Today
- Goldreich-Levin Theorem
- Connection to Coding

Lecture 30
Computational Complexity
(21 May 2020)
Instructor: Prahladh Harsha

Motivation:

Suppose we hard fn (but not necessarily a Boolean fn).

ie, $f: \{0,1\}^n \to \{0,1\}^n$ - permutation.

s.t $\Pr_{x,A}[A(f(x)) = x] \leq \varepsilon$. $\forall$ ptime alg A

$$\begin{tikzcd} x \arrow[r, bend left, "\text{easy}"] & f(x) \arrow[l, bend left, "\text{hard}"] \end{tikzcd}$$

Qn: Is there any bit about x that is also hard.

(hardcore predicate)

More formally. $\exists B: \{0,1\}^n \to \{0,1\}$.

Hardcore bit B: $\Pr_{x,A}[A(f(x)) = B(x)] \leq \frac{1}{2} + \varepsilon$, $\forall$ alg A

$f: \{0,1\}^n \to \{0,1\}^n$ f is a OWP

$$\left.\begin{array}{l} f': \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n} \\ (x, r) \xmapsto{f'} (f(x), r) \end{array}\right\} f' \text{ is also a OWP}$$

Hardcore Predicate $B: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$

$(x, r) \xmapsto{B} \langle x, r \rangle$

where $\langle a, b \rangle = \sum a_i b_i \pmod 2$.

Theorem: Suppose there is an algorithm A of complexity t s.t

$$\Pr_{x,A}\left[A(f(x),r) = \langle x, r\rangle\right] \geq \frac{1}{2} + \varepsilon$$

$$\Downarrow$$

$\exists$ an alg A' of complexity $O\left(\frac{t\, n^{O(1)}}{\varepsilon^4}\right)$ s.t

$$\Pr_{x,A'}\left[A'(f(x)) = x\right] \geq \Omega(\varepsilon)$$

Hadamard Encoding:

$$x \overset{Had}{\longmapsto} l_x(\cdot) \qquad\qquad x \longmapsto \{l_x(y)\}_{y\in\{0,1\}^n}$$

$$l_x : \{0,1\}^n \to \{0,1\}$$

$$y \mapsto \langle x, y\rangle.$$

$$\underbrace{\boxed{x \qquad\qquad}}_{n} \longmapsto \underbrace{\overbrace{\boxed{\qquad\qquad\qquad\qquad\qquad\qquad\qquad}}^{l_x(\cdot)}}_{2^n}$$

Suppose we have an oracle H s.t

$$\Pr_{r,H}\left[H(r) = \langle x, r\rangle\right] \geq \frac{1}{2} + \varepsilon.$$

$$\Downarrow$$

$\exists$ an ckt that computes x, using oracles calls to H.

Lemma: (Goldreich-Levin Algorithm - Weak Version)

There is an algorithm GLW that given oracle access to a fn $H : \{0,1\}^n \to \{0,1\}$

such that for some $x \in \{0,1\}^n$

$$\Pr_r[H(r) = \langle x, r\rangle] \geq 7/8 \longrightarrow \underline{\text{Weak}}$$

outputs x in time $O(n^2 \log n)$

makes $O(n \log n)$ queries to H.

w/ prob $1-o(1)$.

$\underline{\text{Obs}}$: If $7/8 \to 1$ instead, then on querying $H(e_i)$; e_i - unit vectors can recover x.

$\underline{\text{Now}}$: $x_i = \langle x, e_i\rangle$

$= \langle x, r+e_i\rangle - \langle x, r\rangle$

For a random r.

$\langle x, r+e_i\rangle = H(r+e_i)$ - w/ prob $7/8$

$\langle x, r\rangle = H(r)$ - w/ prob $7/8$

Both are correct w/ prob $6/8 = 3/4$.

Algorithm GLW

- For $j = 1$ to k. $[k = O(\log n)]$
  - Pick random $r_j \in \{0,1\}^n$
- For $i = 1$ to n,

$$x_i = \underset{j}{\text{maj}}\{H(r_j + e_i) - H(r_j)\}$$

- Return x

\# oracle calls $= n(k) + k = O(n \log n)$ oracle calls.

