Local Error-Detection and Error-correction

Madhu Sudan
MIT

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):

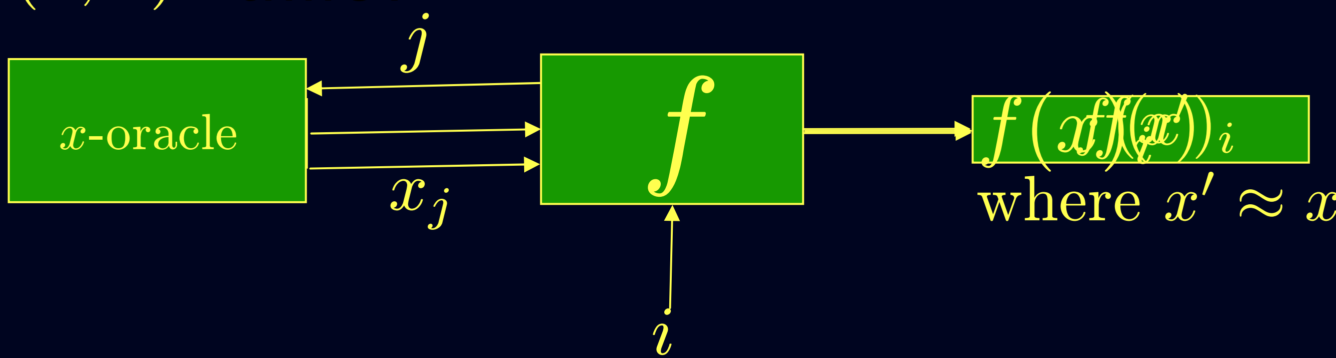
  Given $x \in \Sigma^n$, decide if $\exists m \in \Sigma^k$ s.t. $x = E(m)$.

  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
  - Interactive Proofs/PCPs
- 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).

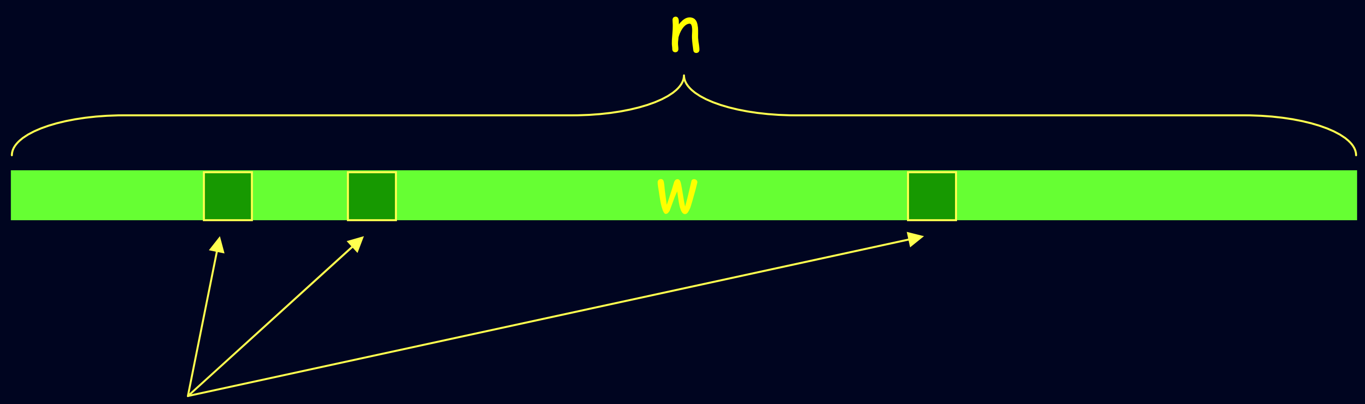
Rest of this talk

- Definitions of LDCs and LTCs
- Quick description of known results
- Some basic constructions
- (Time permitting) Yekhanin's construction of LDCs.

