Today: Tanner Codes

- Sipser-Spielman Decoding

Recall LDPC Codes

Bipartite Graph → Code

$G = (L, R, E) \rightarrow C_6$

$n$ left vertices → block length $n$

$m$ right vertices → $k \geq n - m$

Every right vertex $v$ to a constraint $e_v$

Assignment $x_1, \ldots, x_n$ to variable (left vertex)

must satisfy $\sum_{u \in v} x_u = 0$. $v \subseteq v$
- \((c, d)\) bounded: left degree \(\leq c\)
  right degree \(\leq d\)

- \((\gamma, \delta)\) - expander: \(\forall S \subseteq L \text{ s.t. } |S| \leq \delta n\)
  \(|\Gamma(S)| \geq \gamma \cdot |S|\)

- \((\tilde{\gamma}, \delta)\) - unique expander: \(\forall S \subseteq L \text{ s.t. } |S| \leq \delta n\)
  \(|\Gamma^+(S)| \geq \tilde{\gamma} \cdot |S|\)

  - \(\Gamma(S) = \{ v \mid \exists u \in S \text{ s.t. } u \leftrightarrow v \}\)
  - \(\Gamma^+(S) = \{ v \mid \exists ! u \in S \text{ s.t. } u \leftrightarrow v \}\)

**Lemma:** \((c, d)\) bounded, \((\gamma, \delta)\) expander is also a \((2\delta - c, \delta)\) - unique expander

**Theorem:** \(G\) is \((c, d)\) bounded, \((\gamma, \delta)\) expander 
& \(2\delta > c \implies C_6\) has rel. dist. \(\delta\).
Decoding Algorithm

1. Starting with assignment $x_1, \ldots, x_n$ to variables.
2. Iteratively maintain assignment $y_1, \ldots, y_n$.
   (initially $\bar{y} \leftarrow \bar{x}$)
3. In Iteration $i$:
   - Constraint $v$ is sat. if $\bigcap_{u \in \mathcal{C}_{vu}} x_u = 0$
     and unsat. o.w.
   - If $\exists u$ with more unsat. neighbors than sat. ones then $y_u \leftarrow \bar{y}_u$
   (else STOP; output $\bar{y}$)
Analysis

Assume $\gamma > \frac{3}{4} \cdot c$

Claim 1: Algorithm terminates in $m$ iterations

Claim 1": If $\# \text{ errors} = c$ then algorithm terminates in $< c \cdot c$ iterations.

Proof: Consider a vertex about to be flipped.
- Flipping toggles sat vs. unsat status of neighbor of \( u \). Status of all other constraints unchanged.

- Conclude: Total \# unsat. vertices decreases in each iteration (since \( u \) has more unsat. ngbrs than sat. ones).

- Initially \# unsat. ngbrs. \( \leq m \) \( \Rightarrow \) Claim 1

\[ \leq C \cdot e \]

\[ \uparrow \]

must be neighbor of some corrupted bit.

\[ \Rightarrow \) Claim 1'
- So we know algorithm terminates quickly.
- But does it terminate in right codeword?

**Claim 2:** if $e = \Delta(\bar{c}, \bar{x}) < \frac{8n}{c+1}$
then alg. terminates with all constraints sat.

**Claim 2.1:** if $e = \Delta(\bar{c}, \bar{x}) < \frac{8n}{c+1}$
then throughout $\Delta(c, y) < (c+1) \cdot e$

**Proof of Claim 2.1:** In each iteration $\Delta(c, y)$ increases by at most 1. Initially $\Delta(c, y) = \Delta(c, x) = e$. # Iterations is at most $c \cdot e$ (Claim 1').

\[\]
Proof of Claim 2: At beginning of iteration $i$ (in particular, in final iteration) ...

Let $S = \bigcup_{u \in V} y_u = c_v \emptyset = \emptyset$

\[ u = \Gamma^+(S) = \text{unique \ ngbrs.} \]

\[ \text{for contradiction} \]

$|S| \leq (C+1)e < 5n$

$\Rightarrow |\Gamma^+(S)| \geq (2\gamma - \epsilon). |S|$

$\geq \frac{\epsilon}{2}. |S|$
But this implies some vertex $v \in S$ must have \( \geq \frac{\varepsilon}{2} \) neighbours in $\Pi^+(S)$.

For this vertex $v$ every neighbour in $\Pi^+(S)$ is unsat and there are more than \# other vertices.

Conclude: \( |S| \neq \emptyset \Rightarrow \exists \) flippeable vertex $v$.

Claim 3: If $\Delta(v, x) = e < \frac{8n}{c+1}$ then algorithm terminates with $c$.

Proof: Else alg outputs $\overline{c} \in C_n$ with $\Delta(c, \overline{c}) \leq (c+1)e < 8n$ \( \uparrow \) Contradiction.
What kind of expanders exist?

