Locality in Coding Theory

Madhu Sudan
MSR

Error-Correcting Codes

- (Linear) Code $C \subseteq \mathbb{F}_q^n$.
  - $n \stackrel{\text{def}}{=}$ block length
  - $k = \dim(C) \stackrel{\text{def}}{=}$ message length
  - $R(C) \stackrel{\text{def}}{=} k/n$: Rate of C
  - $\delta(C) \stackrel{\text{def}}{=} \min_{x \neq y \in C} \{\delta(u, v) \stackrel{\text{def}}{=} \Pr_i[u_i \neq v_i]\}$.
- 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(input), o(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 (ℓ, ϵ)-LTC if there exists a tester that
  - Makes $\ell(n)$ queries to u.
  - Accept $u \in C$ w.p. 1
  - Reject u w.p. at least $\epsilon \cdot \delta(u, C)$.
- C is a (ℓ, ϵ)-LCC if 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 ϵ and focus on ℓ

Outline of this talk

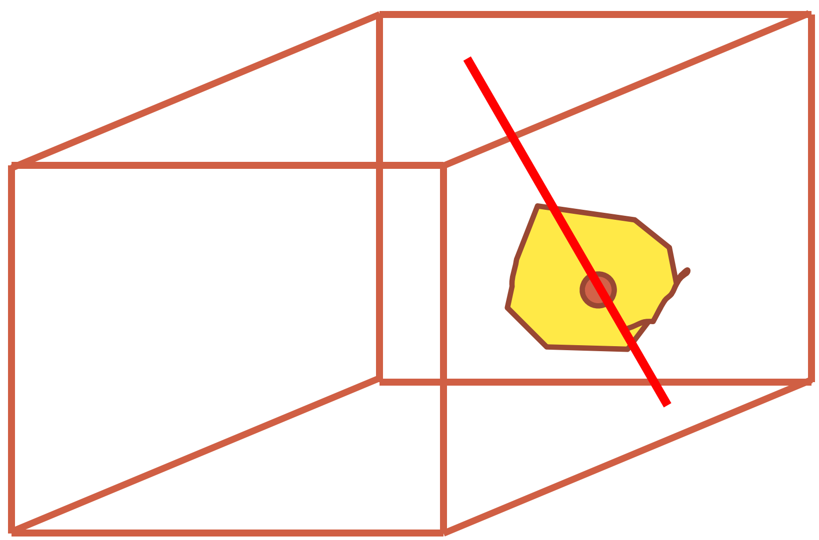
- Part 0: Definitions of LTC, LCC
- Part 1: Elementary construction
- Part 2: Motivation (historical, current)
- Part 3: State-of-the-art constructions
- Part 4 (brief): Towards practicality

Part 1: Elementary Construction

Main Example: Reed-Muller Codes

- Message = multivariate polynomial;
  Encoding = evaluations everywhere.
  - $\mathrm{RM}[m, r, q] \stackrel{\text{def}}{=}$
    $\{\langle f(\alpha)\rangle_{\alpha \in \mathbb{F}_q^m} \mid f \in \mathbb{F}_\mathrm{q}[x_1, \ldots, x_m], \deg(f) \leq r\}$

- Locality? Say $r < q$
  - 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 ($r < q$):
  - Problem: Given $f \approx p, \alpha \in \mathbb{F}_q^m$, compute $p(\alpha)$.
  - Pick random β and consider $f|_L$ where $L = \{\alpha + t\,\beta \mid t \in \mathbb{F}_q\}$ is a random line $\ni \alpha$.
  - Find univ. poly $h \approx f|_L$ and output $h(\alpha)$
- Testing ($r \leq q$):
  - Verify $\deg(f|_L) \leq r$ for random line L

Analysis non-trivial

- Parameters:
  - $n = q^{r}$

Ideas can be extended to $r > q$.

Locality $\approx q^{\frac{r}{q}}$

Part 2: Motivations

Motivation – 1 (“Practical”)

- How to encode massive data?
  - Solution I
    - Encode all data in one big chunk
    - Pro: Pr[failure] = exp(-|big chunk|)
    - Con: Recovery time ~ |big chunk|
  - Solution II
    - Break data into small pieces; encode separately.
    - Pro: Recovery time ~ |small|
    - Con: Pr[failure] = #pieces X Pr[failure of a piece]
  - Locality (if possible): Best of both Solutions!!

