6.440 Spring 2013, Lecture 21 & 22

Note Title 4/28/2013

Today

1. finish AEL Construction + GI algorithm
2. Codes & complexity

AEL Codes

3 ingredients:

1. C_0: small code $[d, R \cdot d, \delta \cdot d]_2$
2. G: bipartite graph with n left & right vertices & degree d; ϵ-uniform
3. C_{big}: big code $\{n, k, D\}_{2^{R \cdot d}}$

Combined code: $C = \{n, R \cdot k, ?\}_{2^d}$

Defn: G is ϵ-uniform if $\forall\ X \subseteq L$, $Y \subseteq R$

$$\left| \#E(X,Y) - d \cdot n \cdot \frac{|X|}{n} \cdot \frac{|Y|}{n} \right| \leq \epsilon n.$$

Lemma: $\delta(C) \geq \delta_0 - \frac{\epsilon n}{D \cdot l}$

Proof:

- Let $w \in C_{big}$ be word of distance $\geq D$.
- Let $X \subseteq L$ be set where $w_i \neq 0$.
- Let $Y \subseteq R$ be set where $w_j \neq 0$.
- # non-zero edges $\geq |X| \cdot \delta_0 \cdot d$
- But # edges $\leq E(X, Y)$

$$\leq \frac{|X|}{n} \cdot \frac{|Y|}{n} \cdot d \cdot n + \epsilon n$$

$$\Rightarrow \frac{|Y|}{n} \geq \frac{(|X| \cdot \delta_0 \cdot d - \epsilon n)}{|X| \cdot d}$$

$$= \delta_0 - \frac{\epsilon \cdot n}{|X| \cdot d} \geq \delta_0 - \frac{\epsilon \cdot n}{D \cdot d}$$

Will skip decoding but it leads to $O(n)$ time decoding with $\frac{1}{4} - \epsilon$ errors (in binary code) [Guruswami - Indyk]

Codes & Computational Complexity

Obvious Direction:

- Codes need Encoding + Decoding.
- Needs efficient algorithms.
- When are they possible? intractable?
  - Will briefly discuss today.

Non-obvious direction

- Codes are combinatorial structures with some nice properties.
- Most extremal structures are connected to one another } I
- Errors model uncertainty / lack of knowledge. Often captures adversary } II

Theme I - Pseudorandomness

(See excellent survey by [Salil Vadhan])

Background: Randomized algorithms

Computational Problem

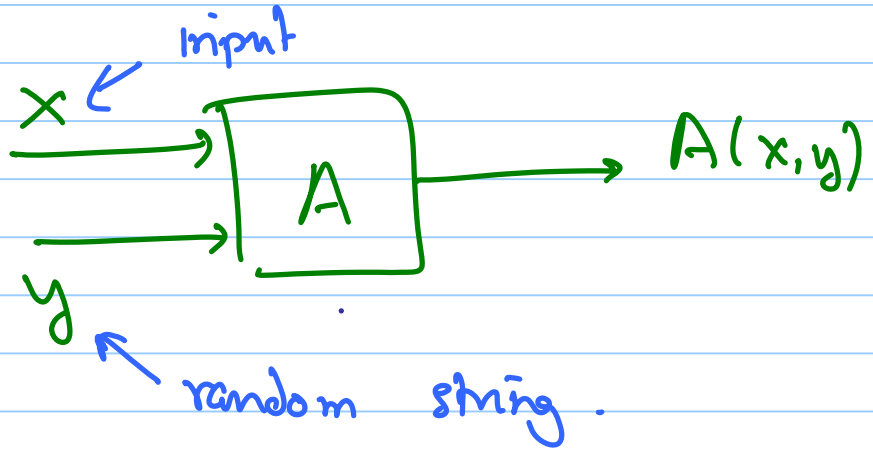
Functional $x \to \boxed{f} \to y = f(x)$

or

Relational $x \to \boxed{R} \to y$ s.t. $(x, y) \in R$

P = class of functional problems solvable in polytime, where range of f is $\{0,1\}$ (Boolean).

Randomized algorithm



A probabilistically computes f if

$$\forall x \quad \Pr_y \left[A(x,y) \neq f(x) \right] \leq \frac{1}{3}$$

A is Polytime if

① $|y| \leq |x|^c$ for constant c.

② running time of A is poly.

Example: MAX 3SAT :

Input: $\phi = C_1, C_2, \ldots C_m$. (clause)

$C_j = x_{i_1(j)}$ or $x_{i_2(j)}$ or $\overline{x_{i_3(j)}}$

literal $\triangleq$ variable or its complement

(Desired) Output: $a_1 \ldots a_n \in \{0,1\}$

that maximizes

$\#\{j \mid C_j$ "satisfied", ie. one of literals in C_j is $1\}$

- Well known: NP-hard to solve optimally.
- Can we do something "near-optimal"?
- $\exists$ prob. $7/8$-approximator:

  finds $\bar{a} = a_1 \ldots a_n$ s.t.

  # satisfied clauses $\geq \left\lfloor \frac{7}{8} m \right\rfloor$

- Alg: Pick $a_1 \dots a_n$ at random (uniformly in $\{0,1\}^n$)

- Analysis:

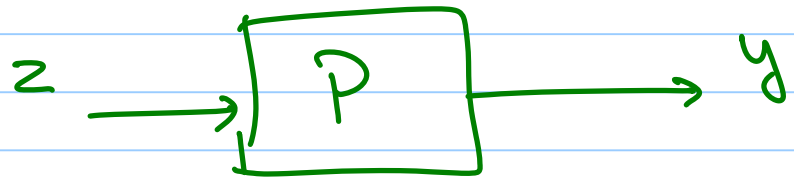
$$\Pr_{\bar{a}}[C_j \text{ satisfied}] = 7/8$$

$$\Rightarrow \mathop{\mathbb{E}}_{\bar{a}}\left[\#\{j \mid C_j \text{ satisfied}\}\right] = \tfrac{7}{8}m.$$

(more formal analysis would pick many vectors $\bar{a}$ & output best).

