Beer Therapy

- At Oberwolfach in 2003, Ralf Kötter and Madhu Sudan had a week long beer drinking competition.
- Who do you think won?

Ralf

Madhu

Vote early, vote often

Local Algorithms & Error-correction

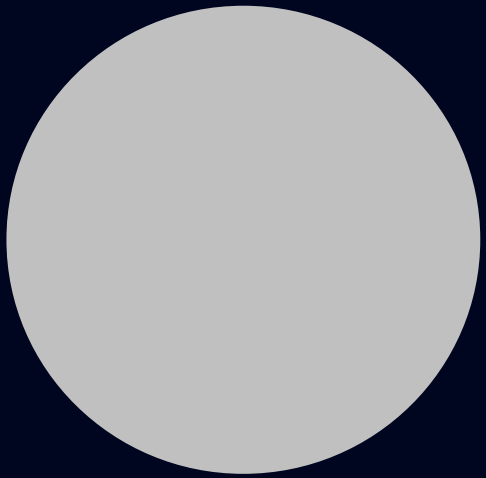
Madhu Sudan
MIT

Beer Therapy

- At Oberwolfach in 2003, Ralf Kötter and Madhu Sudan had a week long beer drinking competition.

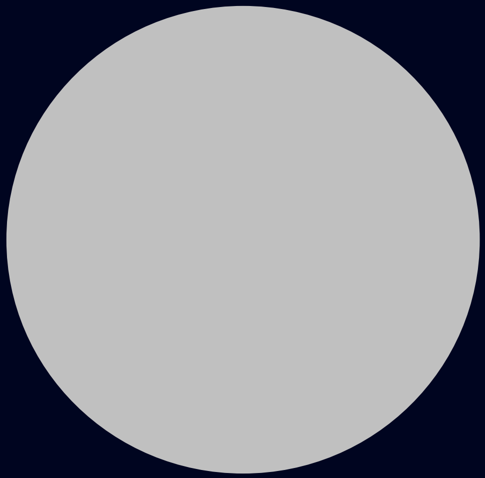
- Who do you think won?

