6.895 Randomness and Computation May 3, 2006

Lecture 21

Lecturer: Ronitt Rubinfeld *Scribe: Benjamin Rossman*

Definition 1 (Computational indistinguishability) *Let $X = (X_n)$ and $Y = (Y_n)$ be sequences of random variables on $\{0,1\}^n$. We say X and Y are* $\epsilon(n)$-indistinguishable for time $t(n)$ *if for every probabilistic algorithm T running in time $t(n)$,*

$$|\Pr[T(X_n) = 1] - \Pr[T(Y_n) = 1]| \leq \epsilon(n)$$

for all large enough n. The quantity $|\Pr[T(X_n) = 1] - \Pr[T(Y_n) = 1]|$ is called the advantage *of T; it is a measure of how much better T is than random guessing at distinguishing X_n from Y_n. We write $X \overset{c}{\equiv} Y$ if X and Y are $\frac{1}{n^c}$-indistinguishable for time n^c, for all $c > 0$. T's advantage is said to be* negligible *if it is $< \frac{1}{n^c}$ for all c.*

The following definition is due to Blum, Micali and Yao.

Definition 2 (Pseudorandom generator, or PRG) *A function $G : \{0,1\}^{\ell(n)} \longrightarrow \{0,1\}^n$ is a* PRG *if*

(1) $\ell(n) < n$

(2) $G(\mathscr{U}_{\ell(n)}) \overset{c}{\equiv} \mathscr{U}_n$

where $\mathscr{U}_n$ is the uniform distribution on $\{0,1\}^n$ and $G(\mathscr{U}_{\ell(n)})$ is the distribution on $\{0,1\}^n$ induced as the image under G of the uniform distribution $\mathscr{U}_{\ell(n)}$ on $\{0,1\}^{\ell(n)}$.

The function $\ell(n)$ is called the seed length *of G.*

G is efficient *if it is computable in time* poly(n) *(not* poly$(\ell(n))$*).*

It is pseudorandom against nonuniform time $t(n)$ *if $G(\mathscr{U}_{\ell(n)})$ and $\mathscr{U}_n$ are computationally indistinguishable with respect to probabilistic algorithms T that run in nonuniform polynomial time (i.e., T is computable by a non-uniform family of polynomial-size circuits).*

Definition 3 (BPP complexity class) *$L \in$ BPP if there is a p.p.t. (probabilistic polynomial time) algorithm A such that for all inputs x,*

- *if $x \in L$ then $\Pr[A$ accepts $x] \geq \frac{2}{3}$;*
- *if $x \notin L$ then $\Pr[A$ accepts $x] \leq \frac{1}{3}$.*

That is, A outputs the correct answer with probability $\geq \frac{2}{3}$. (A tolerates two-sided errors.)

Theorem 4 *If there exists an efficient PRG against nonuniform time n with seed length $\ell(n)$, then* BPP $\subseteq \bigcup_{c>0}$ DTIME$(2^{\ell(n^c)} n^c)$ *and in particular*

$$\ell(n) = O(\log n) \Longrightarrow \text{BPP} \subseteq \text{P}$$
$$\ell(n) = O(\log^c n) \Longrightarrow \text{BPP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$$
$$\ell(n) = O(n^\epsilon) \Longrightarrow \text{BPP} \subseteq \text{Subexponential Time.}$$

Note that BPP $\subseteq$ ExpTime since an exponential time algorithm can enumerate all seeds to a PRG and output the majority answer.

Proof Suppose $G : \{0,1\}^{\ell(n)} \longrightarrow \{0,1\}^n$ is a PRG against nonuniform time n whose runtime is $O(n^{c_1})$. Let A be a p.p.t. algorithm in BPP whose runtime is $O(n^{c_2})$. We define a deterministic algorithm $A' \in \text{DTIME}(2^{\ell(n^{c_2})}(n^{c_1} + n^{c_2}))$ equivalent to A as follows: run A on input x with random bits $G(s)$ for all seeds $s \in \{0,1\}^{\ell(n)}$, and output the majority answer.

Toward a contradiction, assume A' gives the wrong answer on input x. That is, $\Pr_{s \in \mathscr{U}_{\ell(n^{c_2})}}[A(x, G(s))$ is correct$] \leq \frac{1}{2}$. Since $A \in \text{BPP}$, we know $\Pr_{y \in \mathscr{U}_{n^{c_2}}}[A(x,y)$ is correct$] \geq \frac{2}{3}$. But now we have an efficiently computable test $T_{A,x}(*) := A(x,*)$ with advantage $\frac{1}{6}$. This contradicts the fact that G is a PRG. Therefore, A' is equivalent to A. We conclude that $\text{BPP} = \bigcup_{c>0} \text{DTIME}(2^{\ell(n^{c_2})}(n^{c_1} + n^{c_2}))$. ■

Remark In the proof of Theorem 4, it is enough to assume we have a PRG G such that $G(U_{\ell(n)})$ is computationally indistinguishable from $\mathscr{U}_n$ for *linear time algorithms* T. Note that the runtime of G has to be $\text{poly}(n^c)$, but isn't required to match the runtime of A.