Aside: LCCs vs. other Localities

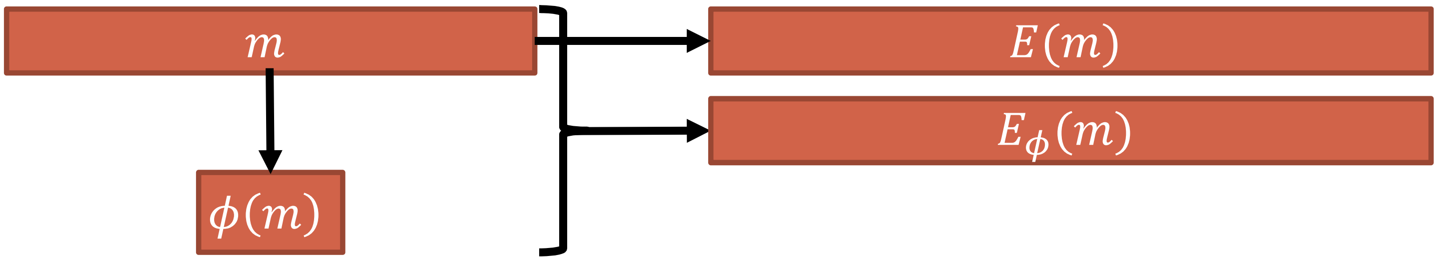
- Local Reconstruction Codes (LRC):
  - Recover from few (one? two?) erasures locally.
  - AND Recover from many errors globally.
- Regenerating Codes (RgC):
  - Restricted access pattern for recovery: Partition coordinates and access few symbols per partition.
- Main Differences:
  - #errors: LCCs high vs LRC/RgC low
  - Asymptotic (LCC) vs. Concrete parameters (LRC/RgC)

Motivation – 2 ("Theoretical")

- (Many?) mathematical consequences:
  - Probabilistically checkable proofs:
    - Use specific LCCs and LTCs
  - Hardness amplification:
    - Constructing functions that are very hard on average from functions that are hard on worst-case.
    - Any (sufficiently good) LCC $\Rightarrow$ Hardness amplification
  - Small set expanders (SSE):
    - Usually have mostly small eigenvalues.
    - LTCs $\Rightarrow$ SSEs with many big eigenvalues [Barak et al., Gopalan et al.]

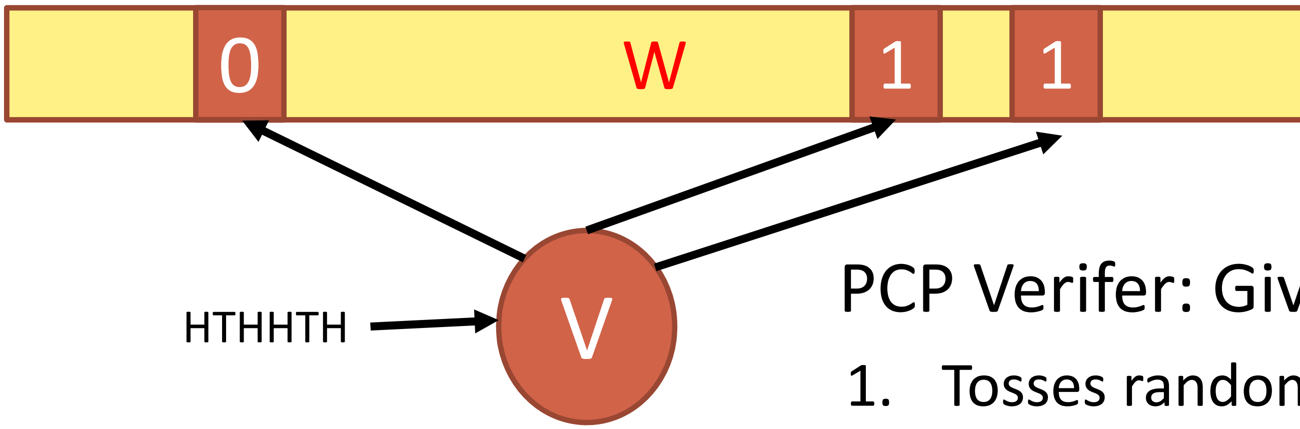
Aside: PCPs (1 of 3)

- Familiar task: Protect massive data $m \in \{0,1\}^k$



- PCP task: Protect m + analysis $\phi(m) \in \{0,1\}$.
  - $\phi(m)$ is just one bit long – would like to read & trust $\phi(m)$ with few probes.
  - Can we do it? Yes! PCPs!
  - "Functional Error-correction"

PCPs (2 of 3) - Definition



PCP Verifer: Given $W \in \{0,1\}^N$

1. Tosses random coins
2. Determines query locations
3. Reads locations. Accepts/Rejects

$W \approx E_\phi(m)$ with $\phi(m) = 1 \Rightarrow$ V accepts w.h.p.

