ST08 LECTURE 17

Today: Graph-Based Codes
- Distance + Rate
- Decoding

Review:

Part I: ExistentiaL & Universal bounds on codes
(Random, Packing, Orthogonality ...)

Part II: Algebraic Codes: Meet many limitation bounds for some settings of parameters

Part III: Algebraic Decoding: Large (Optimal?) fraction of errors, worst-case.
Why study yet more codes?

- Can ask for faster decoding
  - linear time?
  - sublinear time?
  - concrete parameters... not "pure" asymptotics?


Approach: use codes based on graphs.
History:

1. [Gallager '63]: Introduced codes based on graphs as a means of getting good binary codes, with efficient decoding.

   *Constructions:* @ Random procedure to construct codes

   *Decoding Algorithm* (seemingly efficient)

2. [Tanner '84]: Richer class of graph theoretic codes.

3. [Sipser Spielman '94]: Provably efficient Algorithm! (Worst-case errors)
Motivation:

- Codewords of linear code recognized by parity check matrix \( \mathbf{H} \)

\[
\begin{bmatrix}
x_1 & \cdots & x_n
\end{bmatrix}
\begin{bmatrix}
h_{i1} \\
\vdots \\
h_{im}
\end{bmatrix}
\leq 0
\]

\[h_1 & \cdots & h_m \quad m = n-k\]

- Can it also help recover from error?

Say \( (x, \mathbf{H})_j \neq 0 \), i.e., \( \langle x, h_j \rangle \neq 0 \)

\[\Rightarrow \exists \text{ error in one of the coordinates where } h_j \text{ is non-zero.}\]

- Is this (algorithmically) useful?
For random matrix $H$, this is virtually useless. Half the coordinates are "flagged" by $h_j$. Correlation with error $\rightarrow 0$.

But if $H$ is sparse then

$$\Gamma(j) = \{i | h_{ij} \neq 0\}$$

is small $\varepsilon$ saying one of those bit is in error could be quite useful.

Motivates LDPC = Low Density Parity Check codes. (\ =~ $H$ is sparse)
Graph Theoretic View

- $H$: Adjacency matrix of bipartite graph
- Rows of $H =$ Variables - left vertex
- Columns of $H =$ Constraints - right vertex
- Edges $i \leftrightarrow j \iff h_{ij} = 1$.

- Boolean assignment to variables forms codeword iff all constraints satisfied

- Constraint $j$ satisfied if assignment to neighbors has parity 0.
Example

\[ \begin{array}{cccccc}
0 & 0 & 0 & x_1 & & h_1^+ \\
1 & 1 & 0 & x_2 & & h_1^+ \\
0 & 1 & 0 & h_2^+ & & h_2^+ \\
1 & 0 & 0 & & & \\
1 & 0 & 0 & & & \\
1 & 1 & 0 & & & h_m^+ \\
0 & 1 & 0 & & & \\
0 & 1 & 0 & x_n & & h_{m-1}^+ \\
0 & 0 & 0 & & & \\
\end{array} \]

\[ \uparrow \quad \text{codeword} \]

\[ \uparrow \quad \text{non-codeword} \]
Construction: Given: \( n, k, + \) parameter \( c \)

- Let \( m = n-k \) \& \( d = \frac{cn}{m} \)

(assume \( d \) is integer)

- Pick random permutation \( \sigma: [cn] \rightarrow [cn] \)

\[ h_{i,j} = 1 \quad \text{iff} \quad \sigma(c(i-1) + l) \quad \leq \quad \sum_{i=1}^{j} d(i-1) + \ldots dij \quad \text{for some} \quad l \in [c] \]
Theorem: Code reaches $61V$ bound as $L \to \infty$.

Algorithm (Vaguely): Iterate in rounds ....

2nd Round:

- Start with prob. estimates $P_i$.
  - $P_i$ is the prob. the $i^{th}$ variable is 1.

- Constraints compute prob. they are satisfied.
  - Variables update their prob. $P_i$ to improve prob. constraints are satisfied.

Final assertion: Algorithm converges with random errors.

Precise theorem?
Tanner: Use more general constraints!

E.g.

in this order variables should be from [7, 4, 3] Hamming Code.

More general? Not really, since this is still an LDPC code....

But gives better idea how to construct such codes....
[Tanner]: Distance lower bounded by "girth" of underlying graph, leads to concrete results.

("girth" = length of shortest cycle in graph)

[Sipser-Spielman]: Distance lower bounded by "expansion" of underlying graph.

Expansion can also be used to prove performance under worst case errors.
Basic Graph Theoretic Definitions

- **Degree** \( (u) \triangleq \# \text{ vertices adjacent to } u \)

- **\((C, d)\)-bounded graph**: left degrees \( \leq C \)
  
  \[\text{Right degree} \leq d.\]

- **Neighborhood** \( \Pi(S) = \{ v | \exists u \in S \text{ s.t. } u \leftrightarrow v \}\)

- **\((\delta, \delta)\)-expander**: \( G = (L, R, E) \)
  
  \[|L| = n, \quad |R| = m\]

  \[\forall S \subseteq L \quad |S| \leq \delta n\]

  \[\Rightarrow \quad |\Pi(S)| \geq \delta \cdot |S|\]
Expansion?

