Symmetric LDPC codes are not necessarily locally testable

Eli Ben-Sasson, Ghid Maatouk, Amir Shpilka, Madhu Sudan

Algebraic Coding Theory Workshop
EPFL 08.09.2011
Locally testable codes (LTC)

- Linear code $C$, subspace of $\mathbb{F}^N$.
- Given $f \in \mathbb{F}^N$, can determine if $f \in C$ with a constant number of queries.
- Wlog, $C$ is $k$-testable if there exists a canonical $(k, \delta, s)$-tester for $C$. Symmetric LDPC codes are not necessarily locally testable.
A canonical \((k, \delta, s)\)-tester for a linear code \(C\) is specified by a distribution \(\mu\) over \(C_{\leq k}^\perp\).

operates by sampling \(u\) according to \(\mu\) and accepting \(f\) if and only if \(\langle u, f \rangle = 0\), where \(\langle u, f \rangle = \sum_{i=1}^{n} u(i)f(i)\).
Canonical \((k, \delta, s)\)-tester

- comes with a guarantee to reject words which are \(\delta\)-far from \(C\) with probability at least \(s\).
- Think of \(\delta, s\) as being 0.1.
LTCs are LDPC

- If $C$ has a $k$-local tester, then $C^\perp$ is spanned by words of weight $\leq k$. 

Intuition: random LDPC codes lack "structure."
LTCs are LDPC

- If $C$ has a $k$-local tester, then $C^\perp$ is spanned by words of weight $\leq k$.
- If $C$ has a $k$-local tester, $C$ is a $k$-LDPC code.
LTCs are LDPC

- If $C$ has a $k$-local tester, then $C^\perp$ is spanned by words of weight $\leq k$.
- If $C$ has a $k$-local tester, $C$ is a $k$-LDPC code.
- Converse: does $k$-LDPC imply $k$-testability?
LTCs are LDPC

- If $C$ has a $k$-local tester, then $C^\perp$ is spanned by words of weight $\leq k$.
- If $C$ has a $k$-local tester, $C$ is a $k$-LDPC code.
- Converse: does $k$-LDPC imply $k$-testability?
  
  [BHR05] Not for random LDPC codes!
LTCs are LDPC

- If $C$ has a $k$-local tester, then $C^\perp$ is spanned by words of weight $\leq k$.
- If $C$ has a $k$-local tester, $C$ is a $k$-LDPC code.
- Converse: does $k$-LDPC imply $k$-testability?
  
  [BHR05] Not for random LDPC codes!
- Intuition: random LDPC codes lack “structure”.

Symmetric LDPC codes are not necessarily locally testable
Codes with symmetries

- Let $\pi$ be a permutation of $\{1, \ldots, N\}$
- $C \circ \pi = \{(x_{\pi(1)}, \ldots, x_{\pi(N)}) \mid (x_1, \ldots, x_N) \in C\}$
- $C$ invariant under $\pi$ if $C = C \circ \pi$
- Automorphism group of $C$ is $\{\pi \mid C = C \circ \pi\}$.
Examples of automorphism groups

- Symmetric group $S_N$: too restrictive
Examples of automorphism groups

- **Symmetric group** $S_N$: too restrictive

- **Affine group**:

  $\text{Affine}_\mathbb{K} = \{ \pi_{a \neq 0, b} : x \mapsto ax + b \}$

- Index coordinates $\{1, \ldots, N\}$ by elements of a field $\mathbb{K}$
Examples of automorphism groups

- Symmetric group $S_N$: too restrictive
- **Affine group:**

$$\text{Affine}_K = \{ \pi_{a \neq 0, b} : x \mapsto ax + b \}$$

- Index coordinates $\{1, \ldots, N\}$ by elements of a field $K$
- Example of affine-invariant codes: Reed-Muller codes
  - $RM[m, d]_K$ is set of evaluations of $m$-variate polynomials of degree $d$ over $K$
  - invariant under affine transformations $\bar{x} \mapsto A \cdot \bar{x} + \bar{b}$. 