It can be shown, via a probabilistic proof, that:

Theorem 5 *There exists a PRG against nonuniform time $t(n)$ with seed length $O(\log t(n))$*

Note that Theorem 5 says nothing about the efficiency of the PRG. The existence of an efficient PRG satisfying the condition of Theorem 5 implies BPP $\neq$ P, by Theorem 4.

Theorem 6 *If there exists an efficient PRG, then* P $\neq$ NP*.*

Proof Toward a contradiction, suppose $G : \{0,1\}^{\ell(n)} \longrightarrow \{0,1\}^n$ is an efficient PRG and assume P = NP. Define test $T(x)$ by

$$T(x) = \begin{cases} 0 & \text{if } \exists y \text{ s.t. } G(y) = 1, \\ 1 & \text{otherwise.} \end{cases}$$

T distinguishes distributions $G(\mathscr{U}_{\ell(n)})$ and $\mathscr{U}_n$ with advantage $\geq \frac{1}{2}$, as

$$\Pr[T(G(\mathscr{U}_{\ell(n)})) = 1] = 1,$$
$$\Pr[T(\mathscr{U}_n) = 1] \leq \frac{2^{\ell(n)}}{2^n} \leq \frac{1}{2} \text{ since } \ell(n) < n.$$

Notice that T is computable in NP, since a nondeterministic algorithm can guess y and then verify that $G(y) = 1$ in polynomial time. Since we are assuming P = NP, it follows that T is efficient. But this contradicts the assumption that G is a PRG, since T distinguishes $G(\mathscr{U}_{\ell(n)})$ from the uniform distribution $\mathscr{U}_n$. ■

In the previous lecture, we discussed three different notions of randomness. We now add a fourth: unpredictability.

Definition 7 (Next-bit unpredictability) *Let $\mathscr{X} = (X_1, \ldots, X_n)$ be a distribution on $\{0,1\}^n$. $\mathscr{X}$ is* next-bit unpredictable *if for every p.p.t. "predictor" algorithm P, there exists a negligible function $\epsilon(n)$ (where negligible means $\epsilon(n) = O(\frac{1}{n^c})$ for all $c > 0$) such that*

$$\Pr_{\substack{i \in_R [n] \\ \text{coins of } P}}[P(X_1, \ldots, X_{i-1}) = X_i] \leq \frac{1}{2} + \epsilon(n)$$

Surprisingly, next-bit unpredictability turns out to be an equivalent notion to pseudorandomness.

Theorem 8 *$\mathscr{X}$ is pseudorandom if, and only if, it is next-bit unpredictable.*

Proof ($\Longrightarrow$) Suppose P is not next-bit unpredictable. Then for some $c > 0$,

$$\Pr_{i \in_{\mathrm{R}} [n]}[P(X_1, \ldots, X_{i-1}) = X_i] > \frac{1}{2} + \frac{1}{n^c}.$$

In particular, there exists $i \in [n]$ such that

$$\Pr[P(X_1, \ldots, X_{i-1}) = X_i] > \frac{1}{2} + \frac{1}{n^c}.$$

We now define an efficient test $T(y_1, \ldots, y_n)$ by

$$T(y_1, \ldots, y_n) = \begin{cases} 0 & \text{if } P(y_1, \ldots, y_{i-1}) \neq y_i, \\ 1 & \text{if } P(y_1, \ldots, y_{i-1}) = y_i. \end{cases}$$

We have

$$\Pr_{y \in \mathscr{U}_n}[T(y) = 1] = \frac{1}{2}$$
$$\Pr_{y \in \mathscr{X}}[T(y) = 1] > \frac{1}{2} + \frac{1}{n^c}.$$

So T distinguishes between distributions $\mathscr{X}$ and $\mathscr{U}_n$ with advantage $> \frac{1}{n^c}$. Therefore, X is not pseudorandom.

($\Longleftarrow$) Suppose $\mathscr{X}$ is not pseudorandom. Then there is a p.p.t. algorithm T such that

$$\text{advantage}(T) = |\Pr[T(\mathscr{X}) = 1] - \Pr[T(\mathscr{U}_n) = 1]| > \frac{1}{n^c}.$$

Without loss of generality, we assume that $\Pr[T(\mathscr{X}) = 1] > \Pr[T(\mathscr{U}_n) = 1]$; for if the inequality goes the other way, then we substitute T with its complement.

