Pseudorandom Generators (prg)

- Expanders help in derandomizing a specific problem in RL.
- Prg are objects to derandomize general randomized algorithms.
- Defn: • A distribution R, over $\{0,1\}^m$, is (s, ε)-pseudorandom if $\forall$ circuits C of size $\leq s$,

small $\in (0, 1/2)$

$$\left| \Pr_{x \in R}[C(x)=1] - \Pr_{x \in U_m}[C(x)=1] \right| < \varepsilon.$$

$x \in U_m \leftarrow$ (uniform distribution)

$\{(R, U_m)$ are ε-close?

↳ This measures how well can C distinguish R from U_m.
pseudorandom $R \approx R$ indistinguishable from U_m (by boolean circuits)

- Let $S : \mathbb{N} \to \mathbb{N}$ be a function. A $2^{O(n)}$-time computable function $G : \{0,1\}^* \to \{0,1\}^*$ is an S-prg if $\forall \ell$,

(stretch) $G : \{0,1\}^\ell \to \{0,1\}^{S(\ell)}$ &

(random) $G(U_\ell)$ is $(S(\ell)^3, 0.1)$-pseudorandom.

OPEN Qn: We don't know whether S-prg ^{G} exists.

$\hookrightarrow$ explicitness of G is the issue.

$\triangleright$ S-prg saves random bits from $S(\ell)$ to ℓ.

Prg derandomizes classes

Lemma 1: An S-prg exists $\Longrightarrow$ $\forall$ function ℓ, $\mathrm{BPtime}(S\circ\ell(n)) \subseteq \mathrm{Dtime}(2^{\ell(n)} \cdot S\circ\ell(n))$.

Proof: Idea — Use S-prg G as the source of pseudo-random bits (in the randomized algo.) & take the majority vote. ["the algo. gets fooled by G"]

- Language $L \in \mathrm{BPtime}(S\circ\ell(n))$ if $\exists$ algorithm M that on input $x \in \{0,1\}^n$ uses $m := S\circ\ell(n)$ random bits r & runs for $O(S\circ\ell(n))$-time s.t.

$$\Pr_r[M(x,r) = L(x)] \geq 3/4.$$

- The derandomization idea is to use the S-prg G to produce r's:
  - On input x, our det. algo. <u>B</u> goes over all $z \in \{0,1\}^{\ell(n)}$, computes $M(x, G(z))$; outputs the majority-vote.
- We claim: $\Pr_z [M(x, G(z)) = L(x)] \geq \frac{3}{4} - 0.1 > 1/2$.

  $\Rightarrow$ B is correct.
- Suppose not, then $\Pr_z [\text{---}] < 3/4 - 0.1$.

$$\Rightarrow \left| \Pr_z [M(x, G(z)) = L(x)] - \Pr_r [M(x, r) = L(x)] \right| > \left| \frac{3}{4} - 0.1 - \frac{3}{4} \right| = \underline{0.1}.$$

- Consider the circuit $\underline{C_x}$ that on input $y \in \{0,1\}^{S\circ \ell(n)}$ outputs 1 iff $[M(x,y) = L(x)]$. "good" strings

Exercise: Since M is $O(S\circ \ell(n))$-time TM, we can simulate it by a boolean circuit C_x of size $O(S\circ \ell(n))^2$.

uniform distributions

$\Rightarrow$ $C_x(\cdot)$ distinguishes $G(U_{\ell(n)})$ from $U_{S\circ \ell(n)}$ well!

$\Rightarrow$ contradiction to the defn of S-prg G.

$\Rightarrow$ $\Pr_z [M(x, G(z)) = L(x)] > 1/2$

$\Rightarrow$ B solves L (correctly) in $O(2^{\ell(n)} \cdot S\circ \ell(n))$-time. $\square$

– By picking various stretch functions S, we get the following <u>conditional</u> derandomizations:

Corollary: (i) $\exists\ 2^{\varepsilon\ell}$-prg $\Longrightarrow$ BPP = P.
$\leftarrow$ exp. stretch

(ii) $\exists\ 2^{\ell^{\varepsilon}}$-prg $\Rightarrow$ BPP $\subseteq$ QuasiP $:=$ Dtime$\left(2^{\mathrm{polylog}(n)}\right)$.
"sub" exp. stretch

(iii) $\forall c>1$, $\exists\ \ell^{c}$-prg $\Longrightarrow$ BPP $\subseteq$ Subexp $:= \bigcap_{\varepsilon>0}$ Dtime$\left(2^{n^{\varepsilon}}\right)$
poly. stretch [or $\ell^{\omega(1)}$-prg]

Proof: Apply the Lemma on S & ℓ: [ℓ is "inverse" of S]

(i) $S: \mathbb{N}\to\mathbb{N};\ n\mapsto 2^{\varepsilon n}$ & $\ell:\mathbb{N}\to\mathbb{N};\ n\mapsto c\cdot\lg n$.

