist of k uniform random bits followed by the first $p(n) - k$ bits of $G(s_0)$, which we write as $G(s_o)\colon [1, p(n) - k]$. In symbols,

$$H_n^k = U_k \cdot G(U_n)\colon [1, p(n) - k].$$

Clearly, $H_n^0 = G(U_n)$ and $H_n^{p(n)} = U_{p(n)}$.

Suppose D distinguishes $G(U_n)$ from $U_{p(n)}$ with absolute probability difference $\delta(n)$. Then for some k, D distinguishes H_n^k from H_n^{k+1} with absolute probability difference $\geq \delta(n)/p(n)$.

We now describe an algorithm D' that attempts to distinguish $G_1(U_n)$ from U_{n+1}. On length-$(n+1)$ input α, D' does the following:

1. Write $\alpha = \tau \cdot \alpha'$, where $|\tau| = 1$ and $|\alpha'| = n$.
2. Choose index k uniformly from $\{0, 1, \ldots, p(n) - 1\}$.
3. Choose a uniformly distributed string β of length k.
4. Construct $y = \beta \cdot \tau \cdot G(\alpha')\colon [1, p(n) - k - 1]$.
5, Compute and output $D(y)$.

If α is uniformly distributed, then τ and α' are both uniformly distributed, so $y = H_n^{k+1}$. On the other hand, if $\alpha = G_1(s_0)$, where s_0 is uniformly distributed, then $\tau = \sigma_1$ and $\alpha' = s_1$, so $y = H_n^k$. This is because

$$G(s_0)\colon [1, p(n) - k] = \tau \cdot G(s_1)\colon [1, p(n) - k - 1]$$

Hence, D' distinguishes $G_1(U_n)$ from U_{n+1} with absolute probability difference $\geq \delta(n)/p(n)$.

We omit the remaining details of showing how this leads to a contradiction of the assumption that G is not pseudorandom.

30 Unpredictability

Our formal definition of pseudorandomness is based on the indistinguishability of an entire polynomial-length generated sequence from a uniformly distributed random sequence. However, the traditional notion of a pseudorandom generator is based on repeated experiments. The output bits $x_1, x_2, \ldots$ are assumed to be generated one at a time. The generator is called pseudorandom if each x_i "appears" to result from an independent and uniformly distributed random event such as the flip of a fair coin.

The notion of "appears" is can be captured in terms of unpredictability. We say that x_{i+1} is unpredictable if no polynomial time algorithm that attempts to guess it is correct with more than a tiny advantage over chance, even given all of the prior bits $x_1, \ldots, x_i$.

More formally, a *predictor* is a p.p.t. algorithm A that is allowed to read the input sequence x a bit at a time in order. After reading bit i, the algorithm can choose to output a guess b and halt, or it can continue. In any case, it must halt and emit a guess after reading the next-to-last bit of x. Let k be the last bit read by A. Then A is *correct* if $k < |x|$ and $b = x_{k+1}$. In addition to the input x, which A is allowed to read only a bit at a time, A is also given an input 1^n, where $n = |x|$. This way, A can determine the length of x without having to read it all.

Notation: The textbook uses the notation $\mathrm{next}_A(x)$ to denote the next bit of x following the last bit that A read. The intent is that the event $[A(1^{|X_n|}, X_n) = \mathrm{next}_A(X_n)]$ should mean that a string x is chosen according to the distribution X_n, A is run on inputs 1^n and x, A reads the first k bits of x for some k and outputs b, and $b = x_{k+1}$, the "next" bit of x. That is, the event is that A correctly predicts some bit of a randomly chosen x from distribution X_n.

A better notation would make k explicit. For example, we could pretend that A outputs a pair (k, b) with the meaning that k is the index of the last bit of x that A read, and b is A's prediction for x_{k+1}. We could then define $\mathrm{next}_A(x) = \{(k, x_{k+1}) \mid k \in [0, n-1]\}$. Now, A correctly predicts the next bit if $A(1^{|x|}, x) = (k, b)$ and $(k, b) \in \mathrm{next}_A(x)$.

Definition: An ensemble $\{X_n\}_{n\in\mathbb{N}}$ is called *unpredictable in polynomial time* if for every p.p.t. A, every positive polynomial $p(\cdot)$, and all sufficiently large n,

$$\Pr[A(1^{|x|}, x) \in \mathrm{next}_A(x)] < \frac{1}{2} + \frac{1}{p(n)}.$$

Theorem 2 *An ensemble X is pseudorandom if and only if it is unpredictable in polynomial time.*

Proof:
($\Rightarrow$) The theorem in the forward direction is straightforward. We sketch the general ideas and leave the details to the reader.