Symmetric LDPC codes are not necessarily locally testable
Conjecture [AKKLR05]

- For general $\mathcal{C}$, automorphism group is trivial (only identity permutation)

Symmetric LDPC codes are not necessarily locally testable
Conjecture [AKKLR05]

- For general $C$, automorphism group is trivial (only identity permutation)
- Large/rich automorphism group gives hope for testability
Conjecture [AKKLR05]

- For general $C$, automorphism group is trivial (only identity permutation)
- Large/rich automorphism group gives hope for testability
- **Conjecture** [AKKLR05]: If $C$ has a 2-transitive automorphism group and a low-weight dual codeword, $C$ is LTC
Conjecture [AKKLR05]

- For general $C$, automorphism group is trivial (only identity permutation)
- Large/rich automorphism group gives hope for testability
- **Conjecture** [AKKLR05]: If $C$ has a 2-transitive automorphism group and a low-weight dual codeword, $C$ is LTC
- 2-transitivity: lots of (weight-preserving) permutations in automorphism group. Hopefully, $C$ is LDPC..
For general $\mathcal{C}$, automorphism group is trivial (only identity permutation)

Large/rich automorphism group gives hope for testability

**Conjecture** [AKKLR05]: If $\mathcal{C}$ has a 2-transitive automorphism group and a low-weight dual codeword, $\mathcal{C}$ is LTC

2-transitivity: lots of (weight-preserving) permutations in automorphism group. Hopefully, $\mathcal{C}$ is LDPC..

Conjecture refuted in [GKS08]: there exist such codes which are not even LDPC.

Symmetric LDPC codes are not necessarily locally testable
Conjecture (revisited)

- **Conjecture** [AKKLR05]: If $\mathcal{C}$ is LDPC and has a 2-transitive automorphism group, then $\mathcal{C}$ is LTC.

Symmetric LDPC codes are not necessarily locally testable.
**Conjecture (revisited)**

- **Conjecture [AKKLR05]**: If $\mathcal{C}$ is LDPC and has a 2-transitive automorphism group, then $\mathcal{C}$ is LTC
- **Known LTC are symmetric LDPC**
  - RM codes
  - “Sparse” affine-invariant codes ($O(|\mathbb{K}|^\ell)$ codewords) [GKS09]
  - “Single-orbit” affine-invariant codes [KS05]
    (dual is generated by all affine permutations of a low-weight codeword)
**Conjecture (revisited)**

- **Conjecture [AKKLR05]:** If $C$ is LDPC and has a 2-transitive automorphism group, then $C$ is LTC
- Known LTC are symmetric LDPC
  - RM codes
  - “Sparse” affine-invariant codes ($O(|K|^\ell)$ codewords) [GKS09]
  - “Single-orbit” affine-invariant codes [KS05]
    (dual is generated by all affine permutations of a low-weight codeword)
- **Our result:** the conjecture is false. There exists an infinite family of 2-transitive LDPC codes which is not testable with a constant number of queries.
Theorem (Symmetric LDPC codes are not necessarily LTC)

For every prime $p$ there exists a positive integer $k$ and an infinite family of positive integers $N$ such that for every $n \in N$ the following holds:

- There is an affine-invariant code $C^{(n)} \leq \mathbb{F}_p^n$.
- $C^{(n)}$ is a $k$-LDPC code.
- $C^{(n)}$ is not $o(\log n / \log \log n)$-locally testable.
Constructing a symmetric LDPC non-LTC

- Known symmetric (affine-invariant) LDPC codes:
  - $RM[m, d]_K$
  - Sparse affine-invariant codes
Constructing a symmetric LDPC non-LTC

- Known symmetric (affine-invariant) LDPC codes:
  - $RM[m, d]_K$
  - Sparse affine-invariant codes
- Problem: these families are also LTC!