$\Rightarrow\ 2^{\ell(n)} = n^{c}$ & $S\circ\ell(n) = S(c\lg n) = 2^{\varepsilon c\lg n} = n^{\varepsilon c}$.

(ii) $S: n \mapsto 2^{n^\varepsilon}$ & $\ell: n \mapsto c \cdot (\lg n)^{1/\varepsilon}$

(iii) $S: n \mapsto n^c$ & $\ell: n \mapsto n^\varepsilon$. $\square$

OPEN Qns: $BPP \subseteq Subexp$? ---- $BPP = P$?

- How do we construct these prg's ?
The only known way is to exploit the hardness of problems!

(Circuit)(Explicit)

- Hardness, prg's, & derandomizations are all open & highly related ....

Hardness & Prg's

- We define two types of hardness of boolean functions.

Defn: • For $f:\{0,1\}^* \to \{0,1\}$, the average-case hardness $H_{avg}(f)$ is the largest $S(n)$ s.t. $\forall$ circuit $C_n \in \text{size}(S(n))$, $\Pr_{x\in U_n}[C_n(x) = f(x)] < \frac{1}{2} + \underbrace{1/S(n)}_{\text{small advantage}}$.

• Worst-case hardness $H_{wrs}(f)$ is the largest $S(n)$ s.t. $\forall$ circuit $C_n \in \text{size}(S(n))$, $\Pr_{x\in U_n}[C_n(x) = f(x)] < 1$.

▷ $H_{avg}(f) \leq H_{wrs}(f) < 2^{2n}$.
by truth-table of f.

- # functions on $\{0,1\}^n$ is $\approx 2^{2^n}$, while
  # circuits of size-s is $\approx s^{2s} \ll 2^{2^n}$ (if $s < 2^{n/2}$)
  $\Rightarrow$ for random fn. f, $H_{wrs}(f) \geq 2^{n/2}$.

- But, we don't know of "natural" or "explicit" f with super-polynomial hardness!

- The conjectured f, of cryptographic significance, are:
  (1) $H_{wrs}(3SAT) \geq 2^{\Omega(n)}$?

(2) H_{avg} (Integer-Factoring) $\geq n^{\omega(1)}$?

Consider some decision version n-bit input

- We'll later prove: worst-case hardness gives also an average-case hard function.

The tool to do this is local list-decoding of linear error-correcting codes.

Hardness vs Randomness: For now, we relate average-case hardness, to prg, to derandomization.

Theorem (Nisan, Wigderson, 1988): If $\exists f \in E$ with $H_{avg}(f) \geq S(n)$, then $\exists S'(\ell)$-prg, where $\underline{S'(\ell)} := S(n)^{0.01}$, for $\frac{100n^2}{\lg S(n)} < \ell \leq \frac{100(n+1)^2}{\lg S(n+1)}$.

i.e. good for large $S(n)$

Proof: Idea - $\because$ f is hard, its values "look random" to small circuits! So, stretch a seed $z \in \{0,1\}^\ell$ to $\{0,1\}^{S'(\ell)}$ by choosing n-sized subsets $I_1, \ldots, I_m \subseteq [\ell]$ & consider $f(z_{I_1}) \circ f(z_{I_2}) \circ \cdots \circ f(z_{I_m})$.

- hard to guess the next bit, by small circuits?
- take $I_1, \ldots, I_m$ almost disjoint.

Defn: Let $\mathcal{I} := \{I_1, \ldots, I_m\}$ be a family of n-size subsets of $[\ell]$. Let $f: \{0,1\}^n \to \{0,1\}$. [$m \gg \ell$]

The $(\mathcal{I}, f)$-NW generator is the function $NW_{\mathcal{I}}^f : \{0,1\}^\ell \to \{0,1\}^m$; $z \mapsto f(z_{I_1}) \circ \cdots \circ f(z_{I_m})$, where z_I is the restriction of z to the coords. I.

