Today: Hardness vs. Randomness

An interesting combinatorial lemma:

def $\mathcal{I} = \{I_1 \dots I_m\} \subseteq [\ell]$ is an

(m, ℓ, n, d) -design $(\ell > n > d)$

if 1) $|I_j| = n \quad \forall j$

2) $|I_j \cap I_k| \leq d \quad \forall j \neq k$

Thm $\exists$ algorithm running in $2^{O(\ell)}$ time s.t.

for $n > d$, $\ell > \frac{10n^2}{d}$

which outputs (m, ℓ, n, d)-design s.t. $m = 2^{d/10}$

Pf. idea: greedy algorithm

show can always make progress

after have picked $I_1, \dots I_\ell$ for $\ell < 2^{d/10}$

search all subsets to find I^*

s.t. $|I^* \cap I_j| \leq d \quad \forall j \in [\ell]$

runtime: $\text{poly}(m) \cdot 2^\ell$

why doesn't it get stuck? probabilistic proof

if pick I^* randomly,

s.t. prob $x \in [\ell]$ gets chosen with prob $\frac{2n}{\ell}$

(if I^* too big, truncate to size n later on)

$$E[|I^*|] = \ell \cdot \frac{2n}{\ell} = 2n$$

$$\Rightarrow \Pr[|I^*| \geq n] \geq 0.9 \quad \Big\} \text{Chernoff}$$

$$E[|I^* \cap I_j|] = \underset{\substack{\uparrow \\ \text{size } n}}{n} \cdot \frac{2n}{\ell} = \frac{2n^2}{\ell} < \frac{d}{5} \quad \text{b/c } \ell > \frac{10n^2}{d}$$

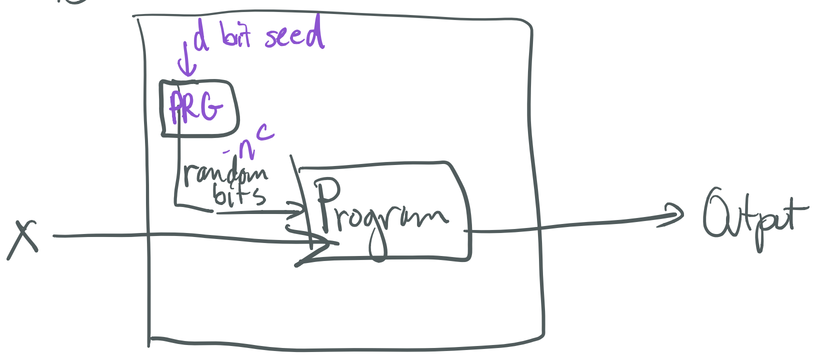
$$\Rightarrow \Pr[|I^* \cap I_j| \geq d] \leq \frac{1}{2} \cdot 2^{-d/10} \quad \Big\} \text{Chernoff}$$

$$\Pr[\forall j \ |I^* \cap I_j| \leq d \ \& \ |I^*| \geq n]$$

$$\geq 1 - (0.1 + \underset{\substack{\uparrow \\ < 2^{d/10}}}{m} \cdot \frac{1}{2} \cdot 2^{-d/10}) \geq 0.4$$

$$\Rightarrow \Pr[I^* \text{ good}] \geq 0.4 \quad \blacksquare$$

Derandomization



previously: derandomized programs using k-wise indep random bits

today: general programs

PRG outputs bits that look random to any time t algorithm

More generally: hard on average fctns
(for programs running in time $\le t(n)$
get advantage $\le 1/t(n)$)

$\Rightarrow$ PR on programs run in time $\le t(n)^c$
adv $\le \frac{1}{10}$

or some other parameters?