Symmetric LDPC codes are not necessarily locally testable
Constructing a symmetric LDPC non-LTC

- Known symmetric (affine-invariant) LDPC codes:
  - $RM[m, d]_k$
  - Sparse affine-invariant codes
- Problem: these families are also LTC!
- Second attempt: use them as building blocks.
Constructing a symmetric LDPC non-LTC

- Known symmetric (affine-invariant) LDPC codes:
  - $RM[m, d]_K$
  - Sparse affine-invariant codes
- Problem: these families are also LTC!
- Second attempt: use them as building blocks.
  - Taking sums ($C = C_1 + C_2$): preserves symmetry and LDPC, but also LTC [BGMSS11]
Constructing a symmetric LDPC non-LTC

- Known symmetric (affine-invariant) LDPC codes:
  - $RM[m, d]_K$
  - Sparse affine-invariant codes
- Problem: these families are also LTC!
- Second attempt: use them as building blocks.
  - Taking sums ($\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2$): preserves symmetry and LDPC, but also LTC [BGMSS11]
  - Taking intersections ($\mathcal{C} = \mathcal{C}_1 \cap \mathcal{C}_2$):
    - Intersection of $k$-LDPC codes is also $k$-LDPC
    - Intersection of $k$-LTC codes not necessarily $k$-LTC!

Symmetric LDPC codes are not necessarily locally testable
Constructing a symmetric LDPC non-LTC

- Known symmetric (affine-invariant) LDPC codes:
  - $RM[m, d]_K$
  - Sparse affine-invariant codes
- Problem: these families are also LTC!
- Second attempt: use them as building blocks.
  - Taking sums ($C = C_1 + C_2$): preserves symmetry and LDPC, but also LTC [BGMSS11]
  - Taking intersections ($C = C_1 \cap C_2$):
    - Intersection of $k$-LDPC codes is also $k$-LDPC
    - Intersection of $k$-LTC codes not necessarily $k$-LTC!
    - **BUT** Simply taking intersection of sparse/RM codes will not work.

Symmetric LDPC codes are not necessarily locally testable
Lifting affine-invariant codes

- Defining a new operation on affine-invariant codes
- Let $F \leq \mathbb{L}$ and $C \leq F^{\mathbb{L}}$
- Recall: we view coordinates of codeword as field elements
- View elements of $C$ as functions from $\mathbb{L}$ to $F$: $C \subseteq \{\mathbb{L} \rightarrow F\}$.
Now let $F \leq L \leq K$ and $C \subseteq \{L \rightarrow F\}$

Want to “lift” $C$ to a code $C' \subseteq \{K \rightarrow F\}$

$Lift(C) = \{f : K \rightarrow F \mid \text{Trace}(f) \in C\}$

Recall

$$\text{Trace}_{L,K}(x) : K \rightarrow L$$

$$x \mapsto \sum_{i=0}^{[K:L]-1} x^{p^i}$$

Any test for $C$ can be viewed as a test for $\text{Lift}(C)$. 
The idea

- Work on an extension field $\mathbb{K} := \mathbb{F}_{p^n}$ with many subfields
- $n := p_1 \cdots p_\ell$, $\ell = \Omega(\log n / \log \log n)$
- $\mathbb{L}_i := \mathbb{F}_{p^{p_i}}$
- Start with (sparse) symmetric LDPC testable codes $\tilde{C}_i \subseteq \{\mathbb{L}_i \to \mathbb{F}_p\}$
- “Lift” the $\tilde{C}_i$ to codes $C_i \subseteq \{\mathbb{K} \to \mathbb{F}_p\}$
  - $C_i$ no longer sparse, but still LDPC and testable
- Intersection of $C_i$’s is our code
  - Keep the LDPC property (easy), lose the testability (harder).
$\mathcal{C}$ is not locally testable

- Proof relies on analysis of structure of symmetric sparse codes (developed in [KS05], [GKS08], [GKS09], [BS10], [KL10],...)
- Constraints lead to the construction of certain rectangular matrices over finite field, whose kernels we analyze
- Combine with [BHR05] strategy for proving non-testability: defining a distribution on faraway words that fools canonical tests.
Thank you!
Questions?