Today: Linear-Time Encodable & Decodable Codes

Recall: LDPC (Expander) Codes

yield $R > 0$ & correct $p > 0$ fraction of error in linear time (worst-case)

But what about the “easier” task of encoding. Now it is slower... or is it?
Hope 1: Maybe code defined by LDPC matrix also has an LDG matrix? (Low Density Generator)

Unfortunately .... This is impossible.

Claim: LDG matrix leads to low Distance Code.

Proof: Some bits of message affect few bits of codeword. Flipping such a bit leads to codewords that are close to each other. B

So ... Hope 1 is shattered.
[Spielman]:

- Use expander graph to "encode" information anyway.

\[ c_j = \bigoplus m_i \quad i \leftrightarrow j \]

- Claim: Given \((m', c)\) s.t. \(m' \approx m\),

\((m, c)\) legal codeword of above code.

Flip finds \(m\).
- **Conclude**: Need to only "protect" c from errors.

- But c is constant factor smaller than m. Smaller problem \(\Rightarrow\) Can resume?

- Or... Not so simple.

  # message bit has gone down. 😊

  But # errors we are protecting from is still same 😞

- Need to be a bit more careful.
[Spielman]: Two uses of LDG code.

\[ m \xrightarrow{E} (m, c) \]

1. Given \((m', c)\) s.t. \(\exists m \approx m'\)
   \[ E(m) = (m, c), \text{ can compute } m. \]
   [if check bits are all correct, can recover message]

2. Given \((m', c')\) s.t. \(\exists m \text{ s.t. } E(m) = (m, c) \approx m', c \approx c'\)
   Can find \(m''\) s.t. for \(d < 1\),
   \[ \Delta(m, m'') = d \cdot \Delta(c, c') \]
   [if # check bit errors is small then
   # message bit errors can be reduced]

Led Spielman to call these ERROR-REDUCTION CODES.
**Lemma:** (ERROR-REDUCERS)

\[ \exists \rho > 0 \text{ s.t. } \forall k \exists \text{ Error-Reduction code } R_k : \Sigma_0^1 \Sigma_k \rightarrow \Sigma_0^1 \Sigma_k^{k/2} \text{ a decoder } \text{FLIP}_k \]

s.t. the following holds:

Let \( c = R_k(m) \) & let \( (m', c') \)

be s.t. \( S((m, c), (m', c')) \leq \rho \)

Then \( S(\text{FLIP}_k(m', c'), m) \leq \frac{1}{6} \cdot S(c, c') \)

\[ \leq \frac{\rho}{2} \]

Will defer proof of this lemma.

Will first see why it suffices.
Theorem: \( \exists p > 0 \text{ s.t. } \forall k, \exists E_k, D_k \) (linear time computable) s.t.

\[
E_k : \{0,1\}^k \rightarrow \{0,1\}^{3k}
\]

\[ m \mapsto c \]

\[
D_k : \{0,1\}^k \rightarrow \{0,1\}^k
\]

\[ (m', c') \mapsto m \]

s.t. if \( S((m', c'), (m, E(m))) \leq p \)

then \( D_k (m', c') = m \)
Construction of $E_R$

$E_R: m \mapsto (c, y, z)$

where

\[ c = R_k(m); \]
\[ y = E_{k/2}(c); \]
\[ z = R_{2k}(c, y); \]
Analysis: (Code parameters, Encoding time)

1. Clearly $k \rightarrow 4k$

   message + check.