Ralf



Information

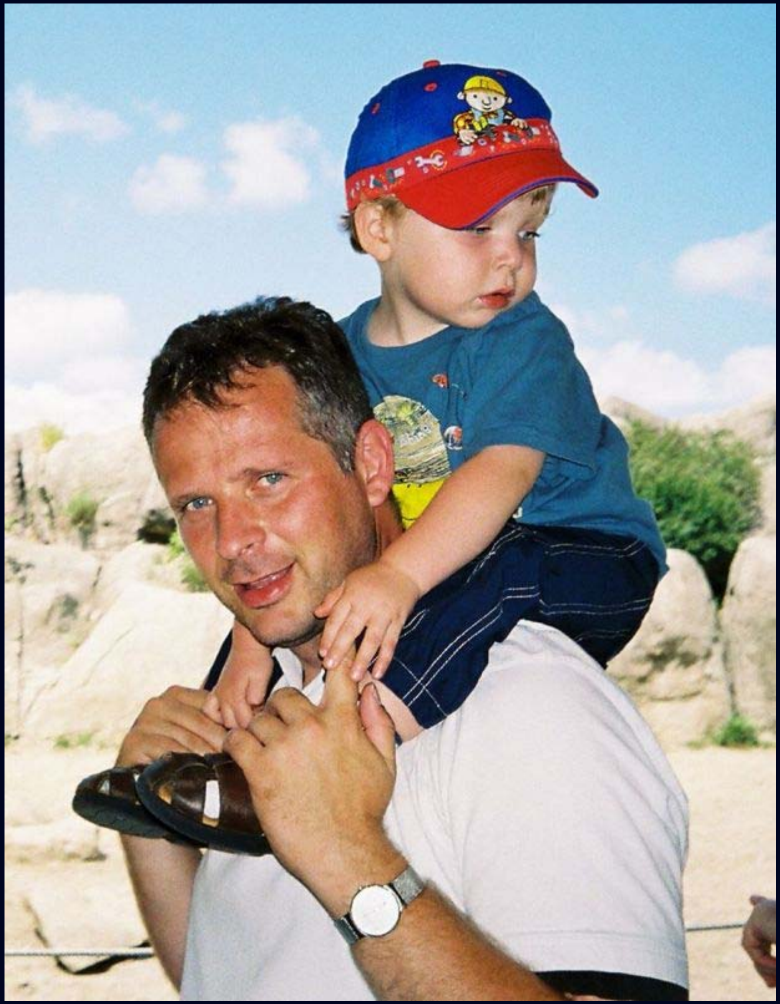
Madhu



Computation

Dedicated to Ralf Kötter

- Dear friend to many …
- Wise beyond his age
- Happy spirit



- … I'll miss him dearly.
- … I already do.

Prelude

- Algorithmic Problems in Coding Theory
- New Paradigm in Algorithms
- The Marriage: Local Error-Detection & Correction

Algorithmic Problems in Coding Theory

- Code: $E : \Sigma^k \to \Sigma^n$; $\text{Image}(E) = C \subseteq \Sigma^n$;
  $R(C) = k/n$, $\delta(C) =$normalized distance.
- Encoding: Fix Code C and associated $E : \Sigma^k \to \Sigma^n$.
  Given $m \in \Sigma^k$, compute $E(m)$.
- Error-detection (ϵ-Testing):

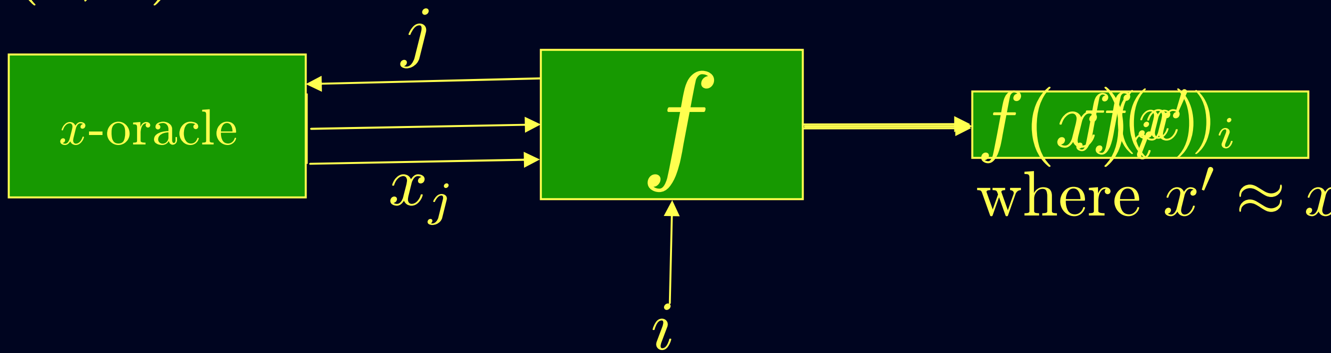
  Given $x \in \Sigma^n$, decide if $\exists m \in \Sigma^k$ s.t. $x = E(m)$.

  Given $x \in \Sigma^n$, decide if $\exists m \in \Sigma^k$ s.t. $\delta(E(m), x) \leq \epsilon$.
- Error-correction (Decoding):

  Given $x \in \Sigma^n$, compute $m \in \Sigma^k$ that minimizes $\delta(E(m), x)$ (provided $\delta(E(m), x) \leq \epsilon$).

Sublinear time algorithmics

- Given $f : \{0,1\}^k \to \{0,1\}^n$ can it be "computed" in $o(k,n)$ time?



- Answer 1: Clearly NO, since that is the time it takes to even read the input/write the output
- Answer 2: YES, if we are willing to:
  1. Present input implicitly (by an oracle).
  2. Represent output implicitly
  3. Compute function on approximation to input.

Extends to computing relations as well.

