CS 229R - LECTURE 20

4/6/17

TODAY:
1) Finish High Rate LDPCs
2) LTC constructions
   2.1) Redundant LDPCs
   2.2) Tensor codes & Testing \( \Rightarrow \) RM-lift perf.
   2.3) Zig-Zag codes & Testing \( \Rightarrow \) polylog locality

---

LDC: See notes for Lecture 19

---

Terminology: LDPC codes: Low Density Parity Check

\( \Rightarrow \) \text{Gallager codes}

\( \Rightarrow \) \text{Tanner codes}

\( \Rightarrow \) \text{Sipser-Spielman codes}

Low Density: Parity Check Matrix = Sparse

\( \Rightarrow \) Underlying graph sparse

\( \Rightarrow \) average constant degree
**Contrast:** LDPC & LTCs

**LTC:** Pick \( S \subseteq [n] \); \( |S| \leq \ell \)

& verify \( \chi_S \in V_S \subseteq \mathbb{F}_2^{\leq |S|} \)

(i) \( \text{LTC} \Rightarrow \text{LDPC} \)

\[ X_1 \circ \]
\[ X_2 \circ \]
\[ \vdots \]
\[ X_n \circ \]

"Strong Soundness" \( \Rightarrow \) \( \forall S \subseteq V_S \exists x \in \mathbb{C} \)

\[ \mathbb{P}[x \neq v_S] \geq \epsilon \cdot S(x, c) \]

\( \Rightarrow S(x, c) = 0 \Rightarrow x \in \mathbb{C} \).

(ii) But \( \text{LDPC} \not\Rightarrow \text{LTC} \) [Bengtsson Harsh Raskhodnikova]
- Key issue: Need redundant local constraints.

1. Removing $\frac{1}{12}$ one or few or constant fraction of constraints should not change the code.

- How can we get redundancy.

  o Ans 1: By building code with many symmetries.
    E.g. RM codes have constraint for each line.

    # coordinates = $n = 2^m$
    # constraints = $\binom{2^m}{2} \approx n^2 \gg n$.

  o Ans 2: Tensor Products [Ben-Sasson-Sudan]

  Tensor Product Code

  $C_1 = [n_1, k, d_1]_q$  &  $C_2 = [n_2, k_2, d_2]_q$

  $C \otimes C_2 = [n_1 n_2, k_1 k_2, d_1 d_2]_q$ code with codewords

  \[
  \begin{bmatrix}
  n_1 & \ell \\
  \ell & n_2 \\
  \end{bmatrix}
  \in C_2
  \]

  with every column in $C_1$ and every row in $C_2$. 
Exercise: Prove dimension & distance

Redundancy in Tensor Product Codes

Let $C_1 = C_2 = C$.

- Code has length $n^2$.
- But specified by $2n$ constraints; assume $C$ is a systematic code with first $k$ coordinates being message.

Then given $k \times k$ matrix $M$, its encoding may be obtained by expanding rows into codewords of $C$ giving matrix $M' \in \mathbb{F}^{k \times n}$

Then expanding columns into codewords of $C$ giving $n \times n$ matrix $M''$.

$$M \rightarrow M' \hspace{1cm} \rightarrow \hspace{1cm} M''$$

- Redundant constraints: bottom $n-k$ rows of $M'' \in C$!

Code specified by taking top $k$ rows in $M'' \{C\}$ & all columns in $M''$. 
Basic Testing Question

If $\Pr \left[ \text{random row } \in C \right] \geq \frac{\alpha}{2} - 1 - \varepsilon$

and $\Pr \left[ \text{random column } \in C \right] \geq 1 - \varepsilon$

then $S(M, C \otimes C) \leq 1 - \varepsilon$ ?

Robust Testing Question

If $\frac{1}{2} \left[ \mathbb{E}_{\text{row}} \left[ S(M_{\text{row}}, C) \right] + \mathbb{E}_{\text{column}} \left[ S(M_{\text{column}}, C) \right] \right] \leq \varepsilon \Rightarrow \varepsilon \in \beta$

$\Rightarrow S(M, C \otimes C) \leq \varepsilon \Rightarrow \alpha \in \beta$

Why Robustness?

