Pseudorandomness – Lecture 14.

Instructor: Ramprasad.
Date: 2021-10-07
Lecture #: 14.

Agenda: - Intro to pseudorandom generators
- Hybrid argument.

What *is* a pseudorandom object/distribution?

An object that exhibits some property that makes an observer think it was picked randomly.

The stupid algo for max-cut — pairwise ~~i~~ independence.

$\bigoplus_{i \in S} x_i$ — ε-biased distributions

Emperical avg of samples — expander walk.

Any randomised algo running in time $O(n^2)$ — ??

Computational indistinguishability:

Two RVs X, Y taking values in $\{0,1\}^m$ is ε-c.i for a class $C = \{T : \{0,1\}^m \to \underset{\{0,1\}}{[-1,1]}\}$ of test functions if

$$\left| \mathbb{E}[T(X)] - \mathbb{E}[T(Y)] \right| \leq \varepsilon \text{ for all } T \in C.$$

That is, as far as "tests" from C are concerned, they behave roughly similarly whether they are fed X or Y.

PRG for a class $\mathcal{C}$: $(u, i_1, i_2, \ldots, i_t) \longmapsto (u_1, u_2, u_3, \ldots, u_{t+1})$

A map $G : \{0,1\}^d \to \{0,1\}^m$ is an ε-PRG for $\mathcal{C}$ if the RVs $\mathcal{U}_m$ and $G.(\mathcal{U}_d)$ are ε-comp. ind for $\mathcal{C}$. ie,

$$\left| \mathop{\mathbb{E}}_{X \sim \mathcal{U}_m}\left[T(X)\right] - \mathop{\mathbb{E}}_{Y \sim \mathcal{U}_d}\left[T(G(Y))\right] \right| \leq \varepsilon$$

for all $T \in \mathcal{C}$.

(Often, $\mathcal{C}$ corresponds to size m^2 circuits or subclasses of "efficient computation")

Again, we often care for families: $\{G_m : \{0,1\}^{d(m)} \to \{0,1\}^m\}$.

Desire: - Want d as small as possible
- Want $G_d(y)$ to be efficiently computable.

Defn: $\{G_m : \{0,1\}^{d(m)} \to \{0,1\}^m\}$ is $t(m)$-computable if there is an algorithm M s.t.

$M(1^m, x) = G_m(x)$
$\&\ M(1^m) = d(m)$, and M runs in time t

Thm: Suppose for all m, there is $t(m)$-comp. $1/8$-PRG $\{G_d : \{0,1\}^{d(m)} \to \{0,1\}^m\}$ for $\{\mathcal{C}_m\}$ where $\mathcal{C}_m$ are boolean fns computed by circuits of size $\leq m$.

Then, $\mathrm{BPP} \subseteq \bigcup_{c>0} \mathrm{DTIME}\left(2^{d(n^c)} \cdot (n^c + t(n^c))\right)$

Pf: A is a rand. algo running in time $\leq n^c = m$.

$\Rightarrow$ A uses $\leq m$ random bits.

$x \in L \Rightarrow \Pr_r[A(x,r)=1] \geq 2/3$

$x \notin L \Rightarrow \Pr_r[A(x,r)=1] \leq 1/3$

Algo B (input x):

- ▷ Build a circuit $T: \{0,1\}^m \to \{0,1\}$
  $T(r) = A(x,r)$
- ▷ Run over all $y \in \{0,1\}^{d(m)}$,
  compute $z = G(y)$.
  count $\#\, y$: $A(z) = 1$.
- ▷ Acc if this $\# > \frac{1}{2} \cdot 2^{d(m)}$.

PRG guarantee $\Rightarrow$ B is correct. $\square$.

Defn: $\{G_m : \{0,1\}^{d(m)} \to \{0,1\}^m\}$ is

- ▷ mildly explicit if it is $\mathrm{poly}(m, 2^{d(m)})$-computable
- ▷ fully explicit if it is $\mathrm{poly}(m)$ computable.

output the whole truth table in poly( )

Computing G on a specific seed.

Do PRG's exist at all?

Always check if a "random object works".

Thm: For any $m \in \mathbb{N}$ and $\varepsilon > 0$, there are lots of PRG (not explicit) $G: \{0,1\}^d \to \{0,1\}^m$ for size m circuits with seed length $d = O(\log m + \log \frac{1}{\varepsilon})$.

Pf: Pick $G: \{0,1\}^d \to \{0,1\}^m$ uniformly at random.
Fix a circuit T of size m,

$Z_i \sim \mathcal{U}_m$.

$$\underset{Y \sim \mathcal{U}_d}{\mathbb{E}}\left[T(G(Y))\right] = \frac{1}{2^d} \cdot \sum_{i=1}^{2^d} T(Z_i)$$

$\mu = \mathbb{E}[T(\mathcal{U})]$. Chernoff says you are within ε w.p $\geq 1 - \exp(-\varepsilon^2 \cdot 2^d)$

If G is <u>not</u> a PRG, then

$$\Pr_G\left[\exists T: \left|\mathbb{E}[T(\mathcal{U})] - \frac{1}{2^d} \cdot \sum_{i=1}^{d} T(Z_i)\right| > \varepsilon\right]$$

$$\leq \left(\#\text{ circuits of size} \leq m\right) \cdot \exp(-\varepsilon^2 \cdot 2^d)$$

$2^{O(m \log m)}$

$\therefore$ If $2^d \cdot \varepsilon^2 > 100 \cdot m \log m$

$\Rightarrow d = O(\log m + \log 1/\varepsilon)$ is suff. $\square$.

"Can we find hay in a haystack?"

Aspirational goal: Find an explicit PRG $G: \{0,1\}^d \to \{0,1\}^m$ for size m circuits with $O(\log m + \log 1/\varepsilon)$-seed length.

