Expander codes

'Good code': Positive constant rate
- Positive constant min. distance
- Efficient to decode/encode

Theorem (Margulis)

For $d \geq 64$, there exists a left $d$-regular bipartite graph with $|L| = n$ and $|R| = \frac{3}{4}n$,
\[ |N(S)| \geq 0.8d |S| \] for all $S \subseteq L$: $|S| \leq \frac{0.02}{d} n$

- Explicit construction!

Tanner code (a type of linear code)

Take $d = 64$ and the Margulis expander.

Our code is defined using the parity check matrix $H$, (codewords) where elements of the code are all length $|L|$ strings $\mathbf{z}$ s.t. $H \mathbf{z} = 0$. (in binary)

Message length $= |L| - |R| = n - \frac{3}{4}n = \frac{1}{4}n$

$[n, \frac{1}{4}n, ???]$ code : rate $= \frac{1}{4}$, constant
Claim: Distance of the code is $> \frac{0.02}{64} n$

Assume minimum distance $\leq \frac{0.02}{64} n$.

If nonzero codeword $z$ with Hamming weight $|z| \leq \frac{0.02}{64} n$. Let $S = \{ u \in [n] / z_u = 1 \} \subseteq \text{vertices}$

Since $z$ is nonzero, $S \neq \emptyset$, and hence $|S| \leq \frac{0.02}{64} n$.

\[
\begin{bmatrix}
1 \\
1 \\
1 \\
\end{bmatrix}
\begin{bmatrix}
1 \\
1 \\
1 \\
\end{bmatrix} = 
\begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}
\]

Each vertex in $R$ must be adjacent to an even number of vertices in $S$.

Claim: $|S| \leq \frac{0.02}{64} n$. There exists a $v \in N(S)$ with exactly one neighbor in $S$.

If all $v \in N(S)$ have $\geq 2$ neighbors in $S$, then since $|N(S)| \geq 0.8 d |S|$:

$|E(S, N(S))| \geq 2 |N(S)| \geq 2 \cdot 0.8 d |S| > 64 |S|$

But the left partition is 64-regular!

Hence we are done. $- z$ is not a code word, so distance $> \frac{0.02}{64} n$. 
Decoding is efficient! Flip bits of \( z \) that decrease the Hamming weight of \( Hz \).
Corrects \( < \frac{D}{2} \) errors in polylogarithmic time.

**ERROR REDUCTION**
(Miller-Rabin primality)

Algorithm A, probabilistic
Uses \( n \) random bits, returns YES if YES
fails with prob \( p \leq 1\% \)

Old way of reducing error: repeat \( d \) times and random bits, fails with prob \( \leq \frac{1}{\log n} \)

Expanders:
Take the Margulis expander, except with \( |L| = |R| = 2^n \).
Vertices on each side are indexed by \( n \)-bit strings.
\[ |N(s)| \geq 0.8 \delta |s| \quad \text{if } |s| \leq \frac{\delta}{2} (2^n) \]

Pick random vertex \( v \in L \)
compute \( d \) neighbors of \( v \) and use them all for \( A \! \)!
Uses NO additional random bits, but what's the error...
How many 'bad' initial seeds are there?

Let $B_x \subseteq R$ be the set of 'bad' strings.

$|B_x| = p \cdot 2^n$

Let $C \subseteq L$ be the set of 'bad' choices $v \in L$

where $N(v) \subseteq B_x$

Claim: $|C| < \frac{0.02}{d} \cdot 2^n$

If not, take $S \subseteq C$ s.t. $|S| = \frac{0.02}{d} \cdot 2^n$

By expander properties, $|N(S)| \geq 0.8 \cdot d \cdot |S|$

$= 0.8 \cdot 0.02 \cdot 2^n = 0.016 \cdot 2^n > |B_x| \quad \text{(because of p)}$

So error has been reduced to $\frac{0.02}{d}$ with NO additional random bits!

Random walks:

Start with random vertex $v_1$

Take a random walk $v_1 v_2 v_3 v_4 \ldots v_m$.

Use the binary strings of $v_i$ as a set

Due to the spectral properties of expanders, $v_i$ are 'approximately' random.

Error $\approx 0.01^n$, random bits $n + m \log d$