2. Let running time of $R_k$ be $< c \cdot k$

   Then running time of $E_k$ is given by

   $T(k) \leq c \cdot k + c \cdot 2k + T(k/2)$

   $= 6ck = O(k).$

   (Six times slower ... )
Decorder $D_k (m', c', y', z')$

1. Use $Flip_{2k} (c', y', z')$ to get $(c'', y'')$ (with hopefully less error)

2. Use $D_{k12} (c'', y'')$ to get $c^{(3)}$  
   (hopefully $c^{(3)} = c$)

3. Use $Flip (m', c^{(3)})$ to get $m''$  
   (hopefully $m'' = m$)

Running Time: Linear
Analysis of Correctness of Dewuder:

(for \( p = ? \))

- Good to switch to absolute error model.
- Let \( e = \Delta((m', c', y', z'), (m, c, y, z)) \)
- Note \( e \leq p \cdot (4k) \)

- We have \( \Delta((c', y', z'), (c, y, z)) \leq e \)

Assume \( \frac{e}{3k} \leq p \iff p \cdot \frac{q}{3} \leq p \)

\( \Rightarrow \) (using the lemma) \( \text{FLIP}_{2k} \) returns

\[ \Delta((c'', y''), (c, y)) \leq \frac{e \cdot (2k)}{3k} \cdot \frac{2}{2} \]

\[ \leq \frac{e}{3} \quad \text{(Step 0)} \]

(error has reduced ... but is this good enough?)
- Now for step 2 we have

\[ S((c^o, y^o), (c, y)) \leq \frac{\epsilon}{3} \leq \frac{4pk}{3} \]

\[ < p \cdot (2k) \]

So

\[ D_{\frac{k}{2}}(c^o, y^o) = c^{(3)} = c ! \]

- Finally for step 2 we have

\[ \Delta((m', c^{(3)}), (m, c)) \]

\[ = \Delta((m', c), (m, c)) \]

\[ = \Delta(m', m) \leq \epsilon \]
if $c < p \left( \frac{3k}{2} \right)$ \[\leq p \cdot 4k \leq p \cdot \frac{3k}{2} \]
\[\leq p \leq \frac{3}{8} p \]

**Lemma**
\[\Rightarrow \delta(m'', m) \leq \frac{1}{6} \delta(c^0, c) = 0\]
\[\Rightarrow m'' = m \text{ as desired.}\]

So theorem holds for $p = \frac{3}{8} p$. □
ERROR-REDUCTION CODES

- Pick $(c, 2c)$-bounded $(\gamma, \delta)$-expander with $\gamma > \frac{7}{8}c$

- Encode using the graph: $E_k(m) = c$ where

$$C_i = \bigoplus_{j \rightarrow i} m_j$$

- Decode using FLIP where $j^{th}$ constraint is SAT if $C_j = \bigoplus_{i \rightarrow j} y_i$ and $y_i, \ldots, y_k$ is current message; iterate till every message bit is adjacent to more SAT constraints than UNSAT.
Analysis

- Let \( \tau = \Delta((m', c'), (m, \mathcal{E}(m))) \)

- Initial \# UNSAT constraints \( \leq C \cdot \tau \)

- So alg. terminates in \( C \cdot \tau \) iterations.

- Total \# message errors \( \Delta(y, m) \) is always less than \( (C+1) \cdot \tau \)

- Stopping condition = ?
- \textbf{Note} \quad \Gamma^+(S) - E \subseteq \text{UNSAT} \subseteq \Gamma(S) \cup E

- \quad |\Gamma^+(S)| \geq (2\gamma - c) \cdot |S|

- \quad |\text{UNSAT}| \geq (2\gamma - c) \cdot |S| - 2|E|

- \quad |\text{UNSAT} \cap \Gamma(S)| \geq (2\gamma - c) |S| - 2|E|

- \quad \text{if} \quad \frac{|\text{UNSAT} \cap \Gamma(S)|}{|S|} > \frac{c}{2} \quad \text{then not done.}

- \text{Stop \quad if} \quad 2\gamma - c - 2 \frac{|E|}{|S|} < \frac{c}{2}

\Rightarrow \quad \frac{2|E|}{|S|} \geq 2\gamma - \frac{3c}{2} \geq \frac{c}{4}

\Rightarrow \quad |S| \leq \frac{8}{c} |E|

Pick \quad c \quad \text{large enough to make this work!}