Some definitions!

def "Pseudorandom"

Let X_n be a sequence of r.v.'s on $\{0,1\}^n$

X_n is (t, ε)-p.r. if $\forall$ probabilistic TM's running in time $\leq t(n)$

$$\left| \Pr[T(X_n)=1] - \Pr[T(U_n)=1] \right| \leq \varepsilon(n)$$

def $f : \{0,1\}^\ell \to \{0,1\}$ is (t, ε)-average case hard

if $\forall$ A in time $t(\ell)$

$$\Pr_{x, \$ \text{ of } A}[A(x) = f(x)] \leq \frac{1}{2} + \varepsilon(\ell)$$ for large enough ℓ

$$\leq \frac{1}{2} + \frac{1}{t(\ell)}$$

pick $\varepsilon(\ell) < \frac{1}{t(\ell)}$

f is t-ave case hard if $\text{adv}(A)$ is $\frac{1}{t(\ell)}$ for $\underbrace{\text{nonuniform}}_{\text{ckt complexity of } A}$ A in time $t(\ell)$

Warmup:

Thm if $f: \{0,1\}^l \to \{0,1\}$ is (t, ε) - ave case hard
then $G(y) = y \circ f(y)$ is PRG

l bits $\to$ $l+1$ bits
extends by 1 bit

Unpredictability:

def $X = X_1 \ldots X_n$ is "next bit unpredictable" (nbu)
with parameters $t(n) + \varepsilon(n)$ if
$\forall P$ using $\leq t(n)$ time
$$\Pr_{X,\, i \in_R [n],\, \text{coins of } P} [P(X_1 \ldots X_{i-1}) = X_i] \leq \frac{1}{2} + \varepsilon(n)$$

Cool theorem:
nbu + pr are equivalent (up to parameters)

more specifically:

P runs in t steps

T runs in $t + O(n)$ steps

1) if next bit i can be predicted
$$\Pr_{X, i, \text{coins of } P} [P(X_1 \ldots X_{i-1}) = X_i] \geq \frac{1}{2} + \frac{1}{n^K}$$
then $\exists$ statistical test T which distinguishes X from U
with adv $1/n^K$

2) if $\exists$ distinguishing test for X from U in t steps
with advantage $\frac{1}{n^k}$
then can predict with advantage $\frac{1}{n^{k+1}}$ in $t+O(n)$ steps

Proof "idea" for $G(y) = y \circ f(y)$ is a PRG:

$y \in_R U \Rightarrow y$ is n.b.u.

since $f(y)$ is hard for any
P using $\leq t(n)$ steps
$\Rightarrow$ any program on $t(n)$ steps
predicts $f(y)$ with
adv $\leq \frac{1}{t(n)}$
$\Rightarrow y \circ f(y)$ is n.b.u
with parameter $\frac{1}{t(n)}$
$\Rightarrow y \circ f(y)$ is p.r.
$\blacksquare$

How do you get >1 bit stretch?

Nisan Wigderson Generator: (Given f)