If $C \otimes C$ - test is $\alpha$-robust

and $C$ is $(\varepsilon, \delta)$-locally testable,

then $C \otimes C$ is $(\frac{\varepsilon}{\alpha}, \varepsilon)$-locally testable.

$\uparrow$ Proposition $\uparrow$

Hope: $\forall S \exists \delta \text{ s.t. if } C \text{ is a code } \delta \text{ dist } S$

[BSS] then $C \otimes C$ - test for $C \otimes C$ is $\alpha$-robust.

Example [Paul Valiant]: No.
Generalizing: \( C^\otimes m = C \otimes C \otimes \ldots \otimes C \)

1. **Test of \( C^\otimes m \):** Pick random \( l \)-dim axis parallel surface \( S \)
   Verify \( (M \mid S, C^{\otimes l}) \leq \alpha_{\text{smal}} \)

**Theorem [Ben-Sasson, S., Viderman]**

\[ \forall m \geq 3 \exists \alpha \quad \text{s.t. if } S(C) \geq \alpha \text{ then } \]

\( C^{\otimes m-1} \) test for \( C^\otimes m \) is \( \alpha \)-robust.

**Corollary**

\[ \forall m \geq 3 \forall S \exists \alpha' \quad \text{s.t. if } S(C) \geq \alpha' \text{ then } \]

\( C^{\otimes 2} \) test for \( C^\otimes m \) is \( \alpha \)-robust.

**Proposition:**

- Test for \( B \) is \( \alpha_1 \)-robust
- Test for \( C \) is \( \alpha_2 \)-robust

\[ \Rightarrow \text{ Test for } C \text{ is } (\alpha_1 \cdot \alpha_2) \text{-robust}. \]
Using Tensor Codes directly

Thm [Viderman]: \( \forall \alpha, \beta > 0 \exists S \text{ s.t. for large } N \exists 
\text{ codes of length } N, \text{ Rate } \geq 1 - \alpha, \text{ Locality } \leq N^\beta, \text{ distance } \geq S \)

**Proof:** Exercise (analogous to Kopparty-Saraf, Yekhanin).

But dependence of \( S \) on \( \alpha, \beta \) much worse (\( S = (\alpha^\beta)^{1/\beta} \)).

Tensor Codes + Distance Amplification

Thm [KMRS1]: \( \exists n^{o(1)} \) function \( c(n) \) st. \( \forall S \exists 
\text{ codes of rate } 1 - S - o(1) \text{ that are } c(n) \text{-locally testable.} \)

But: Can do much better:
We have two operations:

- **Tensor Product**: Makes code longer; Preserves locality; Makes distance worse; Makes rate worse.

- **A-L Transform**: Code longer; Loses locality; Improves distance; Makes rate worse.

Idea: Will pick $s_0 = o(1)$

$m$ repetitions of Tensor Product followed by A-L Transform

<table>
<thead>
<tr>
<th>Rate</th>
<th>$R \rightarrow R^2 \rightarrow R^2 - s_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>dist</td>
<td>$s_0 \rightarrow s_0^2 \rightarrow s_0$</td>
</tr>
<tr>
<td>locality</td>
<td>$l \rightarrow l \rightarrow l/s_0$</td>
</tr>
<tr>
<td>length</td>
<td>$n_i \rightarrow n_i^2 \rightarrow n_i^{2i} = n_{i+1}$</td>
</tr>
</tbody>
</table>

After start with $n_0 = \text{near const.} \over poly(\sqrt{s_0})$; $R = 1 - s_0$

$n_i = n_0^{2i}; R_i = R - 2^i s_0; \ell_i = \ell / s_i$

Yields codes of length $N$ with $R = 1 - o(1); \ell_i = (\log N)$
(Need some care)

Tensor products testable only by $\square$'s .... But can work this out. Details omitted.

Concluding:

LTCs $\rightarrow$ very strong performance
LDCs $\rightarrow$ weaker but substantially $o(n)$ for no-cost!
LRCs $\rightarrow$ $\equiv$

Other concepts

Relaxed LDCs: Either recover a bit or say "Here are too many errors".

Usually as good as LTCs...