(we will take anything! $d = o(m)$)

A "simpler" requirement from a PRG

Defn: (Next bit unpredictable) X r.v on $\{0,1\}^m$ is (t,ε)-NBU if there is no circuit P of size $\leq t$ and no $i \in [m]$ with

$$\Pr_X \left[P(X_1 \ldots X_{i-1}) = X_i \right] \geq \frac{1}{2} + \varepsilon.$$

Given a prefix, guessing the next bit is hard.

Lemma: If $X \sim \{0,1\}^m$ is (t,ε)-pseudorandom, then X is $(t - O(1), \varepsilon)$ NBU.
Conversely, X is (t,ε)-NBU, then X is $(t, \varepsilon m)$-pseudorandom.

Pf: $(\Rightarrow)$: X was pseudorandom but next bit predictable
$\Rightarrow$ There is a circuit P and an index i
s.t $\Pr \left[P(X_1, \ldots, X_{i-1}) = X_i \right] \geq \frac{1}{2} + \varepsilon$

$$Q \begin{cases} \text{Algo: on input } z_1 \ldots z_m. \\ \quad \text{Accept if } z_i = P(z_1, \ldots, z_{i-1}) \end{cases}$$

$\mathbb{E}[Q(\mathcal{U}_m)] = 1/2.$ $\qquad$ $\mathbb{E}[Q(X)] \geq \frac{1}{2} + \varepsilon$

diff. by ε.

$(\Leftarrow)$: Given that X is (t,ε)-NBU
Want to show that $X \underset{\varepsilon m}{\overset{c.i}{\approx}} \mathcal{U}_m$.
Hybrid walk argument!

$Y_m = X$ $\quad$ $\boxed{x_1 \quad \cdots \quad x_{m-2}\ x_{m-1}\ x_m}$

Y_{m-1} $\quad$ $\boxed{x_1 \quad \cdots \quad x_{m-2}\ x_{m-1}\ u_m}$

Y_{m-2} $\quad$ $\boxed{x_1 \quad \cdots \quad x_{m-2}\ u_{m-1}\ u_m}$

$\vdots$

Y_1 $\quad$ $\boxed{x_1\ u_2 \quad \cdots \quad u_m}$

$Y_0 = U$ $\quad$ $\boxed{u_1 \quad \cdots \quad u_m}$

Aim: $X = Y_m \overset{c.i}{\approx}_\varepsilon Y_{m-1} \overset{c.i}{\approx}_\varepsilon Y_{m-2} \cdots \overset{c.i}{\approx}_\varepsilon Y_0$

$$\Rightarrow \quad X \overset{c.i}{\approx}_{m\varepsilon} U$$

Suppose $Y_0 \overset{c.i}{\not\approx}_{m\varepsilon} Y_m$. $\Rightarrow$ there is a P

s.t $\quad \mathbb{E}[P(Y_0)] - \mathbb{E}[P(Y_m)] \geq m\varepsilon.$

$$\Rightarrow \sum_{i=1}^{m} \mathbb{E}[P(Y_{i-1})] - \mathbb{E}[P(Y_i)] \geq m\varepsilon.$$

$$\Rightarrow \exists i : \quad \mathbb{E}[P(Y_{i-1})] - \mathbb{E}[P(Y_i)] \geq \varepsilon.$$

(by replacing P by $\neg P$ if necc, no abs value)

$Y_{i-1} = \boxed{X_1 \cdots X_{i-1}\ u_i\ u_{i+1} \cdots u_m}$

$Y_i = \boxed{X_1 \cdots X_{i-1}\ X_i\ u_{i+1} \cdots u_m}$

P is more likely to acc Y_{i-1} than Y_i.

Define a circuit $\tilde{P}$ which gets input $X_1, \ldots, X_{i-1}$:

Pick $z_i, \ldots, z_m$ at random.

$b = P(X_1, \ldots, X_{i-1}, z_i, \ldots, z_m)$

If $b = 1$, return $\bar{z}_i$. else return z_i.

What is the prob that $\tilde{P}$ is right?

$$\alpha = \Pr[P(x_1, \ldots, x_{i-1}, x_i, u_{i+1}, \ldots, u_m) = 1]$$

$$\alpha' = \Pr[P(x_1, \ldots, x_{i-1}, \bar{x}_i, u_{i+1}, \ldots, u_m) = 1]$$

$$\Pr[P(x_1, \ldots, x_{i-1}, u_i, u_{i+1}, \ldots, u_m) = 1] = \frac{1}{2}(\alpha + \alpha') \geq \alpha + \varepsilon$$

$$\Rightarrow \quad \alpha' \geq \alpha + 2\varepsilon.$$

$\Pr[\tilde{P}$ is correct$]$. if $z_i = x_i$ & $b = 0$
or $z_i = \bar{x}_i$ & $b = 1$

$$\| $$

$$\frac{1}{2} \cdot (1 - \alpha) + \frac{1}{2} \cdot \alpha' = \frac{1}{2} + \frac{1}{2}(\alpha' - \alpha) \geq \frac{1}{2} + \varepsilon.$$

$\square$.

How do we use this to build PRGs?

Toy case: stretch of 1.

$$G : \{0,1\}^d \longrightarrow \{0,1\}^{d+1}$$

[Blum-Micali] $G(x) = x\, \underline{b}$ $\qquad$ $b =$ Hard Function(x).

"hard to guess".

Later in the course:

Suppose we have access to "really hard" functions, then we can use that to build PRGs.

"If you can find hay in one haystack, you can find one in another"

[Impagliazzo-Wigderson] If $E = DTIME(2^{O(n)})$ has a language that requires circuits of size $2^{\Omega(n)}$, then $P = BPP$.