- One measure of connectivity
- Good expanders = large $\gamma$, $S$
  
  $\implies$ well-connected
- How large can $\gamma$, $S$ be?
  - Clearly $S < 1$.
  - Also $S < \frac{1}{\gamma} \cdot \frac{m}{n}$

More critical parameter: $\gamma$

- $\gamma > 0$ non-trivial to get (with $m < n$)
- $\gamma \leq e$ even single vertex does not expand by more.
- Random graph [Gallager] gets
  
  $\gamma \approx C$; $S = \sqrt{\frac{2}{\gamma} \cdot \frac{m}{n}}$
  
  Assume
- Explicitly .... getting there ... "YES WE CAN"
**Distance via Expansion**

**Theorem [Sipser-Spielman]:** If bip. graph $G = (L,R,E)$ is $(c,d)$-bounded $(\delta,\delta)$-expander

and $\delta > \frac{c}{2}$ then $C_{G}$, code associated with $G$ has distance $\geq \delta$.

**Key Concept:** Unique Neighborhood $\Gamma^{+}(S)$ for $S \subseteq L$, $\Gamma^{+}(S) = \{ v \in S \mid \exists u \in S \text{ s.t. } u \sim v \}$.

$G$ is a $(\delta,\delta)$ unique expander if

$\forall S \subseteq L$, $|S| \leq \delta \Rightarrow |\Gamma^{+}(S)| \geq \delta \cdot |S|$.
Key Lemma: \( G \) is \((c, d)\)-bounded \((\delta, \delta)\)-expander

\[ \Rightarrow G \text{ is } (2\delta - c, \delta)\)-unique expander. \]

(useful only if \( \delta > \frac{c}{2} \))

Proof: Fix \( S \subset L \)

\[ S \]

\( \leftarrow \text{Unique} \)

\( \leftarrow \text{Two} \)

Let's count edges \( F \) incident to \( S \)

\[ |F| \leq c \cdot |S| \]

\[ |F| \geq |u_1 + 2 \cdot l_1| \]

\[ \Rightarrow |u_1 + 2 \cdot l_1| \leq c \cdot |S| \]
But using expansion, we also have

\[ |u| + |r| \geq \alpha \cdot |S| \]  

\[ \text{2} \times 2 - \text{1} \]

\[ |u| \geq (2\alpha - \epsilon) |S| \text{ as desired} \]

Proof of Distance:

- Let \( x_1 \ldots x_n \) be vector assignment of \( w^t < s_n \)

- \( S = \{ i \mid x_i = 1 \} \); \( |S| < s_n \)

- \( |T^+(S)| > (2\alpha - \epsilon) |S| \geq 1 \)

- \( v \in T^+(S) \Rightarrow v^{th} \text{ constraint is not satisfied} \)

\( \Rightarrow x_1 \ldots x_n \text{ not a codeword} \Rightarrow s_n + 1 \)
Utility:

- At the time (1994) best expanders achieved $\gamma \to \infty$ with $\delta > 0$ as $c \to \infty$.

- No explicit constructions achieving $\gamma > \frac{c}{2}$.

- Two fixes:
  - Roll clock forward to 2002.
  - Use [Tanner]’s generalization.
A Convenient (potentially incorrect) Abstraction of Expander Technology in the 90s.

∀ Y ∈ C s.t. ∀ d ∈ S ≥ 0 s.t. A

suf. large n

Can construct (c, d)-bounded

(γ, δ)-expander on n left vertices.

Theorem: [Tanner + SS]: Combining G that is

(c, d)-bounded with (γ, δ)-expander

with a [d, l, Δ]-code for constraints
gives code of distance 8 if γ ≥ \frac{c}{Δ}. 

Proof. Similar to before. Fix \( S \), \( |S| \leq \delta n \).

Let \( U_\Delta = \text{neighbors with fewer than } \Delta \text{ neighbors in } S \).

\[ T_\Delta = \text{rest}. \]

Have \( |U_\Delta| + |T_\Delta| \geq \delta \cdot |S| \).

\[ |U_\Delta| + \Delta \cdot |T_\Delta| \leq c \cdot |S| \]

\[ \Rightarrow |U_\Delta| \geq \frac{1}{\Delta-1} (\Delta \delta - c) \cdot |S| \]

\[ \ldots \]
Using Theorem + "Abstraction"

- Pick $\gamma$ as you like (hmm...?)
- Let $C$ be as given by abstraction.
- Pick $\Delta > \frac{C}{\gamma}$

- Pick $(d, l)$ so that $[d, l, \Delta]$ code exists $\Rightarrow (d - l) c < d$

$d - l \geq \Delta \log d \Rightarrow$ code exists

$c \Delta \log d < d$

$\Rightarrow \frac{d}{\log d} < \Delta c$

$\Rightarrow l = \Delta c \log \Delta c$; $d = l + 2 \Delta \log l$ works.