Defn: Let $\ell > n > d$. A family $\mathcal{I} = \{I_1, \ldots, I_m\}$ of n-size subsets of $[\ell]$ is an (ℓ, n, d)-design if $|I_j \cap I_k| \leq d$, for all $j \neq k \in [m]$.

- Later we show that for hard f & $\mathcal{I}$ being a design, the $(\mathcal{I}, f)$-NW-generator is pseudorandom!

Lemma 1 (designs): $\exists$ algorithm A that on input (ℓ, n, d), where $\ell > 10n^2/d$, outputs an (ℓ, n, d)-design $\mathcal{I}$, having $m \geq 2^{d/10}$ subsets, in time $2^{O(\ell)}$.

Proof: Idea – Greedily build $\mathcal{I}$.

0) Initialize $\mathcal{I} \leftarrow \phi$.

1) Say, $\mathcal{I} =: \{I_1, \ldots, I_m\}$ with $m < 2^{d/10}$.
Find $\underline{I} \in \binom{[\ell]}{n}$ s.t. $\underbrace{\forall j \in [m], |I \cap I_j| \leq d}$.

2) $\mathcal{I} \leftarrow \mathcal{I} \sqcup \{I\}$ & goto (1).

Time taken: $\leq (2^{\ell} \cdot n) \times 2^{d/10} \times \underline{2^{d/10}} = 2^{O(\ell)}$.

Qn: Can it get stuck at $m < 2^{d/10}$?

- We show the existence of I, in Step (1), by the **probabilistic method**.

- Build I by picking each element in $[\ell]$ with probability $= 2n/\ell$.

$\Rightarrow$ ▷ $E[\#I] = \sum_{x\in[\ell]} 1\cdot \Pr[\text{pick } x] = \sum_{x} \frac{2n}{\ell} = 2n.$

▷ $\forall j \in [m],\ E[|I \cap I_j|] = \sum_{x\in I_j} 1\cdot \Pr[\text{pick } x] = n \times \frac{2n}{\ell}$
$= (2n^2/\ell) < d/5.$

- Recall **Chernoff's Bound**:

mean/expectation

$$\Pr[\underbrace{|X-\mu|}_{\text{deviation}} \geq c\cdot\mu] \leq 2\cdot \underbrace{e^{-\mu\cdot\min\left(\frac{c}{2},\frac{c^2}{4}\right)}}_{\text{exp. small!}}.$$

- By Chernoff's bound: $\Pr_{I}[|I|<n] \leq \Pr_{I}\left[\left||I|-2n\right| > \frac{1}{2}\cdot 2n\right]$
$< 2\cdot e^{-2n\cdot 1/16} = 2e^{-n/8}$.
- Similarly, $\forall j$, $\Pr_{I}[|I\cap I_j| > d] \leq \Pr_{I}\left[\left||I\cap I_j|-\frac{d}{5}\right| > 4\cdot\frac{d}{5}\right]$
$< 2\cdot e^{-\frac{d}{5}\cdot\frac{4}{2}} = 2\cdot e^{-2d/5}$.

$\Rightarrow \Pr_{I}[|I|<n \text{ OR } \exists j, |I\cap I_j| > d] <$
$2e^{-n/8} + m\times 2e^{-2d/5} < 2e^{-n/8} + 2e^{-d/2}$
< 1.

$\Rightarrow \Pr_{I}[|I|\geq n \text{ AND } \forall j, |I\cap I_j| \leq d] > 0$.

$\Rightarrow$ In Step (1), I exists $\Rightarrow$ Algo A outputs (ℓ,n,d)-design.

$\square$

- We use the design in $(\mathcal{I}, f)$-NW generator now.

<u>Lemma 2</u> (NW-generator): If $\mathcal{I}$ is an (ℓ, n, d)-design with $|\mathcal{I}| = 2^{d/10} =: m$; $f : \{0,1\}^n \to \{0,1\}$ with $H_{avg}(f) > 2^{2d}$, then $NW_{\mathcal{I}}^{f}(U_\ell)$ is $(H_{avg}(f)/10, 0.1)$-pseudorandom.

<u>Proof</u>: <u>Idea</u> - Suppose circuit C "distinguishes" $NW(U_\ell)$ from U_m. Then, we'll design a <u>bit-predictor</u> circuit C' for $f(z_{I_i})$, for some $i \in [m]$.