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

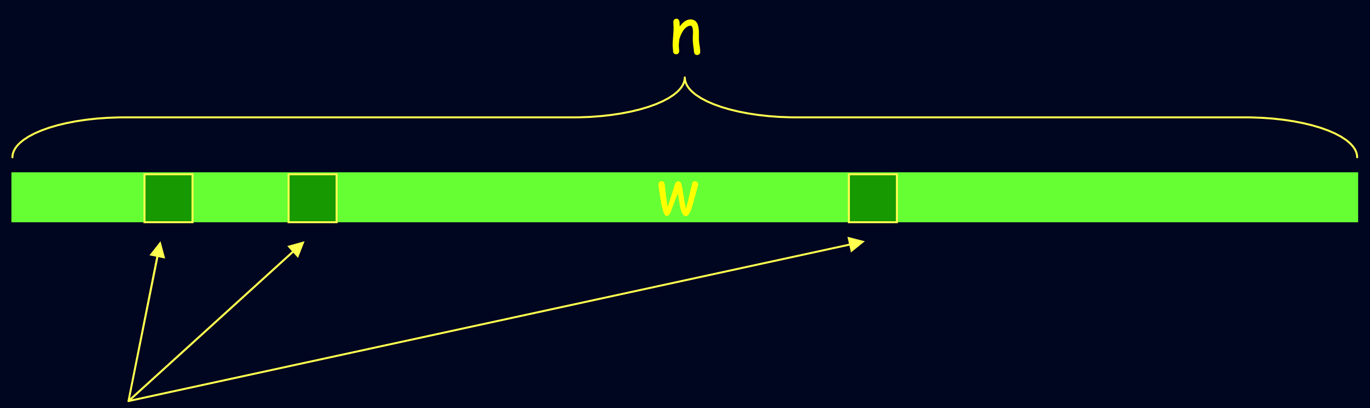
Code: C is $\underline{(\epsilon,\ell)\text{-list-decodable}}$ if $\forall w \in \Sigma^n$,

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

C is $\underline{(q,\epsilon,\ell)\text{-locally}}$ 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 $\delta(w, C(m_j)) \le \epsilon$



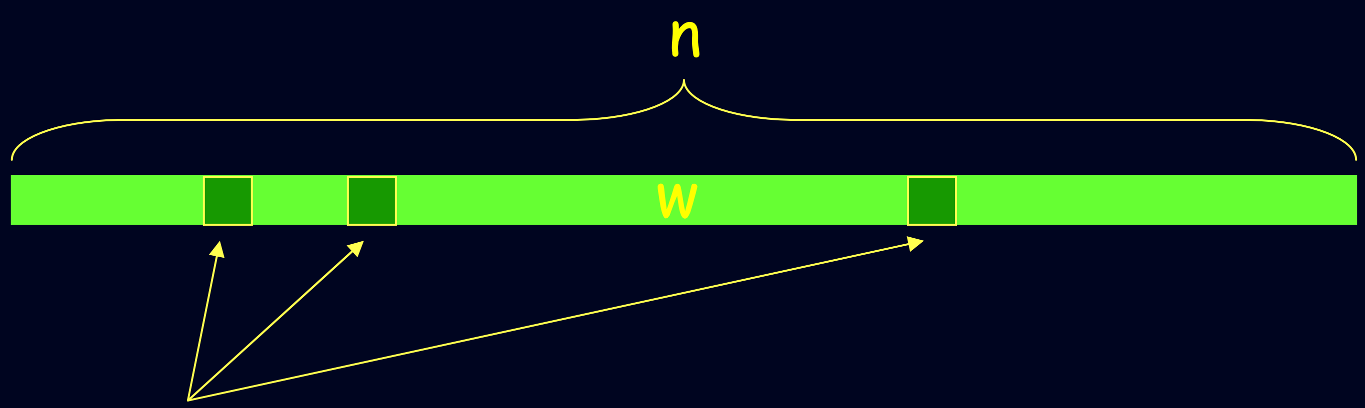
$D(i,j)$ reads $q(n)$ random positions of w
and outputs $(m_j)_i$ w.p. at least $2/3$.

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.



