Locality in Codes and Lifting

Madhu Sudan
MSR

Joint work with Alan Guo (MIT) and Swastik Kopparty (Rutgers)
Error-Correcting Codes

• (Linear) Code $C \subseteq \mathbb{F}_q^n$.
  
  – $n \overset{\text{def}}{=} \text{block length}$
  – $k = \text{dim}(C) \overset{\text{def}}{=} \text{message length}$
  – $R(C) \overset{\text{def}}{=} k/n$: Rate of $C$ (want as high as possible)
  – $\delta(C) \overset{\text{def}}{=} \min_{x \neq y \in C} \{\delta(u, v) \overset{\text{def}}{=} \Pr_i[u_i \neq v_i]\}$.

• Basic Algorithmic Tasks
  
  – Encoding: map message in $\mathbb{F}_q^k$ to codeword.
  
  – Testing: Decide if $u \in C$
  
  – Correcting: If $u \notin C$, find nearest $v \in C$ to $u$. 

Locality in Algorithms

• “Sublinear” time algorithms:
  – Algorithms that run in time $o(\text{input})$, $o(\text{output})$.
  – Assume random access to input
  – Provide random access to output
  – Typically probabilistic; allowed to compute output on approximation to input.

• LTCs: Codes that have sublinear time testers.
  – Decide if $u \in C$ probabilistically.
  – Allowed to accept $u$ if $\delta(u, C)$ small.

• LCCs: Codes that have sublinear time correctors.
  – If $\delta(u, C)$ is small, compute $v_i$, for $v \in C$ closest to $u$. 
LTCs and LCCs: Formally

• $C$ is a $(\ell, \epsilon)$-LTC if there exists a tester that
  – Makes $\ell(n)$ queries to $u$.
  – Accepts $u \in C$ w.p. 1
  – Rejects $u$ w.p. at least $\epsilon \cdot \delta(u, C)$.

• $C$ is a $(\ell, \epsilon)$-LCC if there exists decoder $D$ s.t.
  – Given oracle access $u$ close to $v \in C$, and $i$
  – Decoder makes $\ell(n)$ queries to $u$.
  – Decoder $D^u(i)$ usually outputs $v_i$.
  
  • $\Pr_i[D^u(i) \neq v_i] \leq \delta(u, v) / \epsilon$

• Often: ignore $\epsilon$ and focus on $\ell$
Example: Multivariate Polynomials

• Message = multivariate polynomial; Encoding = evaluations everywhere.
  – \( \text{RM}[m, d, q] \overset{\text{def}}{=} \{ \langle f(\alpha) \rangle_{\alpha \in \mathbb{F}_q^m} \mid f \in \mathbb{F}_q[x_1, \ldots, x_m], \deg(f) \leq d \} \)

• Locality?
  – Restrictions of low-degree polynomials to lines yield low-degree (univ.) polys.
  – Random lines sample \( \mathbb{F}_q^m \) uniformly (pairwise ind’ly)
LDCs and LTCs from Polynomials

• Decoding ($d \leq q$):
  – Problem: Given $f \approx p$, $\alpha \in \mathbb{F}_q^m$, compute $p(\alpha)$.
  – Pick random $\beta$ and consider $f|_L$ where $L = \{\alpha + t \beta \mid t \in \mathbb{F}_q\}$ is a random line $\exists \alpha$.
  – Find univ. poly $h \approx f|_L$ and output $h(\alpha)$

• Testing ($d \leq q$):
  – Verify $\deg(f|_L) \leq d$.

• Parameters:
  – $n = q^m$; $\ell = q = n^\frac{1}{m}$; $R(C) \approx \left(\frac{1}{m}\right)^m$
Decoding Polynomials

• $d < q$
  – Correct more errors (possibly list-decode)
  – can correct $\approx 1 - \sqrt{d/q}$ fraction errors \cite{STV}.

• $d > q$
  – Distance of code $\delta \approx q^{-\frac{d}{q-1}}$
  – Decode by projecting to $\approx \frac{d}{q-1}$ dimensions. “decoding dimension”.
  – Locality $\approx 1/\delta$.
  – Lots of work to decode from $\approx \delta$ fraction errors \cite{GKZ,G}.
  – Open when $q = d = 3$ \cite{Gopalan}.
Testing Polynomials

• $d \ll q$:
  – Even slight advantage on test implies correlation with polynomial.\[RS, AS\]

• $d > q$:
  – Testing dimension $t = \frac{d}{q^{s/p}}$; where $q = p^s$ ;
  – Project to $t$ dimensions and test.
  – $(q^t, \min\{\epsilon_q, q^{-2t}\})$-LTC.
Testing vs. Decoding dimensions

• Why is decoding dimension $d/(q - 1)$?
  – Every function on fewer variables is a degree $d$ polynomial. So clearly need at least this many dimensions.