Analysis of Algorithm.

$$\Pr_{x}\left[H(x+e_i) - H(x) = x_i \right] \geq 3/4.$$

$$\Pr_{x_1 \dots x_k}\left[x_i = \text{maj}_j \left\{ H(x_j+e_i) - H(x_j) \right\} \right] \cdots \quad (*)$$

Chernoff Bound:

$X_1 \dots X_n$ – independent 0/1 random variables

$$\forall \varepsilon. \quad \Pr\left[\sum X_i \leq \mathbb{E}\left[\sum X_i\right] - \varepsilon n \right] \leq e^{-2\varepsilon^2 n}.$$

$$X_j = \mathbb{1}\left[H(x_j+e_i) = \langle x, x_j+e_i \rangle \wedge H(x_j) = \langle x, x_j \rangle \right]$$

$$\Pr[X_j = 1] \geq 3/4.$$

$$(*) \geq 1 - e^{-2(1/4)^2 k} \quad \text{(by Chernoff bd)}$$

$k = O(\log n)$
$$\geq 1 - e^{-O(\log n)}$$
$$\geq 1 - \frac{1}{n^2}.$$

$$\Pr[x_i \text{ is correct}] \geq 1 - \frac{1}{n^2}$$

$$\Pr[x \text{ is correct}] \geq 1 - n \cdot \frac{1}{n^2} = 1 - \frac{1}{n}. \quad \checkmark$$

Chebyshev's Bound:

$X_1 \dots X_n$ – pairwise independent 0/1-random variables

$$\Pr\left[\sum X_i < \mathbb{E} X_i - \varepsilon n \right] \leq \frac{\mathrm{Var}\left(\sum X_i\right)}{4\varepsilon^2 n}$$

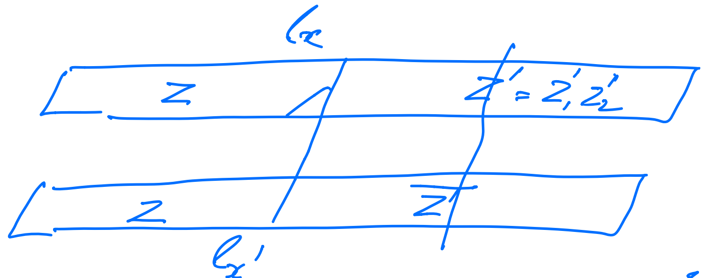
If used Chebyshev instead

$k = n^2$ instead of $k = O(\log n)$.

Remark: GLW can be extended if

$7/8 \to 3/4 + \varepsilon$ for any $\varepsilon \in (0,1)$

$x \neq x'$



$$\Pr_r[\langle x, r\rangle = \langle x', r\rangle] = \frac{1}{2}$$

Can construct H s.t.

$$\Pr_r[H(r) = \langle x, r\rangle] = \frac{3}{4}$$

$$\Pr_r[H(r) = \langle x', r\rangle] = 3/4$$

Lemma: (Goldreich-Levin Algorithm).

There exists an algorithm GL that on oracle access to a fn. $H: \{0,1\}^n \to \{0,1\}$ & an $\varepsilon > 0$, makes $O\left(\frac{n \log n}{\varepsilon^4}\right)$ - oracle calls to H and outputs a list L of at most $O\left(\frac{1}{\varepsilon^2}\right)$ elements s.t if x satisfies

$$\Pr_r[H(r) = \langle x, r\rangle] \geq \frac{1}{2} + \varepsilon$$

then $x \in L$. $\boxtimes$

Consider the following (first attempt).

GL-first-attempt.

1. Pick $r_1 \ldots r_k \in \{0,1\}^n$

2. For all $b_1, \ldots b_k \in \{0,1\}$ (Assume $b_i = \langle x, r_i \rangle$)

   - Define: $H'_{b_1 \ldots b_k}(r) = \operatorname{maj}_j \{ H(r + r_j) - b_j \}$
   - Run GLW on $H'_{b_1 \ldots b_k}$ to obtain some string $x_{b_1 \ldots b_k}$
   - Add to list L.

3. Output list L.

(List-size = $\exp(k)$ $= \exp(\frac{1}{\varepsilon^2})$)

Suppose there exists x & some $b_1 .. b_k$ s.t.

$$\Pr_r [H'_{b_1 \ldots b_k}(r) = \langle x, r \rangle] \geq 7/8.$$

then GLW can extract x from $H'_{b_1 .. b_k}$.