Sub-linear time algorithms

- Initiated in late eighties in context of
  - Program checking [BlumKannan,BlumLubyRubinfeld]
  - Interactive Proofs/PCPs [BabaiFortnowLund]
- Now successful in many more contexts
  - Property testing/Graph-theoretic algorithms
  - Sorting/Searching
  - Statistics/Entropy computations
  - (High-dim.) Computational geometry
- Many initial results are coding-theoretic!

Sub-linear time algorithms & Coding

- Encoding: Not reasonable to expect in sub-linear time.

- Testing? Decoding? – Can be done in sublinear time.
  - In fact many initial results do so!

- Codes that admit efficient ...
  - ... testing: Locally Testable Codes (LTCs)
  - ... decoding: Locally Decodable Codes (LDCs).

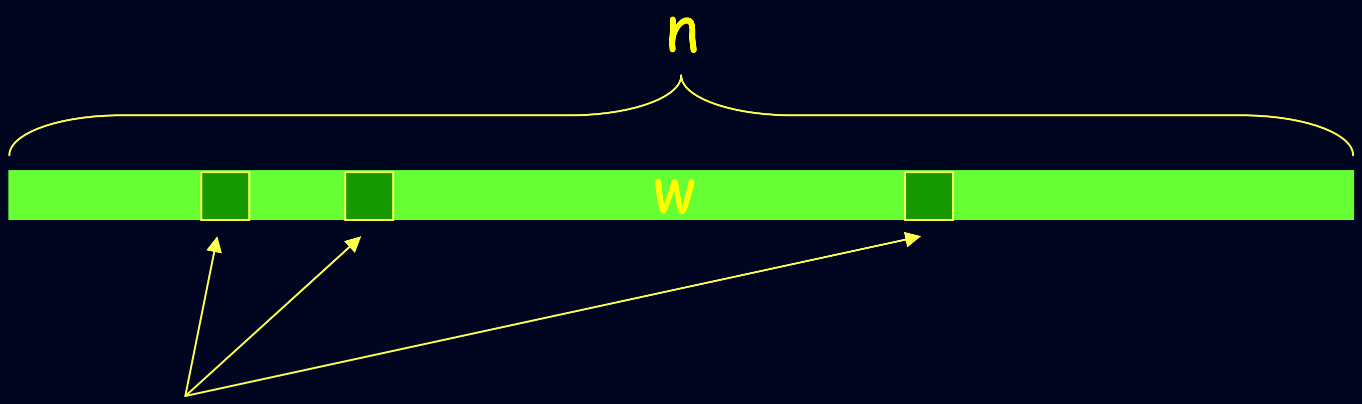
Rest of this talk

- Definitions of LDCs and LTCs
- Quick description of known results
- The first result: Hadamard codes
- Some basic constructions
- Recent constructions of LDCs.
  - [Yekhanin,Raghavendra,Efremenko]

Definitions

Locally Decodable Code

Code: $C : \Sigma^k \to \Sigma^n$ is (q, ϵ)-Locally Decodable
if $\exists$ Decoder D s.t. given $i \in [k]$
and oracle w s.t. $\exists\, m \;\; \delta(w, C(m)) \leq \epsilon \leq \delta(C)/2$,



$D(i)$ reads $q(n)$ random positions of w
and outputs m_i w.p. at least $2/3$.