• Why is testing dimension $d/(q - q/p)$?
  – Consider $q = 2^s$, $d = \frac{q}{2}$ and $f = x^d y^d$.
  – On line $y = ax + b$,
  – $f = x^d (ax + b)^d = x^d (a^d x^d + b^d) = a^d x + b^d x^d$.
  – So $\deg(f) = q$, but $f$ has degree $\leq d$ on every line!
  – In general if $q = p^s$ then powers of $p$ pass through ( ... )
  – Aside: Using more than testing dimension has not paid dividend with one exception [RazSafra]
Other LTCs and LDCs

• Composition of codes yields better LTCs.
  – Reduces $\ell(\cdot)$ (to even 3) without too much loss in $R(C)$.
  – But till recently, $R(C) \leq \frac{1}{2}$

• LDCs
  – 2007+ [Yekhanin, Raghavendra, Efremenko] – great improvements for $\ell(n) = O(1); n = \text{superpoly}(k)$.
  – 2010 [KoppartySarafYekhanin] Multiplicity codes get $R(C) \rightarrow 1$ with $\ell(n) = n^\epsilon$
  – For $\ell(n) = \log n$; multiv. Polys are still best known.
Today

- New Locally Correctible and Testable Codes from “Lifting”.
  - $R(C) \to 1; \ell(n) = n^\epsilon$ for arbitrary $\epsilon > 0$.
  - First “LTCs” to achieve this?
  - Only the second “LCCs” with this property
    - After Multiplicity codes [KoppartySarafYekhanin]
The codes

- Alphabet: $\mathbb{F}_q$
- Coordinates: $\mathbb{F}_q^m$
- Parameter: degree $d$
- Message space:
  \[ \{ f : \mathbb{F}_q^m \to \mathbb{F}_q \mid \deg(f|_L) \leq d, \forall \text{ lines } L \} \]
- Code: Evaluations of message on all of $\mathbb{F}_q^m$
- And oh ... $q = 2^s$; $d = (1 - \epsilon)q$; $m = O(1)$
Recall: Bad news about $\mathbb{F}_2^s$

- Functions that look like degree $d$ polynomials on every line $\neq$ degree $d m$-variate polynomials.
- But this is good news!
  - Message space includes all degree $d$ polynomials.
  - And has more.
  - So rate is higher!
  - But does this make a quantitative difference?

- As we will see ... **YES!** Most of the dimension comes from the ``illegitimate” functions.
Generalizing: Lifted Codes

• Consider \( B \subseteq \{ F^t_Q \rightarrow F_q \} \).
  
  – \( F_Q \) extends \( F_q \)
  
  – Preferably \( B \) invariant under affine transformations of \( F^t_Q \).

• Lifted code \( C \mathrel{\overset{\text{def}}{=} \text{Lift}_m(B)} \subseteq \{ F^m_Q \rightarrow F_q \} \)
  
  – \( C = \{ f \mid f|_A \in B, \forall t\text{-dim. affine subspaces } A \} \).

• Previous example:
  
  – \( B = \{ f : F_q \rightarrow F_q \mid \deg(f) \leq d \} \)
Properties of lifted codes

• Distance:
  \[ \delta(C) \geq \delta(B) - Q^{-t} + Q^{-m} \approx \delta(B) \]

• Local Decodability:
  – Same decoding algorithm as for RM codes.
  – \( B \) is \( (\ell, \epsilon) \)-LDC implies \( C \) is \( (\ell, \Omega(\epsilon)) \)-LDC.

• Local Testability?
Local Testability of lifted codes

• Local Testability:
  – Test: Pick \( A \) and verify \( f|_A \in B \).
  – “Single-orbit characterization”: \( (Q^t, Q^{-2t})\)-LTC [KS]
  – (Better?) analysis for lifted tests: \( (Q^t, \epsilon_Q)\)-LTC [HRS]
    (extends [BKSSZ,HSS] )

• Musings:
  – Analyses not robust (test can’t accept if \( f|_A \approx B \).)
  – Still: generalizes almost all known tests ... [Main exceptions – [ALMSS,PS,RS,AS] ].
  – Key question: what is \( K \) s.t. \( f|_{A_1}, ..., f|_{A_K} \in B \) \( \Rightarrow \)
    there exists an interpolator \( g \in C \) s.t. \( g|_{A_i} = f|_{A_i} \)
Returning to (our) lifted codes

- Distance ✓
- Local Decodability ✓
- Local Testability ✓
- Rate?
  - No generic analysis; has to be done on case by case basis.
  - Just have to figure out which monomials are in \( C \).
Rate of bivariate Lifted RS codes

• \( B = \{ f \in \mathbb{F}_q[x] \mid \deg(f) \leq d = (1 - \epsilon)q \}; \quad q = 2^s \)
  – Will set \( \epsilon = 2^{-c} \) and let \( c \to \infty \).