If there were a predictor A for X, then a distinguisher D is easily built. Namely, $D(x)$ outputs 1 iff $A(1^{|x|}, x)$ correctly predicts the next bit. If x comes from X, $D(x)$ will output 1 with probability at least $\frac{1}{2} + \frac{1}{p(n)}$, but if x comes from U, then clearly $D(x)$ will output 1 with probability exactly $\frac{1}{2}$. Hence, D successfully distinguishes X from U.

($\Leftarrow$) The theorem in the reverse direction is proved by another hybrid argument. We sketch a few of the main ideas. Assume X is both unpredictable but not pseudorandom. Then there is a distinguisher D such that

$$|\Pr[D(X_n) = 1] - \Pr[D(U_n) = 1]| \geq \frac{1}{p(n)}$$

for infinitely many n. We may without loss of generality drop the absolute value brackets and assume that

$$\Pr[D(X_n) = 1] - \Pr[D(U_n) = 1] \geq \frac{1}{p(n)}$$

for infinitely many n. The reasoning is that either $\Pr[D(X_n) = 1] \geq \Pr[D(U_n) = 1]$ for infinitely many n, or $\Pr[D(X_n) = 1] \leq \Pr[D(U_n) = 1]$ for infinitely many n. If the latter, then $\Pr[\bar{D}(X_n) = 1] \geq \Pr[\bar{D}(U_n) = 1]$ for the algorithm $\bar{D}$ that is identical to D except that it complements the output.

We build a next-bit predictor A. Let hybrid H_n^k consist of the first k bits from X_n followed by the last $n-k$ bits from U_n. Then $H_n^n = X_n$ and $H_n^0 = U_n$. The predictor $A(1^{|x|}, x)$ guesses a number $k \in [0, |x|-1]$, reads only the first k bits of x, and constructs the string $y = x_1, \ldots, x_k, u_{k+1}, \ldots, u_n$, where the u_j's are uniformly distributed random bits. It then runs $D(y)$. If $D(y) = 1$, then A predicts bit $k+1$ to be u_{k+1}. Otherwise, A predicts bit $k+1$ to be $\neg u_{k+1}$ (the complement of u_{k+1}).

We omit the non-trivial analysis needed to show that algorithm A has a sufficient advantage as a next-bit predictor to contradict the assumption that X is unpredictable. ■

31 Pseudorandom Generators and One-Way Functions

We now show that the existence of pseudorandom generators implies the existence of one-way functions.

Theorem 3 *Let G be a pseudorandom generator with expansion factor $\ell(n) = 2n$. Define the function $f(x, y) = G(x)$ for all $|x| = |y|$. Then f is a strongly one-way function.*

Proof: Suppose f is not strongly one-way. Let A be an inverter for $f(U_{2n})$ with success probability at least $\frac{1}{p(n)}$ for infinitely many n. We construct a distinguisher D that distinguishes $G(U_n)$ from U_{2n} on those same n.

$D(\alpha)$ uses A to attempt to find β such that $f(\beta) = \alpha$. If A succeeds, then D outputs 1; otherwise D outputs 0. Since $f(U_{2n}) = G(U_n)$, then

$$\Pr[D(G(U_n)) = 1] = \Pr[f(A(f(U_{2n}))) = f(U_{2n})] \geq \frac{1}{p(n)}. \tag{1}$$

On the other hand,

$$\Pr[D(U_{2n}) = 1] = \Pr[f(A(U_{2n})) = U_{2n}] \leq \frac{1}{2^n}. \tag{2}$$

This is because $f(x, y)$ depends only on x, so the range of f on pairs of length-n inputs has size $\leq 2^n$. Since $f(A(U_{2n}))$ is in the range of f, the probability that U_{2n} is in the range, much less actually equal to $f(A(U_{2n}))$, is at most 2^{-n}. Subtracting 2 from 1 gives

$$\Pr[D(G(U_n)) = 1] - \Pr[D(U_{2n}) = 1] \geq \frac{1}{p(n)} - \frac{1}{2^n} \geq \frac{1}{2p(n)}. \tag{3}$$

Thus, D distinguishes $G(U_n)$ from U_{2n} for infinitely many n, contradicting the assumption that G is a pseudorandom generator. ■