C' will <u>contradict</u> the avg-case hardness of f !

- Let $S := H_{avg}(f)$.
- Suppose $\exists$ circuit C of size $\leq S/10$ s.t.

$$\left| \Pr[C(NW_J^f(U_\ell)) = 1] - \Pr[C(U_m) = 1] \right| \geq 0.1.$$

(i.e $NW(U_\ell)$ is not pseudornd)

- Wlog, assume $\Pr[C(NW(U_\ell)) = 1] - \Pr[C(U_m) = 1] \geq 0.1$.
- We'll now devise a bit-predictor for NW_J^f.
- Identify the bit which can be predicted:
- For that, define distributions $\mathcal{D}_0, \ldots, \mathcal{D}_m$ over $\{0,1\}^m$ s.t.

$\forall i$, $\mathcal{D}_i$:
- choose $x \in_R U_\ell$; $z_{i+1}, \ldots, z_m \in_R \{0,1\}$
- Compute $y := NW_J^f(x)$.
- Output $\langle \underbrace{y_1, \ldots y_i}_{} ; z_{i+1}, \ldots, z_m \rangle$.

Hybrid distribution

▷ $\mathcal{D}_0 \equiv U_m$; $\mathcal{D}_m \equiv NW_f^f(U_\ell)$.

- Define $p_i := \Pr[C(\mathcal{D}_i)=1]$. We've $p_m - p_0 \geq 0.1$.

  $\Longrightarrow \exists i_0 \in [m]$, $p_{i_0} - p_{i_0-1} \geq 0.1/m$ (by averaging)

- We intend to use this "advantage" to predict the i_0-th bit of $NW_f^f(U_\ell)$, given the preceding ones.

- Define circuit C': On input $y_1, \ldots, y_{i_0-1}$:
  - Pick $z_{i_0}, \ldots, z_{i_m} \in_R \{0,1\}$.
  - Output $\begin{cases} z_{i_0}, & \text{if } C(\underbrace{y_1, \ldots, y_{i_0-1}, z_{i_0}}, \ldots, z_m) = 1 \\ 1 - z_{i_0}, & \text{else} \end{cases}$

Qn: How well does C' predict y_{i_0}?

$$\Rightarrow \Pr_{\substack{y \in NW(\mathcal{U}_\ell) \\ \bar{z} \in \mathcal{U}_m}} [C'(y_1, \ldots, y_{i_0-1}) = y_{i_0}] =$$

$$\Pr[z_{i_0} = y_{i_0}] \cdot \Pr[C(y_1, \ldots, y_{i_0-1}, z_{i_0}, \ldots, z_{i_m}) = 1 \mid z_{i_0} = y_{i_0}]$$

$$+ \Pr[z_{i_0} \neq y_{i_0}] \cdot \Pr[\quad \text{''} \quad \neq 1 \mid z_{i_0} \neq y_{i_0}]$$

$$= \tfrac{1}{2} \cdot \Pr[C(D_{i_0}) = 1] + \tfrac{1}{2} \cdot (1 - \Pr[C(y_1, \ldots, y_{i_0-1}, \bar{y}_{i_0}, z_{i_0+1}, \ldots, z_{i_m}) = 1])$$

$$= p_{i_0} + \tfrac{1}{2} - \tfrac{1}{2}\left(p_{i_0} + \Pr[C(y_1, \ldots, y_{i_0-1}, \bar{y}_{i_0}, z_{i_0+1}, \ldots, z_{i_m}) = 1]\right)$$

$$\Pr[C(y_1, \ldots, y_{i_0-1}, y_{i_0}, z_{i_0+1}, \ldots, z_{i_m}) = 1]$$

$$= p_{i_0} + \tfrac{1}{2} - \tfrac{1}{2}(2 \times p_{i_0-1}) = \tfrac{1}{2} + (p_{i_0} - p_{i_0-1}) \geq \tfrac{1}{2} + \frac{0.1}{m}.$$

$\Rightarrow$ C' is a <u>decent</u> bit-predictor!

- To make C' deterministic, fix $z_{i_0}, \ldots, z_{i_m}$ suitably; get circuit

C'': $\Pr_{y \in NW(\mathcal{U}_e)} [C''(y_1, \ldots, y_{i_0-1}) = y_{i_0}] \geq \frac{1}{2} + \frac{0.1}{m}$

(Averaging argument)

▷ $\text{size}(C'') < 2 \cdot \text{size}(C) \leq S/5$.