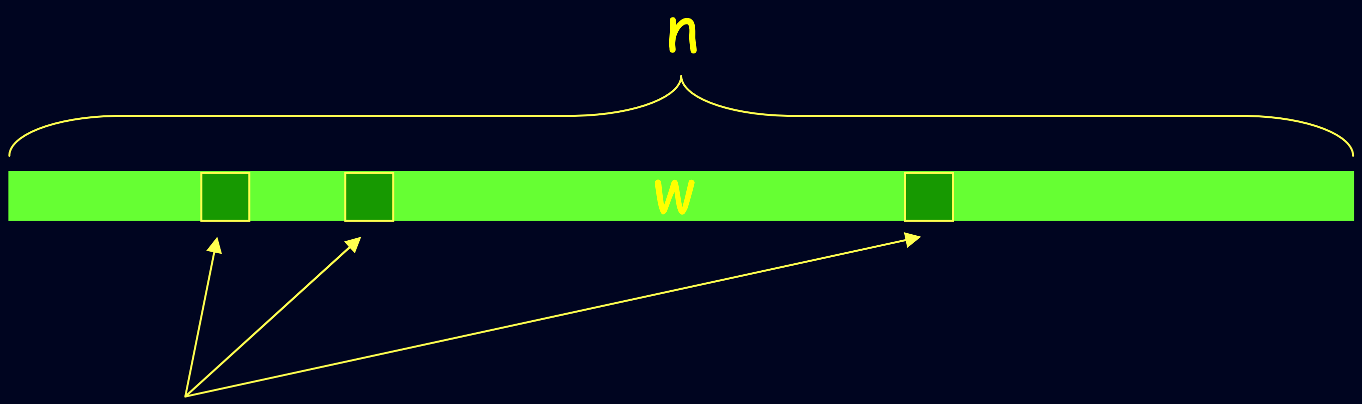
T reads $q(n)$ random positions:

- 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

Motivations for Local decoding

• 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

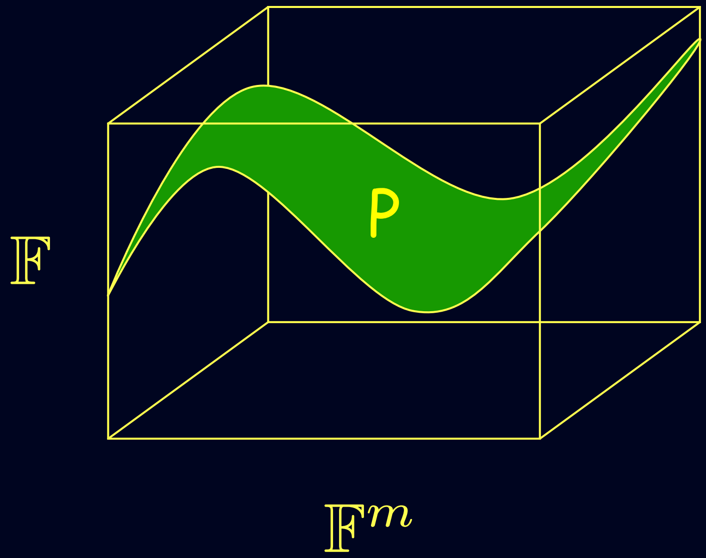
- Decoding: Property of words near codewords.
- Testing: Property of words far from code.

- 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

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.

Summary of Constructions

- Polynomial Codes: (Locally decodable and testable)

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

- Polynomial Codes + Composition/Concatenation:

  $$\text{Local Testability with}$$
  $$q = O(1) \text{ and } n = \tilde{O}(k) \quad = k \cdot (\log k)^c.$$
  $$\text{Local Decodability with } n = \exp(k^{1/\mathrm{poly}(q)})$$

- Codes based on “Algebraic Designs” [Yekhanin]

  $$\text{Local Decodability with } q = 3 \text{ and } n = \exp(k^{\epsilon})$$

[Yekhanin '07]'s LDCs

Recall: Combinatorial Designs

- Families of Sets: $S_1, \ldots, S_k, T_1, \ldots, T_k$.
  $S_i, T_i \subseteq \{1, \ldots, m\}$.
- Restrictions on Intersections:
  - E.g.,

i vs. i: $|S_i \cap T_i|$ even. (Large) (Small)

i vs. j: $|S_i \cap T_j|$ odd. (Small) (Large)

- Basic Question:

How large can k be?