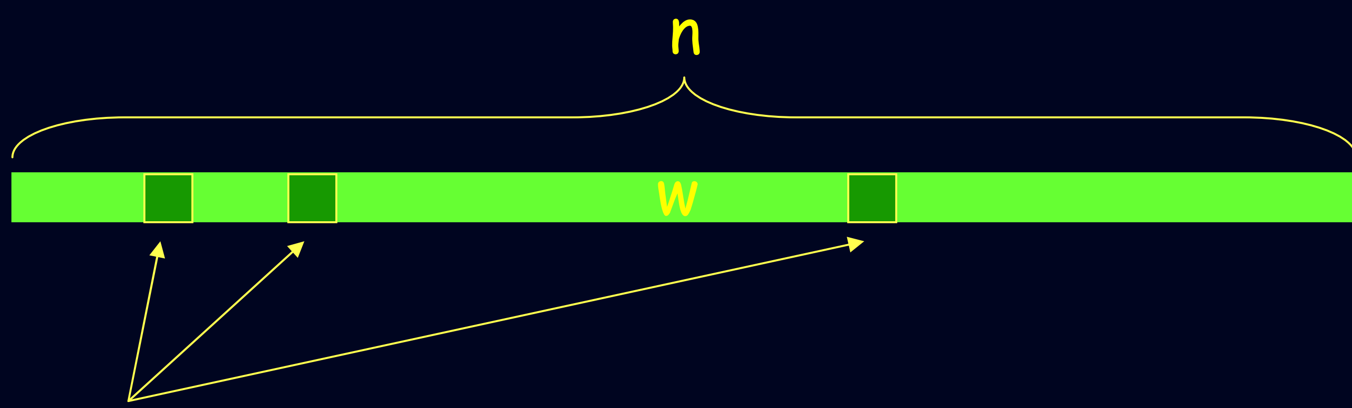
What if $\epsilon > \delta(C)/2$? Might need to
report a list of upto ℓ codewords.

Locally List-Decodable Code

Code: C is (ϵ, ℓ)-list-decodable if $\forall w \in \Sigma^n$,

$\#$ codewords $c \in C$ s.t. $\delta(w, c) \leq \epsilon$ is at most ℓ.

C is (q, ϵ, ℓ)-locally list-decodable if $\exists$ Decoder D s.t.
given $i \in [k]$ and $j \in [\ell]$ and oracle w s.t.
$m_1, \ldots, m_\ell$ are all messages satisfying $\delta(w, C(m_j)) \leq \epsilon$



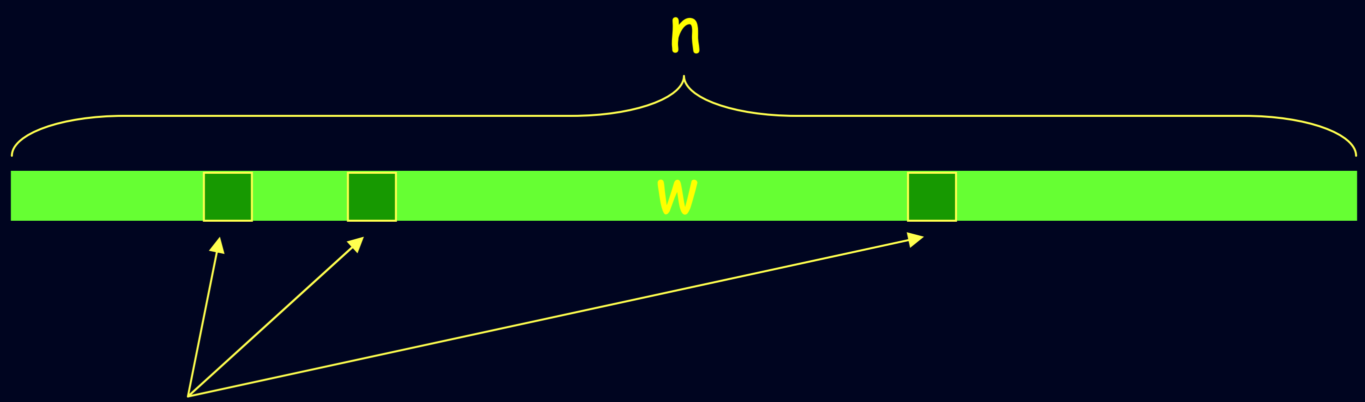
$D(i, j)$ reads $q(n)$ random positions of w
and outputs $(m_j)_i$ w.p. at least $2/3$.