· Plugging the defn. of NW_f^f, we get:

▷ $\Pr_{Z \in \mathcal{U}_e} [C''(f(Z_{I_1}), \ldots, f(Z_{I_{i_0-1}})) = f(Z_{I_{i_0}})] \geq \frac{1}{2} + \frac{0.1}{m}$.

Idea: Fix $Z_{[e] \setminus I_0}$ s.t. the above prob. is retained.

$\Rightarrow$ $\forall j \in [i_0 - 1]$, Z_{I_j} has all places fixed except $|I_j \cap I_{i_0}|$ many variables. $(\leq d)$

$\Rightarrow$ $f(Z_{I_j})$ is d-variate.

$\Rightarrow f(Z_{I_1}), \ldots, f(Z_{I_{i_0-1}})$ can be computed (trivially) by circuits of size $O(d \cdot 2^d)$.

$\Rightarrow \exists$ circuit B of size $< \frac{S}{5} + O(d \cdot 2^d) \cdot m$
$= \frac{S}{5} + O(d \cdot 2^d \cdot 2^{d/10}) < S$ $(\because S > 2^{2d})$ s.t.

$$\Pr_{Z_{I_{i_0}} \in U_n} [B(Z_{I_{i_0}}) = f(Z_{I_{i_0}})] > \frac{1}{2} + \frac{0.1}{m} > \frac{1}{2} + \frac{1}{S}.$$

So, B contradicts the assumption: $H_{avg}(f) = S$.

$\Rightarrow$ $NW_J^f(U_\ell)$ is $(S/10, 0.1)$-pseudorandom.

$\square$

Proof (of NW-Theorem): • Let $f \in E = \mathrm{Dtime}(2^{O(n)})$ & $H_{avg}(f) \geq S(n)$.

• Define $S'(\ell)$-prg G : On input $z \in \{0,1\}^\ell$:

1) Pick n s.t. $100n^2/\lg S(n) < \ell \leq \frac{100(n+1)^2}{\lg S(n+1)} \leq \frac{200n^2}{\lg S(n)}$
2) Set $d := \lg S(n)/10$. (Lemma 1)
3) Compute an (ℓ, n, d)-design $\mathcal{I} =: \{I_1, \ldots, I_m\}$; $m := 2^{d/10}$.
4) Output $NW_{\mathcal{I}}^f(z)$.

• This takes time : $2^{O(\ell)} + \underbrace{2^{O(n)}}_{\text{computing } f} \cdot m \leq 2^{O(\ell)}$.

• Since $H_{avg}(f) \geq S(n) = 2^{10d}$, by Lemma-2 we get:

$NW_{\mathcal{I}}^f(U_\ell)$ is $(S(n)/10, 0.1)$-pseudorandom.

▷ The stretch is $m = 2^{d/10} = S(n)^{0.01} =: S'(\ell)$.

$\Rightarrow$ G is an $s'(\ell)$-prg $(\because\ s'(\ell)^3 < S(n)/10)$. $\square$

[stretch $s'(\ell) > \ell$ requires $S(n)^{0.01} \geq n^2$

$\Leftrightarrow$ $S(n) \geq n^{200}$.]

[Thus, superpoly-hardness of f gives a good stretch!]

- Simply, "hardness $\Rightarrow$ prg" !

Qn: Is there a converse ?

Claim: $\exists\ S(\ell)$-prg $\Rightarrow$ $\exists f \in E$: $H_{wrs}(f_n) > n^3$.

Pf: • Let $G : \{0,1\}^\ell \to \{0,1\}^n$ be an $S(\ell)$-prg.

• Define $f = f_n$ on $\{0,1\}^n$: $f_n(x) = 1$ iff $x \in Im(G)$.

$\Rightarrow \quad f \in E.$

- Let C_n be the smallest circuit computing f_n.

▷ $\Pr[C_n(G(U_\ell)) = 1] = 1.$

▷ $\Pr[C_n(U_n) = 1] \leq 2^\ell/2^n \leq 1/2$

$\Rightarrow$ C_n distinguishes $G(U_\ell)$ from U_n.

$\Rightarrow \quad size(C_n) > S(\ell)^3 = n^3.$ $\square$

- We will now see more impressive applications of prg in complexity results: