Lecture 7

- p.i. random bits to reduce error
- Random bits for interactive proofs
  - IP
  - Graph
Using Pairwise Independence to reduce error

Setting:

Given RP algorithm \( \hat{a} \) on input \( x \), random bits \( R \), outputs \( \text{Accept} \geq \frac{1}{2} \)

\( \text{if } x \in L \quad \Pr_{R} [ \hat{a} \text{ on input } x, \text{random bits } R, \text{outputs } \text{Accept} \geq \frac{1}{2} ] = 0 \)

How can we reduce error?

1) Repeat at \( k \) times
   - use new random bits each time
   - if ever see "Accept" then output "Accept"
   - else output "Reject"

   \[ \text{behavior:} \]
   \[ \text{if } x \in L \quad \Pr [\text{"Accept"}] = 1 - (1 - \frac{1}{2})^k = 1 - \frac{L}{2^k} \]
   \[ \text{if } x \notin L \quad \Pr [\text{"Accept"}] = 0 \]

   \[ \therefore \text{error probability} \leq 2^{-k} \text{ (1-sided error)} \]
2) "2-point sampling"

\[
\text{idea: use p.i. samples instead}
\]

\[
\text{assumption: given } \mathcal{H}, \text{ family of p.i. \( f \), st. can pick random } h \in \mathcal{H} \text{ with } O(K + |R|) \text{ random bits + poly } (K, |R|) \text{ time}
\]

\[
\text{Sampling algorithm}
\]

\[
\begin{align*}
\text{pick } & h \in \mathcal{H} \\
\text{for } & i = 1 \ldots 2^k + 2 \\
& r_i \leftarrow h(i) \\
& \text{if } A(x, r_i) = "\text{Accept}" \text{ output } "\text{Accept}" \text{ and halt}
\end{align*}
\]

\[
\text{output } "\text{REJECT}"
\]

\[
\text{random bits used: } O(K + |R|) \\
\text{runtime: } O(2^k \times \text{time for } A) \quad \Theta \text{ (bit doesn't depend on } n) \]

\[
\text{behavior:}
\]

\[
\begin{align*}
\text{if } x \not\in L & , P_h[\text{Accept}] = 0 \\
\text{if } x \in L & \\
\text{will misclassify if never see } \Gamma & \text{ st. } A(x, r_\Gamma) = "\text{Accept}" \\
\text{let } b & (r_\Gamma) = \begin{cases} 
0 & \text{if } A(x, r_\Gamma) = "\text{Reject}" \\
1 & \text{o.w.}
\end{cases}
\end{align*}
\]

\[
\text{let } Y = \sum_{i=1}^{2^{K+2}} b(r_i)
\]

\[
E[\frac{Y}{2}] \geq \frac{2^{K+2} - \frac{1}{2}}{2^{K+2}} = \frac{1}{2}
\]

\[
E[6(r)] = P_h[\text{Accept}]
\]

\[
\text{if } x \not\in L \text{ expect to see } \frac{1}{2} "\text{accept}" \quad \text{what is probability you don't see any } ?
\]

\[
\text{terms in sum}
\]
Two useful lemmas:

**Chebyshev's Inequality:**

- For a random variable $X$, with $E[X] = \mu$ and $\text{Var}[X] = \sigma^2$,
- \[ \Pr[|X - \mu| \geq \epsilon] \leq \frac{\text{Var}[X]}{\epsilon^2} \]

**Pairwise Independence Tail Inequality:**

- Let $X_1, \ldots, X_t$ be random variables in $\mathbb{R}$.
- Let $X = \frac{\sum_{i=1}^{t} X_i}{t}$,
- $\mu = E[X]$.
- Then,\[ \Pr[|X - \mu| \geq \epsilon] \leq \frac{1}{t} \epsilon^2 \]

---

What is $\Pr\left[ \frac{Y}{q} = 0 \right]$? i.e., $\Pr["\text{REJECT}""]$

- $\Pr["\text{REJECT}"{}] = \Pr\left[ \frac{Y}{q} = 0 \right]$  
  \[ \leq \Pr\left[ \left| \frac{Y}{q} - E\left[ \frac{Y}{q} \right] \right| \geq E\left[ \frac{Y}{q} \right] \right] \]
  \[ \leq \frac{1}{q^2 \cdot \left( \frac{1}{a} \right)^2} \]
  \[ = 2^{-(k+2)} \cdot 4 = 2^{-k} \]

Note: runtime is $O(2^k \cdot T_d(n))$

It bad, but doesn't depend on $n$
Interactive Proofs

$NP = \text{all decision problems for which "Yes" answers can be verified in pt ime by a deterministic TM ("verifier")}$

$IP$: generalization of $NP$:

- short proofs $\Rightarrow$ short interactive proofs

"Conversations that convince"

Model

```
def "Interactive Proof Systems" (IPS) [Goldwasser, Micali, Rackoff]
  for language $L$ is protocol s.t.
  \* if $V, P$ follow protocol & $x \in L$ then $Pr_{V, S} [V \text{ accepts } x] \geq \frac{2}{3}$
  \* if $V$ follows protocol & $x \notin L$ then (no matter what $P$ does)
    \* $Pr_{V, S} [V \text{ rejects } x] \geq \frac{2}{3}$
    \* what if require that $P$ follow protocol? For cryptography, useless!
```
\[
\text{def } \text{IP} = \exists L \mid L \text{ has IP-SZ}
\]

**Note** Clearly \( \text{NP} \subseteq \text{IP} \)

turns out \( \text{IP} = \text{PSFACE} \)

---

**Today [Goldwasser Sipser]**

Protocol in which \( P \) can convince \( V \) that

- size of set \( S \) is "big"
- only need that \( \forall x \in S \), can verify that \( x \) is in \( S \) in poly time

Let \( S_\emptyset = \{ x \mid x \text{ satisfies } \emptyset \} \)

**Note** given \( x \), \( V \) can check that \( x \) satisfies \( \emptyset \)

Claim exist protocol \( \sigma \) on input \( \emptyset \)

- if \( |S_\emptyset| > K \)
  - if \( V |P \) follow protocol then \( \Pr[\text{V accepts}] = 2/3 \)

- if \( |S_\emptyset| \leq K \)
  - if \( V \) follows protocol then \( \Pr[\text{V accepts}] < 1/3 \)

- no requirement on \( P \)
- \( V \) will not accept even if \( P \) cheats!
- important for crypto

\[ \Delta^I \text{uhomme} \Delta = 4 \]
Why interesting?

- Can use to show \( \# \) random strings which cause algorithm \( A \) to accept on input \( x \geq 2/3 \).
- Used to show "public coin" model \( \approx \) "private coin" model (i.e., can prove same set of statements).

**First idea**: Random sampling

Repeat \( ? \) times:

1. \( V \) picks random assignment \( x \).
2. \( V \) evaluates \( \Phi(x) \).

Outputs \( \frac{\# \text{ satisfied } x's}{\text{total } \# \text{ repetitions}} \).

Needs \( \Omega\left(\frac{\# \text{ total assignments}}{\# \text{ satisfying assignments}}\right) \) (could be \( \Omega(2^n) \)).

**Problem**: What if \( \Sigma \Phi \) is very small compared to set of all assignments?
Fix: Universal Hashing

\[ \text{All assignments} \]

\[ \text{size } 2^n \]

\[ h \]

\[ \text{size } 2^\ell \]

\[ \text{pick } \ell \text{ s.t. } k > \frac{2^\ell}{2} \]

\[ \text{need:} \]
1. \( |h(S_p)| \approx |S_p| \)
2. \( \frac{|h(S_p)|}{2^\ell} \approx \frac{1}{\text{poly}(n)} \)
   (in our case, constant)
3. \( h \) computable in poly time

Protocol: \xi \text{For distinguishing set size } k \text{ from set size } \xi \text{? \xi} \]

Given \( H \), collection of p.i. time mapping \( \mathbb{E}_0, \mathbb{I}_0 \rightarrow \mathbb{E}_0, \mathbb{I}_0 \)

1. \( V \) picks \( h \in H \)
2. \( V \rightarrow P; \quad h \)
3. \( P \rightarrow V; \quad x \in S_p \text{ s.t. } h(x) = 0^\ell \)
4. \( V \) accepts iff \( x \in S_p \)

Idea: hope: \( h(S_p) \) fills a "random" portion of range

Case 1: \( |S_p| > k \):
   - hopefully \( |h(S_p)| \approx k \) so all powerful \( P \) can find preimage in \( S_p \)

Case 2: \( |S_p| \leq k \):
   - \( |h(S_p)| \leq k \) so less likely \( 0^\ell \) hit
   - \( P \) can't send fake preimage because \( V \) will detect
Recall \( H \) is p.i. family of hash functions if
\[
\forall x, y \in \Sigma_0^1 \times \Sigma_0^1 \quad \forall a, b \in \Sigma_0^1 \times \Sigma_0^1 \quad \Pr_{h \in H} \left[ h(x) = a \quad \&\quad h(y) = b \right] = 2^{-2l}
\]

Lemma \( H \) is p.i., \( U \subseteq \Sigma_0^1 \times \Sigma_0^1 \), \( a = \frac{|U|}{2^l} \)

then \( a - \frac{a^2}{2} \leq \Pr_h \left[ 0^l \in h(U) \right] \leq a \)

Proof:

RHS:
\[
\forall x \quad \Pr_h \left[ 0^l = h(x) \right] = 2^{-l} \quad \text{since } H \text{ is p.i.}
\]

so \( \Pr_h \left[ 0^l \in h(U) \right] \leq \sum_{x \in U} \Pr_h \left[ 0^l = h(x) \right] = \frac{|U|}{2^l} = a \)

LHS:
\[
\Pr \left[ \bigcup_i A_i \right] \geq \sum_i \Pr[A_i] - \sum_{i \neq j} \Pr[A_i \cap A_j]
\]

\[
\Pr_h \left[ 0^l \in h(U) \right] \geq \sum_{x \in U} \Pr_h \left[ 0^l \in h(x) \right] - \sum_{x \neq y \in U} \Pr_h \left[ 0^l = h(x) = h(y) \right]
\]

\[
= \frac{|U|}{2^l} - \left( \frac{|U|}{2^l} \right) \frac{1}{2^l} \geq \frac{|U|}{2^l} - \frac{|U|^2}{2^l} \cdot \frac{1}{2^l}
\]

\[
\geq a - \frac{a^2}{2}
\]
Finishing up:

Pick \( k \) s.t. \( 2^{k-1} \leq k \leq 2^k \). Let \( a = \frac{|h(S_p)|}{2^k} \).

If \( |S_p| > k \) then \( a \leq \frac{1}{2} \).

So \( \Pr[0 \in h(S_p)] \geq a - \frac{a^2}{2} \geq \frac{3}{8} \).

If \( |S_p| \leq \frac{k}{\Delta} \) then \( a \leq \frac{k}{\Delta} \leq \frac{1}{\Delta} \).

So \( \Pr[0 \in h(S_p)] \leq \frac{1}{\Delta} \).

(Picking \( \Delta = 4 \) \( \Rightarrow \)) \( \leq \frac{1}{4} \).

If repeat \( O(\log \frac{1}{\beta}) \) times,

Chernoff \( \Rightarrow \) with prob \( \geq 1 - \beta \).

If \( |S_p| > k \) then \( P \) is successful \( \geq \frac{3}{8} - o(1) \) of the repetitions.

If \( |S_p| \leq \frac{k}{\Delta} \) then \( P \) is successful \( \leq \frac{1}{4} + o(1) \) of the repetitions.

Can improve so that \( \Delta = 1 - \epsilon \).

How???
Idea for general Thm:

\[ \text{l.e. } \text{IP}_{\text{private coins}} = \text{IP}_{\text{public coins}} \]

argue that l.b. protocol can be used to show that size of accepting region probability mass is large.

(need that am verify a conversation / random coin to be in accept region)