Any $c, d = O(1) ; \ n \to \infty$

$s \to c \ e \ \delta < \frac{1}{c} \cdot \frac{m}{n}$

achievable, existentially ... but what about constructive stuff?
History

[Gabber, Galil] - first constructive results
\[ \alpha > 0, \quad \frac{\alpha}{c} < 1 \]

[Margulis]

[Jansen '84]

[Lubotsky, Phillips, Sarnak] \[ \frac{\alpha}{c} \rightarrow \frac{1}{2} \]

[Margulis] \[ \uparrow \]

just short of being useful!

(What can you do with above?)

[2001] [Capalbo, Reingold, Vaclav, Wigderson]: Can get \( \frac{\alpha}{c} \rightarrow 1 \).
[Tanner]: Recursive construction: $G \oplus C_{\text{small}}$

graph with min code ordering on edges

$(x_{i_1}, x_{i_2}, \ldots, x_{i_d}) \in C_{\text{small}}$

Variables Constraint

Lemma: $C_{\text{small}} = [\mathbb{d}, 2, \Delta] - \text{code}$

$G$ is $(c, d)$ regular, $(r, d)$-expander

Then $C = G \oplus C_{\text{small}}$ has rate $R \geq 1 - \frac{c(d-r)}{d}$

and $S$, provided $r \geq \frac{c}{\Delta}$.
Proof: \# linear constraints \leq (d-l) \cdot m

\Rightarrow \text{rate} \geq \frac{n - (d-l)m}{n}

= n - \frac{(d-l)cn}{c^l}

= n \left(1 - \frac{c(d-l)}{d}\right)

Distance: Usual arguments (see notes of last lecture).

Issue: Can we \(C_{\text{small}}\) of large \(\Delta\) but this increases \(d-l\) which reduces rate. Will this technique ever be successful in
Abstraction of Expander technology in 90's

Specify \( \frac{\gamma}{c} < \frac{1}{2} \); Can find \( c \) s.t.

\[ \forall d \exists g, n_0 s.t. \forall n \geq n_0 \]

\( n \) vertex expanders could be constructed.

\[ \ldots \] Now can verify that good codes can be constructed this way.

Decoding = ?

- Usual decoding doesn't seem to work
- But parallelized variant does.
Parallel-FLIP (parameter: $t < \Delta$)

In iteration $i$:

- All constraints that are within distance $t$ from codeword of $C_{\text{Small}}$
  Send FLIP message to neighbors in ERROR

- In parallel all variables that receive FLIP messages flip their assignment

Stop when all constraints satisfied.

Analysis: Will argue that in each iteration
  # bits in error, $\Delta(y, c)$, goes down
  (by constant fraction).
Let \( S = \text{variables in error} = \{ v \mid c_v + y_v \} \)

\[ U = \{ v \mid \# \{ u \in S \mid v \rightarrow u \} \geq \{ 1, \ldots, \ell \} \} \]

\[ T = \Pi(S) - U \]

\( F = \text{variables that receive FLIP message} \)
Claim: Constraint in $U$ sends FLIP message to (and only to) neighbors in $S$.

Proof: Note $U$ vertices are within distance $t$ of $C_{small}$ & errors are from vertices in $S$. 

Lower bound on $|F \setminus S|$:

$$|F \setminus S| \geq \frac{|U|}{c} \geq \frac{1}{c} \frac{1}{t-1} (t \cdot r - c) \cdot |S|$$

Upper bound on $|F - S|$:

$$|F - S| \leq \left( \frac{c \cdot |S|}{\Delta - t} \right) \cdot t$$

Vertex must have $\Delta - t$ neighbours in $S$ to send wrong FLIP messages.
Which is greater?

\[
\frac{1}{c} \cdot \frac{1}{t-\gamma} \cdot (t \cdot \gamma - c) \quad \text{vs.} \quad \frac{c \cdot t}{\Delta - t}
\]

**Setup:** Fix \( \gamma \); \( c \);

Let \( t \to \infty \); Fix \( t \);

Let \( \Delta \to \infty \); Fix \( \Delta \);

Let \( d \to \infty \); (So code has i\textsuperscript{ve} rate)

Then

\[
\frac{1}{c} \cdot \frac{1}{t-\gamma} \cdot (t \cdot \gamma - c)
\]

\[
\to \frac{1}{t-1} \left( t \cdot \frac{\gamma}{c} - 1 \right) \to \frac{\gamma}{c} > 0
\]

While

\[
\frac{c \cdot t}{\Delta - t} \to \frac{c \cdot t}{\Delta} \to 0
\]
Conclusions

- Can design codes of $R > 0$
  - Meeting $p > 0$ fraction of error in linear time

- But $R$ vs. $p$ relationship not great!
  - (much worse than algebraic constructions)

- Some extremal settings.
  - Can let $R \to 1$ with $p > 0$
  - Can't see $p \to \frac{1}{2}$ with $R > 0$
    - (needs fast decoding...)