W far from every $E_\phi(m)$ with $\phi(m) = 1 \Rightarrow$ rejects w.h.p.

Distinguishes $\phi^{-1}(1) \neq \emptyset$ from $\phi^{-1}(1) = \emptyset$

PCPs (3 of 3): "Polynomial-speak"

- $m \to M(x)$ low-degree (multiv.) polynomial
- $\phi \to \Phi$: local map from poly's to poly's
- $\phi(m) = 1 \Leftrightarrow \exists\, A, B, C$ s.t. $\Phi(M, A, B, C) \equiv 0$
- $E_\phi(m) = (\langle M \rangle, \langle A \rangle, \langle B \rangle, \langle C \rangle)$ (evaluations)
- Local testability of RM codes ⇒ can verify $E_\phi(m)$ syntactically correct. ($\langle M \rangle, \langle A \rangle, \langle B \rangle, \langle C \rangle \approx$ polynomials)
- Distance of RM codes + $\Phi(M, A, B, C)[a] = 0$ for random a ⇒ Semantically correct ($\phi(m) = 1$)).

Part 3: Recent Progress on LCCs + LTCs

Summary of Recent Progress

- Till 2010: $\mathrm{locality}(n) = o(n) \Rightarrow Rate < \frac{1}{2}$.
- 2015: $\mathrm{locality}(n) = n^{o(1)}$ & $Rate \rightarrow 1$

  $\Rightarrow \ell(n) = n^{o(1)}$ meeting Singleton Bound

  $\Rightarrow \ell(n) = n^{o(1)}$ binary, Zyablov bound.

  (locally correcting half-the-distance!)

Main References

- Multiplicity codes [KoppartySarafYekhanin’10]
- See also
  - Lifted Codes [GuoKoppartySudan’13]
  - Expander codes [HemenwayOstrovskyWootters’13]
- Tensor codes [Viderman ‘11] (see also [GKS’13])
- Above + Alon-Luby composition:

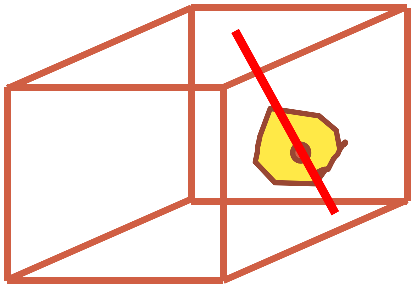
  [KoppartyMeirRon-ZewiSaraf’15]

2

1

3

Lifted Codes



- Codes obtained by inverting decoder:
  - Recall decoder for RM codes.
  - What code does it decode?
  - $C_{m,r,q} = \{f : \mathbb{F}_q^m \to \mathbb{F}_q \,|\, \deg(f|_L) \leq r \;\forall\, L\}$
  - What we know: $\mathrm{RM}[m,r,q] \subseteq C_{m,r,q}$
- Theorem [GKS'13]: $\delta(C_{m,r,q}) \approx \delta(\mathrm{RM}[m,r,q])$

  $Rate(C_{m,r,q}) \to 1$ if $q = 2^t$ and $t \to \infty$
- Local decodability by construction.
- Local testability [KaufmanS'07,GuoHaramatyS'15].

Multiplicity Codes

- Basic example
- Message = (coeffs. of) poly $p \in \mathbb{F}_q[x, y]$.
- Encoding = Evaluations of $\left(p, \frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}\right)$ over $\mathbb{F}_q^2$.