History of definitions

- Constructions predate formal definitions
  - [Goldreich-Levin ’89].
  - [Beaver-Feigenbaum ’90, Lipton ’91].
  - [Blum-Luby-Rubinfeld ’90].
- Hints at definition (in particular, interpretation in the context of error-correcting codes): [Babai-Fortnow-Levin-Szegedy ’91].
- Formal definitions
  - [S.-Trevisan-Vadhan ’99] (local list-decoding).
  - [Katz-Trevisan ’00]

Locally Testable Codes

Code: $C \subseteq \Sigma^n$ is (q, ϵ)-Locally Testable
if $\exists$ Tester T s.t.



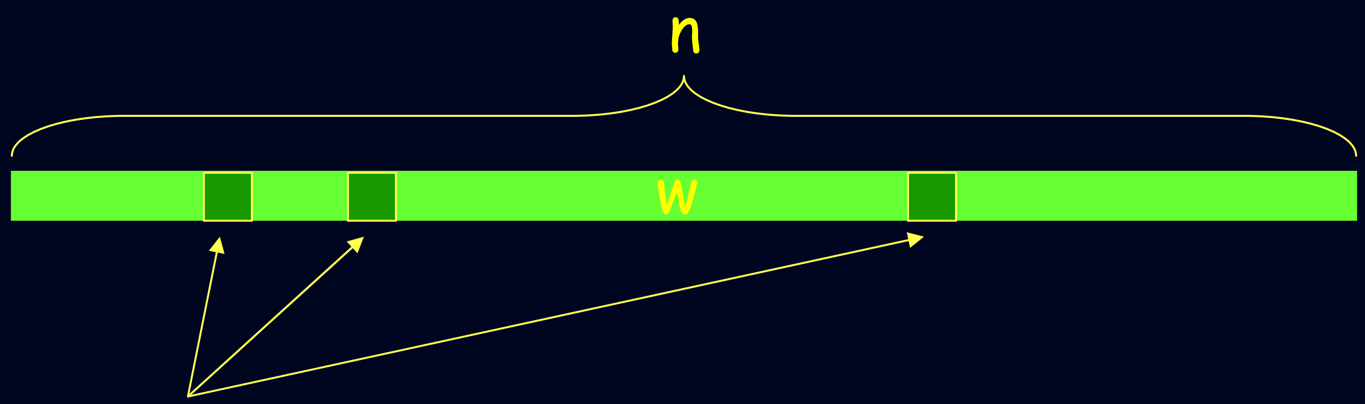
T reads $q(n)$ random positions:

- If $w \in C$ accepts w.p. 1.
- If w is ϵ-far from C, then rejects w.p. $\geq 1/2$.

"Weak" definition: hinted at in [BFLS], explicit in [RS'96, Arora'94, Spielman'94, FS'95].

Strong Locally Testable Codes

Code: $C \subseteq \Sigma^n$ is (q, ϵ)-Locally Testable
if $\exists$ Tester T s.t.



T reads $q(n)$ random positions:

- If $w \in C$ accepts w.p. 1.
- For every $w \in \Sigma^n$,
  T rejects w.p. $\geq \Omega(\delta(w, C))$.

"Strong" Definition: [Goldreich-S. '02]

Motivations

Local decoding: Average-case vs. worst-case

- Suppose $C \subseteq \Sigma^N$ is locally-decodable code for $N = 2^n$. (Further assume can locally decode bits of the codeword, and not just bits of the message.)
- $c \in C$ can be viewed as function $c : \{0,1\}^n \to \Sigma$.
- Local decoding $\approx\Rightarrow$ can compute $c(x)$ for every x, if one can compute $c(x')$ for most x'. Relates average-case complexity to worst-case. [Lipton, STV]
- Alternate interpretation: Compute $c(x)$ without revealing x. Leads to Instance Hiding [BF], Private Information Retrieval [CGKS].

Motivation for Local-testing

- No generic applications known.
- However,
  - Interesting phenomenon on its own.
  - Intangible connection to Probabilistically Checkable Proofs (PCPs).
  - Potentially good approach to understanding limitations of PCPs (though all resulting work has led to improvements).