$|I_j| = n$

$|I_j \cap I_k| \le d$

$\ell > n > d$

Given (ℓ, n, d) design $\mathcal{I} = \{I_1, \dots, I_m\} \subseteq [\ell]$

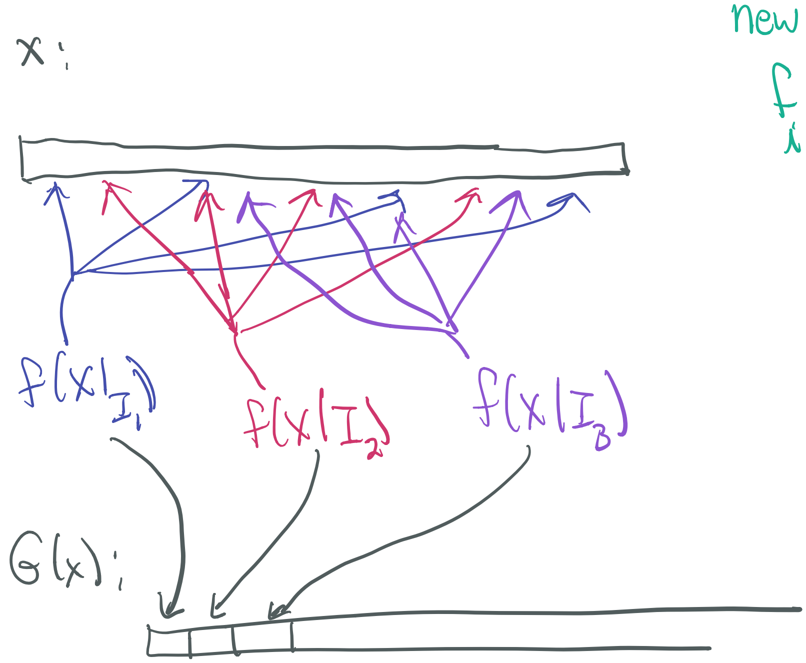
$$G : \{0,1\}^\ell \to \{0,1\}^m$$

is
$$G(x) = f(\underbrace{x|_{I_1}}_{\text{string of length } n \text{ selecting bits indexed by } I_1}) \circ f(x|_{I_2}) \circ \dots \circ f(x|_{I_m})$$

$$= f_1(x) \circ f_2(x) \circ \dots \circ f_m(x)$$

new notation:

$$f_i(x) = f(x|_{I_i})$$



Thm If (1) $\exists f : \{0,1\}^n \to \{0,1\}$ s.t.

$$f \in E = DTIME(2^n)$$

s.t. f is t-ave case hard

(2) $\exists$ (ℓ, n, d) design with m sets constructable in time $2^{O(\ell)} \leq t(\ell)/2$

s.t. $m = 2^{d/10}$ $= t(\ell)^{1/c}$ e.g. $c=2$, $\ell > \frac{10n^2}{d}$, $n > d$

then G is $\frac{\varepsilon}{m}$-PRG against nonuniform time m

think of $\varepsilon = \frac{1}{10}$

Pf.

If G not $\frac{1}{m}$-PRG against time m

$\exists$ n.b. predictor P s.t.

$$\Pr_{i,x}[P(f_1(x), f_2(x), \ldots f_{i-1}(x)) = f_i(x)] \geq \frac{1}{2} + \frac{\varepsilon}{m}$$

$$\underbrace{\& \text{ time}(P)}_{\text{will use to compute } f} = \text{time}(\underset{\uparrow}{T}) + O(m)$$

time prg predictor

will use to compute f

with $\frac{\varepsilon}{m}$ advantage in $O(t(\ell))$ time

where $m \approx t^{1/c}$

$t \approx m^c$

As usual:

averaging $\Rightarrow$ $\exists$ i^* s.t. achieve expectation

& averaging $\Rightarrow$ $\exists$ $\underbrace{\text{choice of bits of } X}_{\text{call it } Z}$

$\underbrace{\text{not in } I_{i^*}}_{\overline{I}_{i^*}}$ achieving expectation

notation: $Y \leftarrow X$ with bits in $\overline{I}_{i^*}$ set to Z

and other bits I_{i^*} set randomly

$$\Pr_Y[P(f_1(y)\, f_2(Y) \ldots f_{i^*-1}(Y)) = f_{i^*}(Y)]$$

$$\geq \frac{1}{2} + \frac{\varepsilon}{m}$$

- each depend on $\leq d$ bits of Y

  since $|I_{i^*} \cap I_j| \leq d \quad \forall j$

- since $f \in E$ can compute each f_j

  in time $\leq 2^d$

$$A(y) = P(f_1(y)\, f_2(Y) \ldots f_{i^*-1}(Y))$$

predicts $f_{i^*}(Y)$ with adv $\geq \frac{\varepsilon}{m} = \frac{1}{10\, 2^{d/10}}$

runtime $\underbrace{\tilde{O}(2^d)}_{\text{compute fctn on } d \text{ bits}} \times \underbrace{O(m)}_{\#\text{ times}} + \frac{t(l)}{2}$

↑ time to construct design

$2^{d/10}$

Set $d \approx \frac{\log t}{10}$

$\Rightarrow 2^d \approx \tilde{O}(t^{1/10})$

total time: $2^d \cdot O(m) + O(m) = \tilde{O}(t^{1/10}) \cdot O(t^{1/c}) + \frac{t}{2}$

$< t$

Contradicts hardness of f ∎