Suppose x satisfies

$$\Pr_r [H(r) = \langle x, r \rangle] \geq \frac{1}{2} + \varepsilon.$$

If $b_1 \ldots b_k$ are guessed correctly

$$\Pr_{r, r_1 \ldots r_k}\left[H'_{b_1 \ldots b_k}(r) = \langle x, r\rangle\right]$$

$$= \Pr_{r, r_1 \ldots r_k}\left[\mathrm{maj}_j\left\{H(r+r_j) - b_j\right\} = \langle x, r\rangle\right]$$

$$= \Pr_{r, r_1 \ldots r_k}\left[\mathrm{maj}_j\left\{H(r+r_j) - \langle x, r_j\rangle\right\} = \langle x, r\rangle\right]$$

$$= \Pr_{r, r_1 \ldots r_k}\left[\mathrm{maj}_j\left\{H(r+r_j) = \langle x, r+r_j\rangle\right\}\right]$$

$$\geq 1 - e^{-2\varepsilon^2 k}$$

$k = O\left(\frac{1}{\varepsilon^2}\right)$

$$\geq \frac{99}{100}.$$

$$\Pr_{r, r_1 \ldots r_k}\left[H'_{b_1 \ldots b_k}(r) = \langle x, r\rangle\right] \geq \frac{99}{100}$$

$$\Pr_{r_1 \ldots r_k}\left[\Pr_r\left[H'_{b_1 \ldots b_k}(r) = \langle x, r\rangle\right] \geq 7/8\right] \geq \cdots 1/2$$

Idea to reduce list-size to $O\left(\frac{1}{\varepsilon^2}\right)$ from $\exp\left(\frac{1}{\varepsilon^2}\right)$ is to choose $r_1 \ldots r_k$ - not completely independently but only pairwise independent (as member of a subspace U of $\{0,1\}^n$ so that it suffices to guess b only for the basis of subspace.

Algorithm GL:

(1) Pick $r_1, \dots r_t \in \{0,1\}^n$ $\quad t = O(\log \frac{1}{\varepsilon})$.

(2) Define $r_S := \sum_{j \in S} r_j$ for all non-empty $S \subseteq \{1, 2, \dots t\}$

(3) For all $b_1 \dots b_t \in \{0,1\}^t$

- Define $b_S := \sum_{j \in S} b_j$
- Define $H'_{b_1 \dots b_t}(x) = \mathrm{maj}_{\phi \neq S \subseteq \{1..t\}} \{H(x + r_S) - b_S\}$
- Apply GLW to $H'_{b_1 \dots b_t}$
- add result to list L

(3) Output L.

If $\langle x, r_i \rangle = b_i \quad \forall\, i \in \{1 \dots t\}$

$\langle x, r_S \rangle = b_S, \quad \forall\, S \subseteq \{1 \dots t\}$.

Analysis of this is similar to before except that instead of Chernoff we will use Chebyshev.

$1 - e^{-2\varepsilon^2 k}$ will be replaced

$$1 - \frac{\mathrm{Var}(R)}{4\varepsilon^2 k} \geq 0.99$$

$$k = 2^t - 1 = O(1/\varepsilon^2).$$

$t = O(\log \frac{1}{\varepsilon})$.

Connections to Coding:

$x \xmapsto{\text{Had}} \ell_x(\cdot)$

Unique Decoding / x exactly $\Leftarrow$ agr $\geq \frac{3}{4} + \varepsilon$: GLW

List decoding $\left\{ \begin{array}{l} \text{output a list} \\ L = O(\frac{1}{\varepsilon^2}) \text{ elts} \\ x. \end{array} \right.$ $\Leftarrow$ agr $\geq \frac{1}{2} + \varepsilon$ GL

distance(Had) $= \frac{1}{2}$.

Hadamard Code is $\left. \begin{array}{l} \text{efficiently} \\ \text{list-decodable} \end{array} \right\}$ GL algorithm.

Suppose there exists a

$C : \{0,1\}^n \to \{0,1\}^m$ ($m = 2^n$ for $C = \text{Had}$)

such

(1) $\text{dist}(C) \leq \frac{1}{2} + \varepsilon$.

(2) There exist an "<u>efficient</u>" list decoding alg A s.t. $\forall$ words $r \in \{0,1\}^m$

$A(r)$ outputs a list L s.t.

If $\delta(C(x), r) \leq \frac{1}{2} - 2\varepsilon \Rightarrow x \in L$.

then we could have used C
instead of Ascl.