Contrast between decoding and testing

- Decoding: Property of words near codewords.
- Testing: Property of words far from code.

- Decoding:
  - Motivations happy with n = quasi-poly(k), and q = poly log n.
  - Lower bounds show $q = O(1)$ and n = nearly-linear(k) impossible.
- Testing: Better tradeoffs possible! Likely more useful in practice.
  - Even conceivable: $n = O(k)$ with $q = O(1)$?

Some LDCs and LTCs

Hadamard (1st Order RM) Codes

Message:

(Coefficients of) Linear Functions L from $\mathbb{F}_2^k$ to $\mathbb{F}_2$.

Encoding:

evaluations of L on all of $\mathbb{F}_2^k$.

Parameters:

k bit messages $\to 2^k$-bit codewords

Locality:

$$L(x) = L(x+y) - L(y)$$

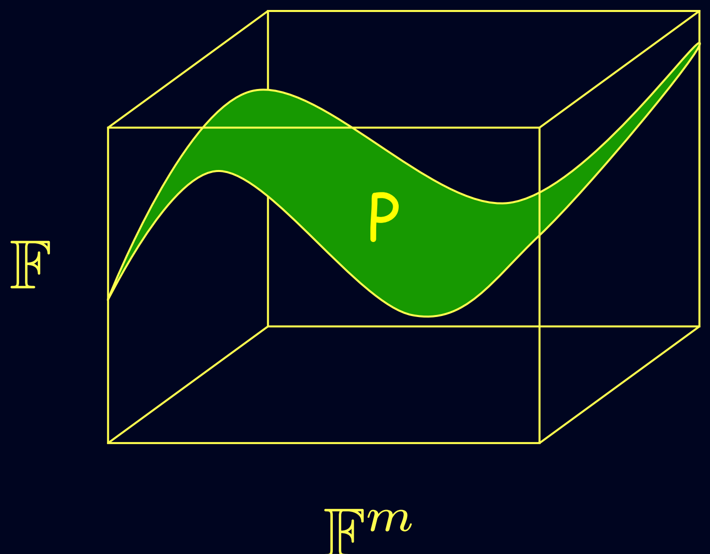
2-Locally Decodable [Folklore/Exercise]
3-Locally Testable [BlumLubyRubinfeld]

Hadamard (1^{st} Order RM) Codes

- Conclusions:
  - There exist infinite families of codes
  - With constant locality (for testing and correcting).

Codes via Multivariate Polynomials

Message: coefficients of deg t, m-variate polynomial P over finite field $\mathbb{F}$



(Reed Muller code)

Encoding: evaluations of P on all of $\mathbb{F}^m$.

Parameters: $k \approx (t/m)^m$, $n = |\mathbb{F}|^m$, $\delta \geq t/|\mathbb{F}|$.

Basic insight to locality

- m-variate polynomial of degree t restricted to $m' < m$-dim. (affine) subspace is polynomial of degree t.
- Local Decoding:

  Pick subspace through point x of interest, and decode on subspace.

  Query complexity $q = |\mathbb{F}|^{m'}$; Time $=$ poly(q).
  $m' \ll m \Rightarrow$ sublinear!
- Local Testing:

  Verify f restricted to space is of degree t.
  Same complexity.

Polynomial Codes

- Many parameters: m, t, $\mathbb{F}$

- Many tradeoffs possible:

  Locality q with $n = \exp(k^{1/(q-1)})$

  Locality $(\log k)^2$ with $n = k^4$

  Locality $\sqrt{k}$ with $n = O(k)$.

Are Polynomial Codes (Roughly) Best?

- No! [Ambainis97] [GoldreichS.00] ...

- No!! [Beimel,Ishai,Kushilevitz,Raymond]

- Really ... Seriously ... No!!!!