  Length = $n = q^2$; Alphabet = $\mathbb{F}_q^3$; Rate → $\frac{2}{3}$
- Local-decoding via lines. Locality = $O(\sqrt{n})$
- More multiplicities ⇒ Rate $\rightarrow 1$
- More derivatives ⇒ Locality $\rightarrow n^{\epsilon}$

Multiplicity Codes - 2

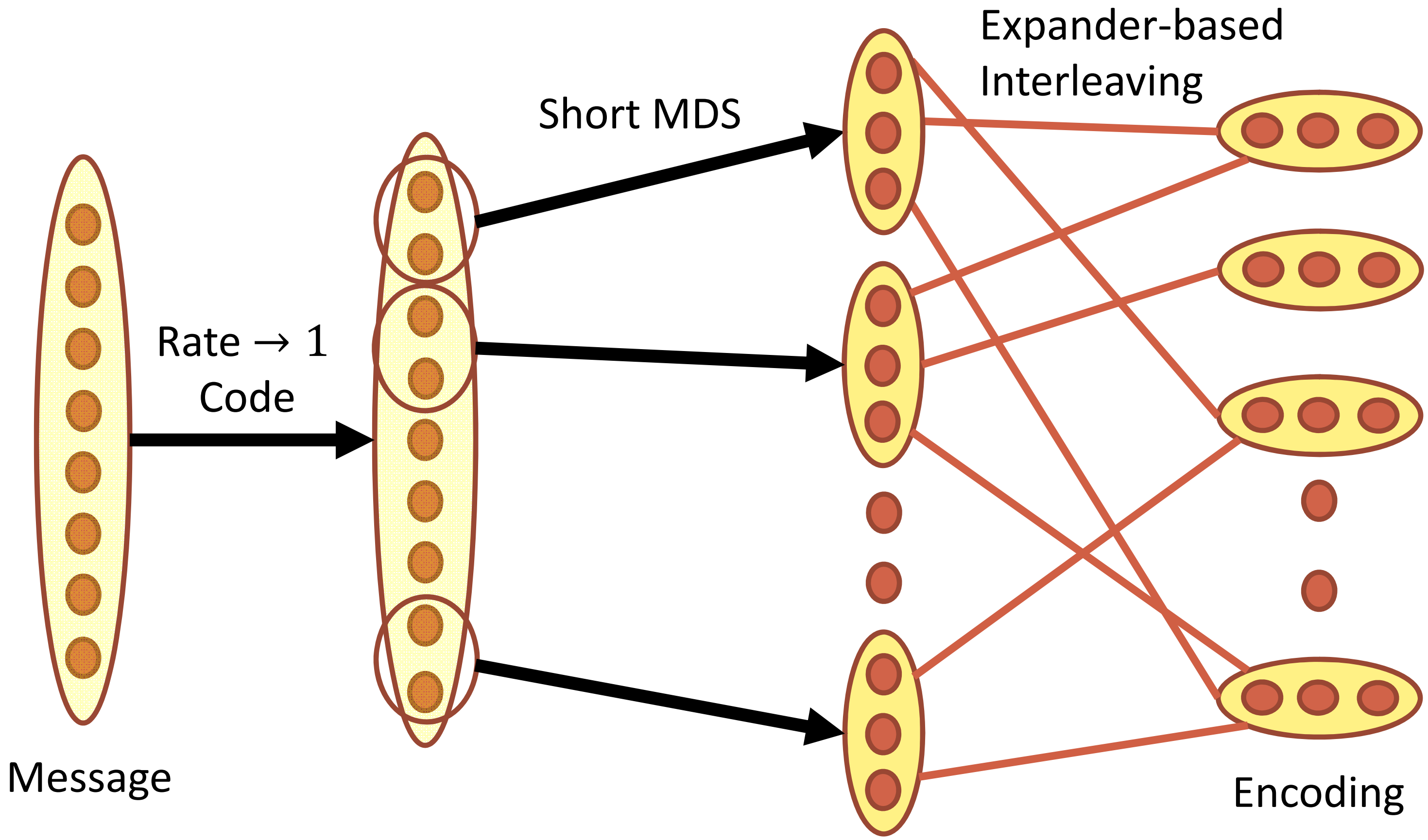
- Why does Rate $\rightarrow \frac{2}{3}$?
- Every zero of $\left(p, \frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}\right) \equiv$ two zeroes of p
- Can afford to use p of degree $\rightarrow 2q$.
- Dimension $\uparrow \times 4$; But encoding length $\uparrow \times 3$
  (Same reason that multiplicity improves radius of list-decoding in [Guruswami,S.])