Question:
- Can we build "deterministic" algorithms?
- Or an algorithm that uses less randomness?
- Or doesn't need "pure" (unbiased, independent) randomness?

The Randomness Processing Industry



$P: \{0,1\}^{\ell} \to \{0,1\}^{n}$

- Exact functionality of P is unimportant.
- Key issue: if $z \sim D_1$ then how is $P(z)$ distributed?
- Pseudorandom generator: (for (A, f))
  - $\ell \ll n$.
  - if z uniform over $\{0,1\}^{\ell}$ then $\forall x$

$$\Pr_z[A(x, P(z)) = f(x)] \approx_{\epsilon} \Pr_y[A(x, y) = f(x)]$$

- Other concepts: Dispersers, Extractors, Condensers, Mergers ...
  (can be seeded/unseeded, lossless/lossy, zero-error/two-sided-error)
- Today: PSEUDO-RANDOMNESS.
- if p.r.g.s exist for every polytime A, with $\ell = O(\log n)$, then $BPP = P$
  ↑
  prob. polytime
- "for every A" - open
  but simple A - like MAX 3SAT above can be derandomized.

I. Limited Independence

- Note that $a_1 \dots a_n$ don't have to be completely independent; only limitedly so.
- Suffices that $\forall\, i, j, k$

  (a_i, a_j, a_k) are uniform.

- **Defn:** t-wise independence

  $\bar{y} = (y_1 \dots y_n) = P(z_1 \dots z_\ell)$ is t-wise independent if

  $\forall\, S \subseteq [n]$, $|S| = \ell$

  $\forall\, b \in \{0,1\}^\ell$

  $$\Pr_z \left[P(z)\big|_S = b \right] = \frac{1}{2^\ell}$$

("$P(z)$ restricted to S is uniform")

Claim: MAX 3SAT algorithm works as well with 3-wise independent sources.

☒

Lemma: Let C be an $[n, k, ?]_2$ code with $C^\perp = [n, n-k, t+1]_2$ code. Let $E: \{0,1\}^k \to \{0,1\}^n$ be encoder of C. Then $\{E(z)\}_{z \leftarrow \{0,1\}^k}$ is t-wise independence

Proof: Follows from definitions.

No codewords in $C^\perp$ of wt. $\leq t$

$\Rightarrow$ No linear dependence in $\leq t$ coordinates of C

$\Rightarrow$ No dependence on $\leq t$ coordinates of C

☒

To make generator good, use smallest k possible $\Rightarrow$ use best possible (highest rate) $C^\perp$.

Using best-known codes

① Pairwise Independence:

C = Hadamard Code = linear function

$C^\perp$ = Hamming Code

$k = \log n$ [w.o. "$O(\cdot)$"]

② t-wise independence:

C = dual-BCH code

$C^\perp$ = BCH code

$k \approx \frac{t}{2} \log n$

③ 3-wise independence

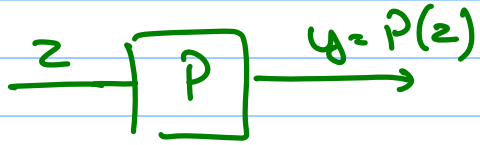
C = affine functions.

$|C| = 2^n$

∎

II. Small-Biased Spaces

Definition:



y is ϵ-biased if $\forall\ S \subseteq [n], S \neq \phi$

$$\left| \Pr_z\left[\bigoplus_{i \in S} y_i = 1\right] - \Pr_z\left[\bigoplus_{i \in S} y_i = 0\right] \right| \leq \epsilon$$

Motivation:

- Definitionally: Output of P "fools" every linear algorithm.
- Real reasons:
  1. Almost Limited Independence
  2. Ingredients in fooling many other algorithms.

δ- almost t-wise independence

$y_1 \dots y_n$ is

if $\forall\, S \subseteq [n]$, $|S| = t$

$$\sum_{b \in \{0,1\}^t} \left| \Pr[y|_S = b] - 2^{-t} \right| \leq \delta$$

Lemma: ϵ-biased space is $\epsilon \cdot 2^t$- almost t-wise independent

Lemma:
- Let $P_1 : \{0,1\}^r \to \{0,1\}^\ell$ generate ϵ-biased bits
- Let $P_2 : \{0,1\}^\ell \to \{0,1\}^n$ be linear & t-wise independent
- Then $P_2 \circ P_1 : \{0,1\}^r \to \{0,1\}^n$ is $(\epsilon \cdot 2^t)$-almost t-wise independent

Lemma: Let $G \in \{0,1\}^{k \times N}$ be the generator of code of distance $(\frac{1}{2} - \epsilon)N$. & let $\uparrow^N \in \text{code}(G)$.

Then the map

$$P: \{0,1\}^{\log N} \to \{0,1\}^k$$

that maps $i \mapsto i^{th}$ column of G is an ϵ-biased generator. ⊠

Putting all together with MAX 10SAT

Method	fraction clauses set.	Sample space.
Random assignment	$1-2^{-10}$	2^n
10-wise indep.	$1-2^{-10}$	n^5
ϵ-bias with $\epsilon = 2^{-22}$	$1-2^{-10}-2^{-11}$	$O(\frac{n}{\epsilon^3})$
$\epsilon 2^{10}$-almost 10-wise indep.	"	$(\frac{\log n}{\epsilon^3})^5$