[Yekhanin07,Raghavendra08,Efremenko09]

Recent LDCs

[Yekhanin ‘07, Raghavendra ‘08, Efremenko ‘09]

The recent result:

- Fix q = 3; n = ??? (as function of k)
- Till 2007: $n \approx \exp(k^{1/5})$ (non-binary).

$$n \approx \exp(\sqrt{k}) \text{ (binary)}.$$

- [Yekhanin '07]:

$$n \approx \exp(k^{0.0000001}) \text{ (binary)}.$$

- [Raghavendra '08]:
- [Efremenko '09]:

$$n \approx \exp(\exp(\sqrt{\log k})) \text{ (binary)}.$$

Essence of the idea:

- Build “good” combinatorial matrix over $\mathbb{Z}_m$
- Embed $\mathbb{Z}_m$ in multiplicative subgroup of $\mathbb{F}$
- Get locally decodable code over $\mathbb{F}$

“Good” Combinatorial matrix

$$A = \left[\begin{array}{ccccc|c} 0 & \ldots & & & & \\ \ldots & 0 & \ldots & & & \\ \ldots & & 0 & \ldots & & \\ \ldots & & & 0 & \ldots & \\ \ldots & & & & 0 & \end{array}\right]$$

arbitrary

$k \times n$ matrix over $\mathbb{Z}_m$

Zeroes on diagonal

Non-zero off-diagonal

Columns closed under addition

Embedding into field

- Let $A = [a_{ij}]$ be “good“ over $\mathbb{Z}_m$
- Let $\omega =$ primitive mth root of unity in $\mathbb{F}$.
- Let $G = [\omega^{a_{ij}}]$.

Theorem [Yekhanin,Raghavendra,Efremenko]:
G generates m query LDC over $\mathbb{F}$.

Highly non-intuitive!

Improvements

- $A = [a_{ij}]$; $G = [\omega^{a_{ij}}]$.
- Off-diagonal entries of A from S
  $\Rightarrow G$ generates $|S|+1$-query LDC.
  (Suffices for [Efremenko])
- ω^S zeroes of t-sparse polynomial over $\mathbb{F}$
  $\Rightarrow G$ generates t-query LDC.
  (Critical to [Yekhanin])

"Good" Matrices?

- [Yekhanin]:
  - Picked m prime.
  - Hand-constructed matrix.
  - Achieved $n = \exp(k^{1/|S|})$
  - Optimal if m prime!
  - Managed to make S large with t=3.
- [Efremenko]
  - m composite!
  - Achieves $|S| = 3$ and $n = \exp(\exp(\sqrt{\log k}))$ ([Beigel,Barrington,Rudich];[Grolmusz])
  - Optimal?

Limits to Local Decodability: Katz-Trevisan

- q queries $\Rightarrow n = k^{1+\Omega(1/q)}$.
- Technique:
  - Recall D(x) computes C(x) whp for all x.
  - Can assume (with some modifications) that query pattern uniform for any fixed x.
  - Can find many random strings such that their query sets are disjoint.
  - In such case, random subset of n^{1-1/q} coordinates of codeword contain at least one query set, for most x.
  - Yields desired bound.

Some general results

- Sparse, High-Distance Codes:
  - Are Locally Decodable and Testable
    - [KaufmanLitsyn, KaufmanS]
- 2-transitive codes of small dual-distance:
  - Are Locally Decodable
    - [Alon,Kaufman,Krivelevich,Litsyn,Ron]
- Linear-invariant codes of small dual-distance:
  - Are also Locally Testable
    - [KaufmanS]

Summary

- Local algorithms in error-detection/correction lead to interesting new questions.

- Non-trivial progress so far.

- Limits largely unknown
  - O(1)-query LDCs must have Rate(C) = 0
    - [Katz-Trevisan]

Questions

- Can LTC replace RS (on your hard disks)?
  - Is a significant rate-loss necessary?
  - Lower bounds?
  - Better error models?

- Simple/General near optimal constructions?
- Other applications to mathematics/computation? (PCPs necessary/sufficient)?
- Lower bounds for LDCs?/Better constructions?

Thank You!