State-of-the-art as of 2014

- $\forall \epsilon, \alpha > 0 \; \exists \delta = \delta_{\epsilon,\alpha} > 0$ s.t. ∃ codes w.
  - Rate $\geq 1 - \alpha$
  - Distance $\geq \delta$
  - Locality $= n^{\epsilon}$
- Promised:
  - Locality $n^{o(1)}$
  - Singleton bound [What if you need higher distance?]
  - Zyablov bound [What if you want a binary code?]

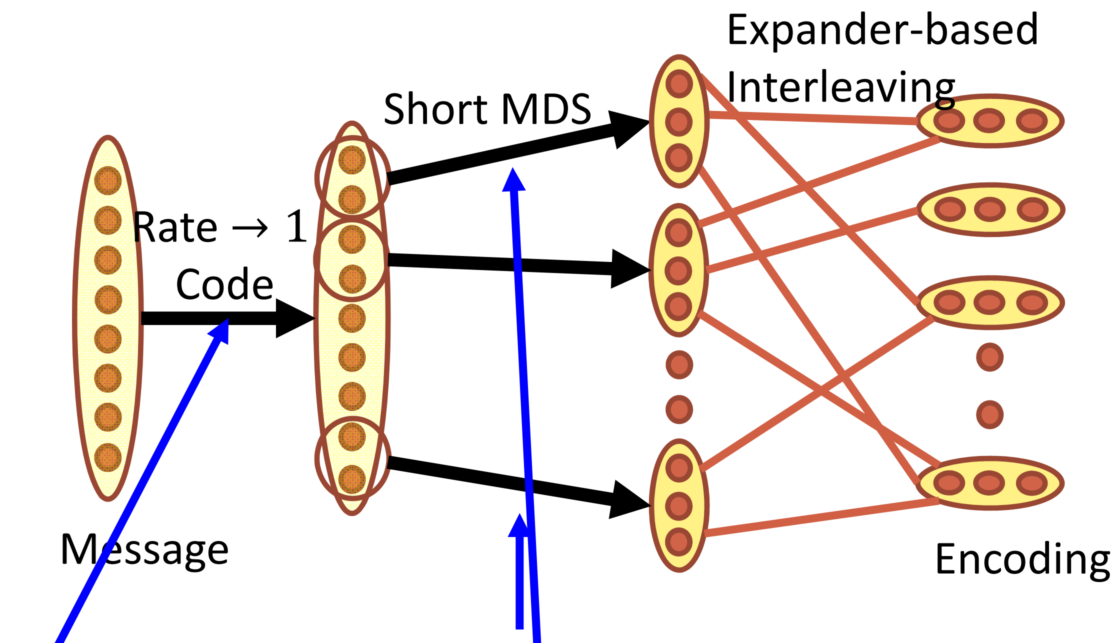
Alon-Luby Transformation

- Key ingredient in [Meir14], [Kopparty et al.'15]



Alon-Luby Transformation

- Key ingredient in [Meir14], [Kopparty et al.'15]



Rate/Distance of final code ~ Rate of MDS

[Meir] Locality ~ Locality of Rate 1 code

Proof = Picture

Subpolynomial Locality

- Apply previous transform, with initial code of Rate $1 - o(1)$ and locality $n^{o(1)}$!
  - [e.g., multiplicity codes with $m = \omega(1)$]
- Singleton bound ✓
- Zyablov bound?
  - Concatenation [Forney'66] ✓

Part 4: Conclusions

The Locality Advantage

- Asymptotically:
  - Achieves best known parameters for explicit codes
  - While achieving significant locality $\ell(n) = 2^{\sqrt{\log n}}$
- Limits?
  - LCCs must satisfy $n = k^{1+\frac{1}{\ell(n)}}$ [Katz-Trevisan]
  - LTCs – no lower bounds known; could match best known 3-LDPC, with $\ell(n) = 3$
  - Linear rate LCC+LTCs with $\ell(n) = \log n$? Open!

Locality in Practice?

- Why don't we see LCCs in practice?
  - Is locality with many errors a natural model?
    - Are LRCs good enough?
    - LCCs allow for lazy recovery (each recovery step local/quick); can prioritize according to needs.
  - Randomized decoding schemes?
  - Moderately big hidden constants
    - More study needed for concrete settings of k, ℓ, δ

Conclusion

- Locality: (moderately) new model
- Remarkable effects possible
- Connect to many other questions in combinatorics/computer science
- Useful as a data storage mechanism?

Thank You