• \( C = \{ f : \mathbb{F}_q[x, y] \mid f|_{y=ax+b} \in B, \forall a, b \} \)
  – When is \( x^i y^j \in C \)?
  – Clearly if \( i + j \leq d \); But that is at most \( \frac{q^2}{2} \) pairs.
  – Want \( \approx \frac{q^2}{2} \) more such pairs.
  – When is every term of \( x^i(ax + b)^j \mod(x^q - x) \)
    of degree at most \( d \)?
Lucas’s theorem & Rate

• Notation: \( r \leq_2 j \), if \( r = \sum_i r_i 2^i \) and \( j = \sum_i j_i 2^i \) (\( r_i, j_i \in \{0,1\} \)) and \( r_i \leq j_i \) for all \( i \).

• Lucas’s Theorem: \( x^r \in \text{supp}\left( (ax + b)^j \right) \) iff \( r \leq_2 j \).

• \( \Rightarrow \text{supp}(x^i(ax + b)^j) \ni x^{i+r} \) iff \( r \leq_2 j \)

• So given \( i, j; \exists r \leq_2 j \) s. t. \( i + r \mod q > d \)?
Binary addition etc.

\[ u = i + r \]

\[
\begin{array}{cccccc}
\text{lsb} & \text{msb} \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}
\]

\[ i_{k-1} = 0 \land i_k = 0 \land j_{k-1} = 0 \land j_k = 0 \implies u_k = 0 \]

\[ u_k = 0 \implies u \leq d \]

\[
\Pr_{i,j}[i_{k-1} \ldots j_k \neq 0000] \leq \frac{15}{16} \]

\[
\Pr_{i,j}[i + r \ (\text{mod}'q) > d] \leq \left(\frac{15}{16}\right)^c \quad \text{as } c \to \infty
\]
Other lifted codes

- Best LCC with $O(1)$ locality.
  - $B = \{ f : \mathbb{F}_{2^s} \rightarrow \mathbb{F}_2 \ | \sum_a f(a) = 0 \}$
  - $s = \log_2 \ell = O(1)$
  - $C = \text{Lift}_m(B)$
  - $n = 2^{sm}; \ell$-LCC; $\dim(C) = (\log n)^\ell$

- Alternate codes for BGHMRS construction:
  - $B = \{ f : \mathbb{F}_4^{m - \log 1/\epsilon} \rightarrow \mathbb{F}_2 \ | \sum_a f(a) = 0 \}$
  - $C = \text{Lift}_m(B)$
  - $\ell = \epsilon n; \dim(C) = n - \text{polylog}(n)$
Nikodym Sets

- $N \subseteq \mathbb{F}_q^m$ is a Nikodym set if it almost contains a line through every point:
  - $\forall a \in \mathbb{F}_q^m, \exists b \in \mathbb{F}_q^m$ s.t. $\{a + tb \mid t \in \mathbb{F}_q\} \subseteq N \cup \{a\}$

- Similar to Kakeya Set (which contain line in every direction).
  - $\forall b \in \mathbb{F}_q^m, \exists a \in \mathbb{F}_q^m$ s.t. $\{a + tb \mid t \in \mathbb{F}_q\} \subseteq K$

- [Dvir], [DKSS]: $|K|, |N| \geq \left(\frac{q}{2}\right)^m$
Proof ("Polynomial Method")

- Find low-degree poly \( P \neq 0 \) s.t. \( P(b) = 0, \forall b \in N \).
- \( \deg(P) < q - 1 \) provided \( |N| < \binom{m+q-2}{m} \).
- But now \( P|_{L_a} = 0, \forall \) Nikodym lines \( L_a \Rightarrow P(a) = 0 \ \forall a \), contradicting \( P \neq 0 \).
- Conclude \( |N| \geq \binom{m+q-2}{m} \approx \frac{q^m}{m!} \).
- Multiplicities, more work, yields \( |N| \geq \left( \frac{q}{2} \right)^m \).
- But what do we really need from \( P \)?
  - \( P \) comes from a large dimensional vector space.
  - \( P|_L \) is low-degree!
  - Using \( P \) from lifted code yields \( |N| \geq (1 - o(1))q^m \) (provided \( q \) of small characteristic).
Conclusions

• Lifted codes seem to extend “low-degree polynomials” nicely:
  – Most locality features remain same.
  – Rest are open problems.
  – Lead to new codes.

• More generally: Affine-invariant codes worth exploring.
  – Can we improve on multiv. poly in polylog locality regime?
Thank You