(As a function of m?)

Typical answer $k = \Theta(m)$

[Yekhanin]'s Algebraic Designs

- Families of Vectors: $u_1, \ldots, u_k,\ v_1, \ldots, v_k.$
  $u_i, v_i \in \mathbb{F}_p^m.$
  p small prime
- Restrictions on Inner Products:

$$\langle u_i, v_i \rangle = 0 \qquad \langle u_i, v_i \rangle = 0$$

$$\langle u_i, v_j \rangle \neq 0 \qquad \langle u_i, v_j \rangle \in S \not\ni 0$$

Basic p-design $\qquad$ (p, S)-design

- Basic Question: How large can k be?

$\binom{m}{p-1} \sim m^{p-1}$

At most $m^{|S|}$!
Can we achieve it?

[Yekhanin]'s Algebraic Designs

- Families of Vectors: $u_1, \ldots, u_k, v_1, \ldots, v_k$.
  $u_i, v_i \in \mathbb{F}_p^m$.
  p small prime
- Restrictions on Inner Products:

$\langle u_i, v_i \rangle = 0$ | $\langle u_i, v_i \rangle = 0$

$\langle u_i, v_j \rangle \neq 0$ | $\langle u_i, v_j \rangle \in S \not\ni 0$

Basic p-design | (p, S)-design

- Basic Question: How large can k be?

$\binom{m}{p-1} \sim m^{p-1}$

At most $m^{|S|}$!
Can we achieve it?

[Y'07] Algebraic designs and LDCs

Lemma 1: Basic p-design with k vectors in $\mathbb{F}_p^m$
$\Rightarrow$ p-query (binary) LDCs mapping k-bits to p^m bits

$$k = m^{p-1} \quad \Rightarrow \quad n = \exp(k^{1/p-1})$$

(Matches some of the early constructions)

Lemma 2: $\exists q = q(p, S) \leq p$ s.t.
(p, S)-design with k vectors in $\mathbb{F}_p^m$
$\Rightarrow$ q-query LDCs mapping k bits to p^m bits.

$q(p, S)$ - Algebraic niceness of $S \subseteq \mathbb{F}_p^*$.

[Y'07] Algebraic designs and LDCs

Lemma 2: $\exists q = q(p, S) \leq p$ s.t.

(p, S)-design with k vectors in $\mathbb{F}_p^m$

$\Rightarrow$ q-query LDCs mapping k bits to p^m bits.

$q(p, S)$ - algebraic niceness of $S \subseteq \mathbb{F}_p^*$.

Definition: S is q-algebraically nice if

$\exists$ a q-sparse polynomial $h(x) \in \mathbb{F}_2[x]/(x^p - 1)$ s.t.

ideal generated by $\{h(x^\beta) | \beta \in S\}$ is non-trivial.

(One of two equivalent definitions)

[Y'07] Algebraic designs and LDCs

Lemma 2: $\exists q = q(p, S) \leq p$ s.t.

(p, S)-design with k vectors in $\mathbb{F}_p^m$

$\Rightarrow$ q-query LDCs mapping k bits to p^m bits.

$q(p, S)$ - algebraic niceness of $S \subseteq \mathbb{F}_p^*$.

Example: $p = 127$; $S = \{1, 2, 4, 8, 16, 32, 64\}$

S is 3-algebraically nice

m^7 long (p, S)-designs exist!

$\Rightarrow$ 3-query LDC mapping k bits to $\exp(k^{1/7})$ bits

[Y'07] Algebraic designs and LDCs

Lemma 2: $\exists q = q(p, S) \leq p$ s.t.
(p, S)-design with k vectors in $\mathbb{F}_p^m$
$\Rightarrow$ q-query LDCs mapping k bits to p^m bits.

$q(p, S)$ - algebraic niceness of $S \subseteq \mathbb{F}_p^*$.

Lemma 3: $p = 2^t - 1 \Rightarrow S = \{1, 2, 4, \ldots, 2^{t-1}\}$
is 3-algebraically nice.

Lemma 4: S multiplicative subgroup of $\mathbb{F}_p$
$\Rightarrow \exists$ (p, S)-design of length $\sim m^{|S|}$.

Theorem: $\exists$ 3-query LDC
mapping k bits to $\exp(k^{0.0000001})$ bits.

Proofs?

- Disclaimer: Proof of Lemma 2, Lemma 3 too long to fit here. (Many context switches, but elementary.)

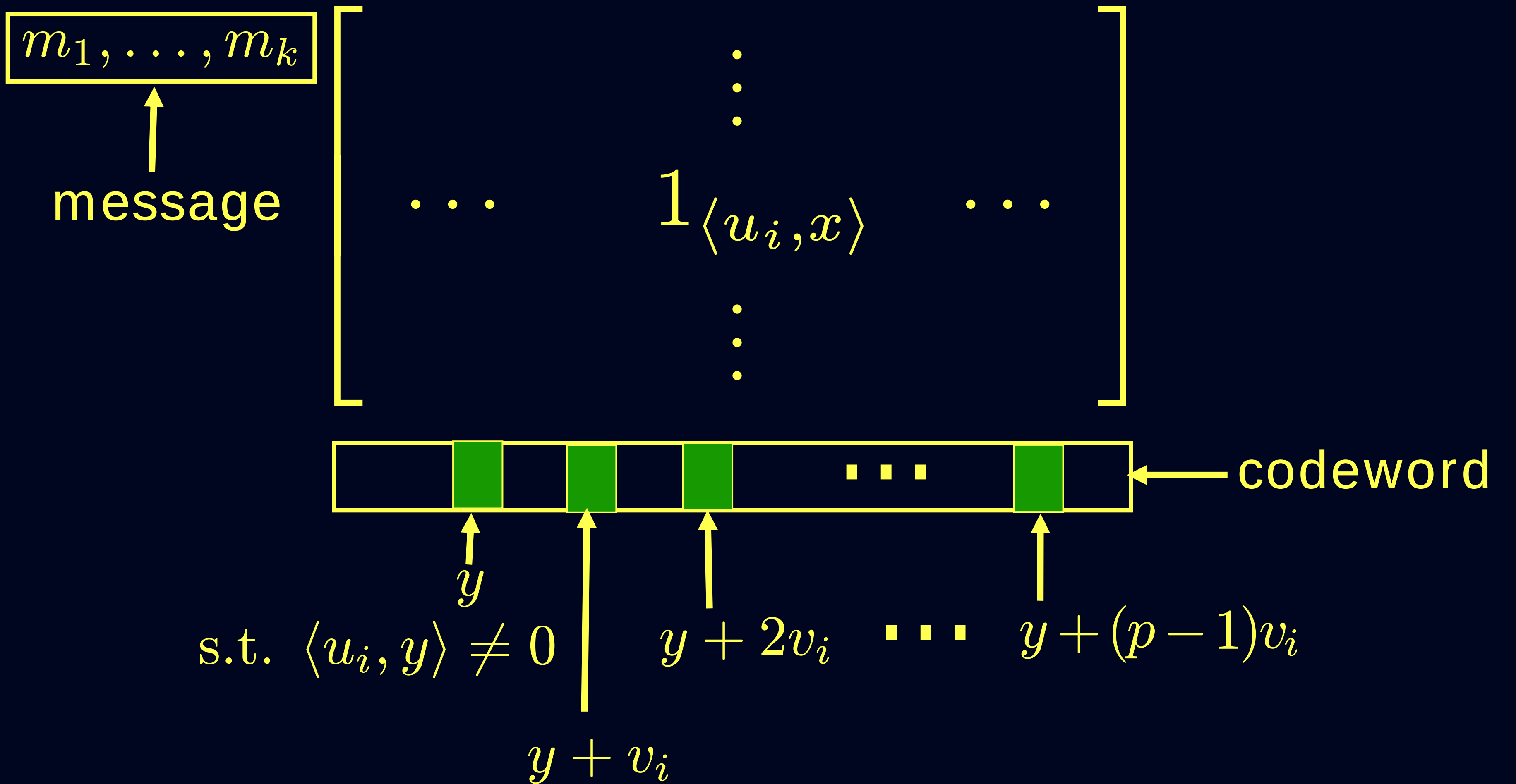
- Will only attempt to show Lemmas 1 and 4.

Basic designs and LDCs

Given $u_1, \ldots, u_k; v_1, \ldots, v_k$

$$G = \begin{array}{cc} \begin{bmatrix} & \vdots & \\ \cdots & 1_{\langle u_i, x \rangle} & \cdots \\ & \vdots & \end{bmatrix} & u_i \\ x & \end{array}$$

Basic designs and LDCs

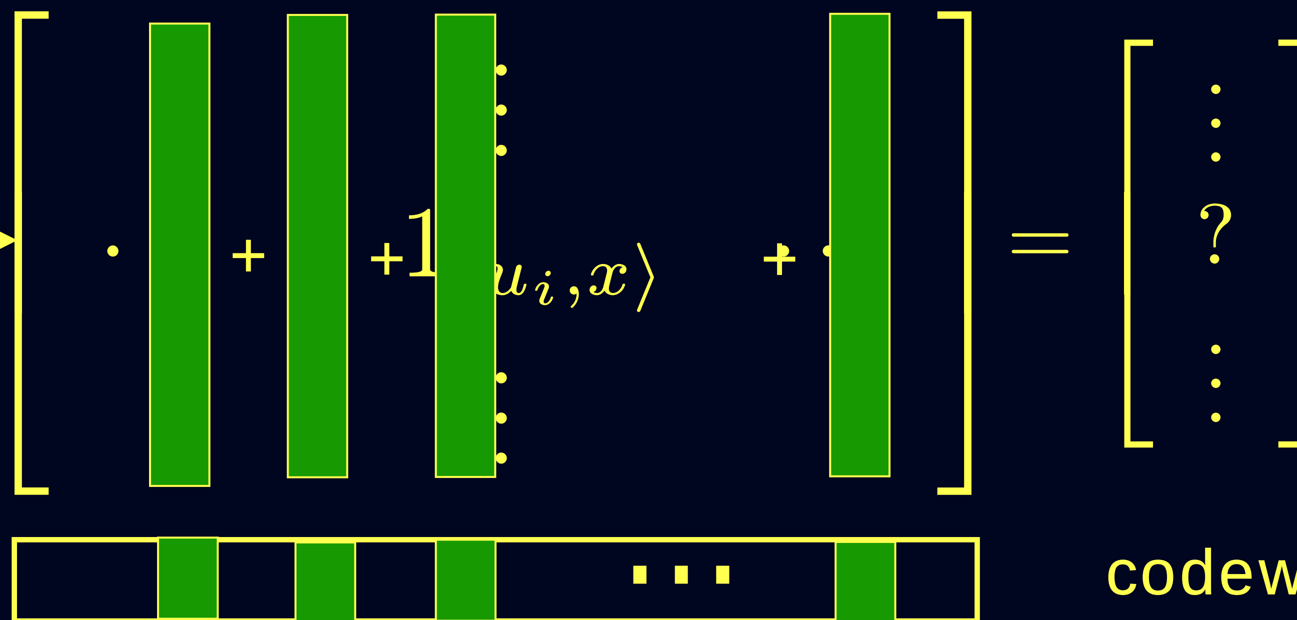


Report parity of ■ locations

Basic designs and LDCs

ith row
- all ones.

other rows
- $p-1$ ones.



codeword

Proof of Lemma 4

- Construction of Basic p-designs:

$$i \leftrightarrow \text{set of size exactly } p-1$$

$$u_i = \text{characteristic vector of set } i.$$

$$v_i = \text{characteristic vector of complement of set } i.$$

$$\langle u_i, v_i \rangle = 0; \quad \langle u_i, v_j \rangle = |i \cap j| \in \{1, \ldots, p-1\}$$

- Construction of (p,S)-designs for S multiplicative:

Take u_i, v_i as above;

Use $\tilde{u}_i, \tilde{v}_i = p/|S|$th tensor powers of u_i, v_i.

Conclusions

- 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 R(C) = 0 [Katz-Trevisan]