We use a "hybrid argument" to construct a next-bit predictor algorithm. Let $U_1, \ldots, U_n$ be uniform independent random variables on $\{0, 1\}$, so that $\mathscr{U}_n = (U_1, \ldots, U_n)$. We define a sequence of distributions:

$$\begin{aligned}
\mathscr{D}_0 &= (U_1, \ldots, U_n) = \mathscr{U}_n \\
\mathscr{D}_1 &= (X_1, U_2, \ldots, U_n) \\
\mathscr{D}_2 &= (X_1, X_2, U_3, \ldots, U_n) \\
&\vdots \\
\mathscr{D}_i &= (X_1, \ldots, X_i, U_{i+1}, \ldots, U_n) \\
&\vdots \\
\mathscr{D}_n &= (X_1, \ldots, X_n) = \mathscr{X}.
\end{aligned}$$

Notice that

$$T(\mathscr{D}_{i-1}) = \frac{1}{2}\Big(T(\mathscr{D}_i) + T(X_1, \ldots, X_{i-1}, 1 - X_i, U_{i+1}, \ldots, U_n)\Big) \tag{$\star$}$$

Now, we have

$$\frac{1}{n^c} < \Pr[T(\mathscr{D}_n) = 1] - \Pr[T(\mathscr{D}_0) = 1] = \sum_{i \in [n]} \Pr[T(\mathscr{D}_i) = 1] - \Pr[T(\mathscr{D}_{i-1}) = 1].$$

Therefore, there exists $i \in [n]$ such that $\Pr[T(\mathscr{D}_i) = 1] - \Pr[T(\mathscr{D}_{i-1}) = 1] > \frac{1}{n^{c+1}}$.

We define p.p.t. "predictor" algorithm $P(x_1, \ldots, x_{i-1}, y_i, \ldots, y_n)$ with input bits $x_1, \ldots, x_{i-1}$ and random bits (coins) $y_i, \ldots, y_n \in_{\mathrm{R}} \{0, 1\}$ by

$$P(x_1, \ldots, x_{i-1}, y_i, \ldots, y_n) = \begin{cases} y_i & \text{if } T(x_1, \ldots, x_{i-1}, y_i, \ldots, y_n) = 1 \\ 1 - y_i & \text{otherwise.} \end{cases}$$

$$\begin{aligned}
&\Pr[P(X_1, \ldots, X_{i-1}, U_i, \ldots, U_n) = X_i] \\
&= \frac{1}{2}\Big(\Pr[P(X_1, \ldots, X_{i-1}, U_i, \ldots, U_n) = X_i \mid U_i = X_i] + \Pr[P(X_1, \ldots, X_{i-1}, U_i, \ldots, U_n) = X_i \mid U_i \neq X_i]\Big) \\
&= \frac{1}{2}\Big(\Pr[P(X_1, \ldots, X_i, U_{i+1}, \ldots, U_n) = X_i] + \Pr[P(X_1, \ldots, X_{i-1}, 1 - X_i, U_{i+1} \ldots, U_n) = X_i]\Big) \\
&= \frac{1}{2}\Big(\Pr[T(X_1, \ldots, X_i, U_{i+1} \ldots, U_n) = 1] + \Pr[T(X_1, \ldots, X_{i-1}, 1 - X_i, U_{i+1} \ldots, U_n) = 0]\Big) \\
&= \frac{1}{2}\Big(\Pr[T(\mathscr{D}_i) = 1] + \Big(1 - \Pr[T(X_1, \ldots, X_{i-1}, 1 - X_i, U_{i+1} \ldots, U_n) = 1]\Big)\Big) \\
&= \frac{1}{2} + \frac{1}{2}\Big(\Pr[T(\mathscr{D}_i) = 1] - \underbrace{\Pr[T(X_1, \ldots, X_{i-1}, 1 - X_i, U_{i+1} \ldots, U_n) = 1]}_{= 2\Pr[T(\mathscr{D}_{i-1}) = 1] - \Pr[T(\mathscr{D}_i) = 1] \text{ by } (\star)}\Big) \\
&= \frac{1}{2} + \Big(\Pr[T(\mathscr{D}_i) = 1] - \Pr[T(\mathscr{D}_{i-1} = 1)]\Big) \\
&> \frac{1}{2} + \frac{1}{n^{c+1}}.
\end{aligned}$$

By defining $P(x_1, \ldots, x_j) \in_{\mathrm{R}} \{0, 1\}$ for values of $j \in [n] - \{i\}$, we get

$$\begin{aligned}
&\Pr_{j \in_{\mathrm{R}} [n]}[P(X_1, \ldots, X_{j-1}) = X_j] \\
&= \frac{1}{n}\Big(\Pr[P(X_1, \ldots, X_{i-1}) = X_i] + \sum_{j \in_{\mathrm{R}} [n] - \{i\}} \Pr[P(X_1, \ldots, X_{j-1}) = X_j]\Big) > \frac{1}{n}\Big(\frac{n}{2} + \frac{1}{n^{c+1}}\Big) = \frac{1}{2} + \frac{1}{n^{c+2}}.
\end{aligned}$$

Thus, we have shown that $\mathscr{X}$ is not next-bit unpredictable. ■