Lecture 23

5/7/2012

Note Title 5/6/2012

Today

Locally Decodable Codes

- Definition / Motivation
- LDCs via multiv. polynomials
- LDCs via multiplicities.
- LDC from matching vectors

  (matching vector construction).

Definition

ℓ-LDC : maps $\Sigma^k \longrightarrow \Sigma^n$

corrects ϵ-fraction error ℓ-locally

i.e., given oracle access to y s.t.

$\exists m$ s.t. $\Delta(y, E(m)) \leq \epsilon \cdot n$

$\forall$ input $i \in [k]$

$$\Pr\left[D^y(i) = m_i\right] \geq \frac{2}{3}$$

— x —

Motivation:

Current Encodings :

- Either store all info in 1 big block
  - good error-correction, but slow ($\approx$ length of block).
- Or break into small pieces
  - quick correction, but $\Pr[\exists \text{error}] \longrightarrow 1$

LDC's try to bridge gap.

Sublinear LDCs via Multiv. Polynomials

Idea:
- Need code to have local "redundancies" (small set of coordinates that have dependencies)
- Reed-Solomon has no such redundancies
- Idea: - use low-degree multiv. polynomials
  - locality? On every line, function values restricted.

General setting:

- \# variables $= m$
- degree $= d$
- Field size $= q > \frac{d}{(1-2\epsilon)}$
- Code = Evaluations of deg. d polys in m variables.
- Resulting parameters

$$n = q^m$$

$$k = \binom{d+m}{m} \geq \left(\frac{d}{m}\right)^m \approx \frac{(1-2\epsilon)^m}{m^m} \cdot n$$

$$d = (1-2\epsilon) \cdot n$$

$$\text{locality } \ell = q = n^{1/m}$$

Some interesting choices:

- $m = \frac{1}{\epsilon}$: $\frac{k}{n} \approx \epsilon^{1/\epsilon}$

  $\ell = n^{\epsilon}$

- $q = O(1)$: $n = \exp(k^{1/q-1})$

  $\ell = q$

- $m = \frac{\log n}{\log \log n}$ $n = \text{poly}(k)$

  $\ell = \text{poly} \log n$.

Initial Beliefs:

Maybe roughly best possible behavior?

- locality $\to n \Rightarrow \frac{k}{n} \to o(1)$?
- locality $O(1) \Rightarrow n = \exp(k^{\epsilon})$?

Multiplicity Codes : [Kopparty - Saraf - Yekhanin]

Messages: deg. d, m-variate polynomials.

Encoding: Evaluations of message polynomial, and all its derivatives upto order s,

Alphabet: $\Sigma = \mathbb{F}_q^{\binom{m+s}{m}}$

\# such derivatives

Key Lemma: "Multiplicity Schwartz-Zippel"

$p \in \mathbb{F}_q[x_1 \ldots x_m]$, $\deg p \leq d$, $p \neq 0$

$\Rightarrow \Pr_{\bar{a}}$ [p & all its partial derivatives of order s vanish at $\bar{a}$] $\leq \frac{d}{(s+1) \cdot q}$

Parameters

- $k = \dfrac{\binom{d+m}{m}}{\binom{m+s}{m}} \approx \left(\dfrac{d+m}{m+s}\right)^m \to \left(\dfrac{d}{s}\right)^m$

- $q \approx \dfrac{d}{s}(1-\epsilon)$

- $n = q^m = \dfrac{k}{(1-\epsilon)^m}$

By letting m grow, & then s grow even faster, can have

$$\ell = n^\delta, \quad \frac{k}{n} = (1-\delta) \quad \text{simultaneously!}$$

$O(1)$-Locality Regime [Yekhanin '07]
[Raghavendra '08]
[Efremenko '09]

Main Result: $\exists$ codes with $\ell = 3$

& $n = \exp(\exp(\sqrt{\log k}))$

(compare with $n = \exp(k^{1/2})$)

- More generally $\ell = O(1)$

$$n = \exp(\exp((\log k)^{\epsilon}))$$

Construction without intuition in rest of notes.

Ingredients

- Parameter $m \in \mathbb{Z}^+$ (small ..)
- field $\mathbb{F}_q$ with $m \mid q-1$

(so $\mathbb{F}_q$ has primitive m^{th} root)

- $S \subseteq \mathbb{Z}_m$, $0 \notin S$
- "S-Nice" matrix $M \in \mathbb{Z}_m^{k \times n}$

<u>Defn</u>: M is S-nice if, for

$$M = k\Big[\underbrace{M_1}_{k} \;\Big|\; \underbrace{M_2}_{n-k}\Big]$$

① $(M_1)_{ii} = 0$, ② $(M_1)_{i,j} \in S \quad \forall\, i \neq j$

③ M is closed under column sums.

$\mathbb{Z}_m$- matrices $\Longrightarrow$ $\mathbb{F}_q$- matrices

$$m_{ij} \longmapsto g^{m_{ij}} = G_{ij}$$

[g primitive m^{th} root in $\mathbb{F}_q$]

Theorem: G is generator of $(|S|+1)$-LDC

(Terminology: G is "generator" of the encoding map $x \longmapsto x \cdot G$)

Defn: $p \in \mathbb{F}_q[x]$ is S-zeroing poly if

1. $p(1) = 1$
2. $p(g^s) = 0 \quad \forall s \in S$

Defn: $p \in \mathbb{F}_q[x]$ is t-sparse if

p has at most t non-zero coefficients.

Lemma: If S has a t-sparse S-zeroing poly, then $G = G(m)$ is t-LDC.

Proposition: Every $S \subseteq \mathbb{Z}_m^*$ has a $(|S|+1)$-zeroing polynomial.

(Lemma + Proposition $\Rightarrow$ Theorem)

Proof of Lemma

to be filled.

Construction of nice matrices

to be added.