Lecture 12: Codes on Graphs

Course instructor: Alexey Frolov

al.frolov@skoltech.ru

Teaching Assistant: Stanislav Kruglik

stanislav.kruglik@skolkovotech.ru

February 27, 2018

Outline

Outline

1. LDPC codes

2. Expander graphs and expander codes

Introduction

- LDPC codes were invented by Robert G. Gallager in the 1960s and forgotten for three decades.
- After Turbo codes were invented 1993, LDPC codes found new attention

Introduction

Robert G. Gallager

LDPC codes

- LDPC codes are defined with use of *sparse* parity-check matrix, i.e. the percentage of 1s in the parity check matrix for a LDPC code is low.
- A regular LDPC code has the property that:
  - every code digit is contained in the same number of equations,
  - each equation contains the same number of code symbols.
- An irregular LDPC code relaxes these conditions.

Definition by parity-check matrix: [Gallager, '62]

$$
H = \left[\begin{array}{cccccccccccccccccccc}
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{array}\right]_{15 \times 20}
$$

Code: $\{\mathbf{v} \mid \mathbf{v}\mathbf{H}^{\mathrm{T}} = \mathbf{0}\}$

(null space of a **sparse** parity-check matrix $\mathbf{H}$)

Definition by parity-check matrix

Definition by parity-check matrix: [Gallager, '62]

$$
H = \left[\begin{array}{cccccccccccccccccccc}
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{array}\right]
$$

15×20

Code: $\{\mathbf{v} \mid \mathbf{v}\mathbf{H}^{\mathrm{T}} = \mathbf{0}\}$

(null space of a **sparse** parity-check matrix $\mathbf{H}$)

Regular LDPC code:

Column weight: $J = 3$

Row weight: $K = 4$

$$\boxed{R \geq 1 - \frac{J}{K}}$$

Definition by parity-check matrix: [Gallager, '62]

$$
H = \begin{array}{|cccccccccccccccccccc|}
\hline
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1\\
\hline
\end{array}
$$

15×20

Code: $\{\mathbf{v} \mid \mathbf{v}\mathbf{H}^{\mathrm{T}} = \mathbf{0}\}$
(null space of a **sparse** parity-check matrix $\mathbf{H}$)

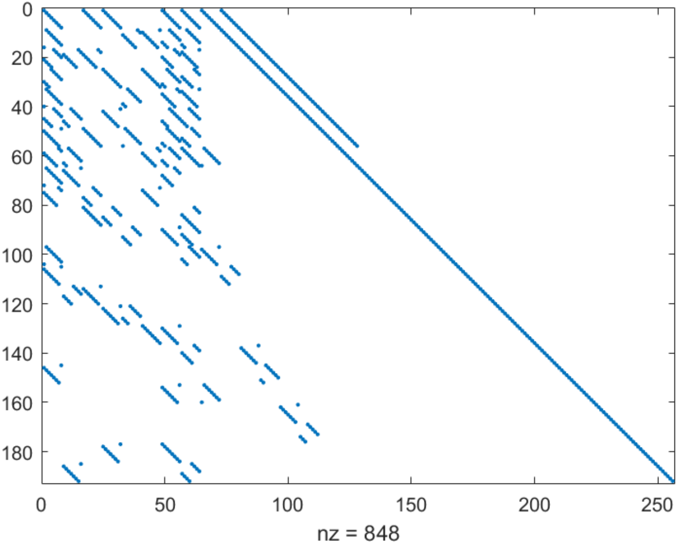
Regular LDPC code:

Column weight: $J = 3$
Row weight: $K = 4$

$$\boxed{R \geq 1 - \frac{J}{K}}$$

- If the row and column weights J and K are not constant, then the LDPC code is **irregular** (more later)

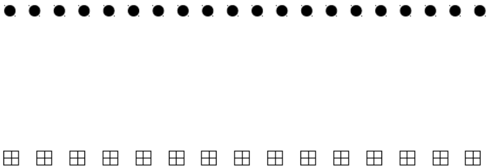
Irregular LDPC example, PCM with $R = 1/4$



Representation by bi-partite graph: [Tanner, '81]

$$H = \underbrace{\begin{bmatrix}
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{bmatrix}}_{15 \times 20}$$

$n = 20$ **variable nodes** of degree $J = 3$



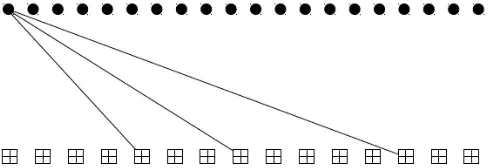
$l = 15$ **check nodes** of degree $K = 4$

Tanner graph

Representation by bi-partite graph: [Tanner, '81]

$$H = \left[\begin{array}{cccccccccccccccccccc}
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{array}\right]_{15 \times 20}$$

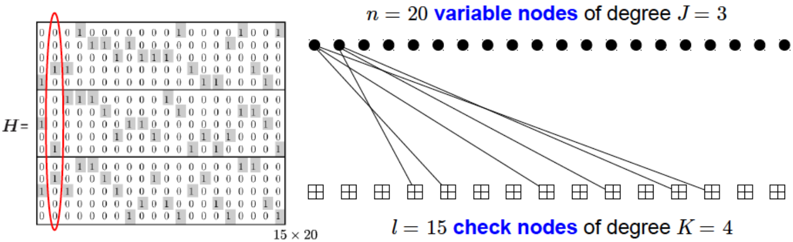
$n = 20$ **variable nodes** of degree $J = 3$



$l = 15$ **check nodes** of degree $K = 4$

Tanner graph

Representation by bi-partite graph: [Tanner, '81]

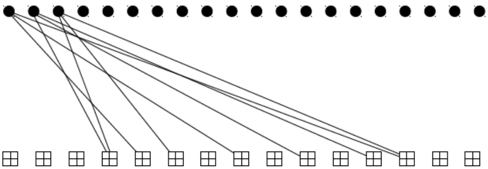


Tanner graph

Representation by bi-partite graph: [Tanner, '81]

$$
H = \left[\begin{array}{cccccccccccccccccccc}
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{array}\right]_{15 \times 20}
$$

$n = 20$ **variable nodes** of degree $J = 3$



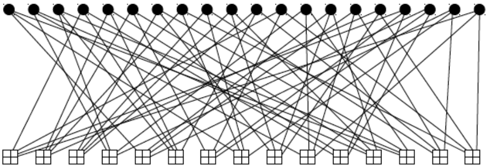
$l = 15$ **check nodes** of degree $K = 4$

Tanner graph

Representation by bi-partite graph: [Tanner, '81]

$$H = \left[\begin{array}{cccccccccccccccccccc}
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{array}\right]_{15 \times 20}$$

$n = 20$ **variable nodes** of degree $J = 3$



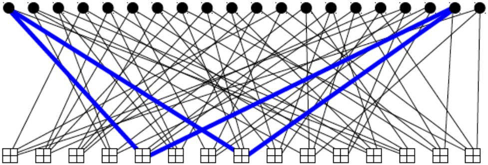
$l = 15$ **check nodes** of degree $K = 4$

Tanner graph

Representation by bi-partite graph: [Tanner, '81]

$$H = \left[\begin{array}{cccccccccccccccccccc}
0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0&1\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0\\
\hline
0&0&1&1&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&1&1&0&0\\
1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0\\
1&0&1&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1
\end{array}\right]_{15 \times 20}$$

$n = 20$ **variable nodes** of degree $J = 3$



$l = 15$ **check nodes** of degree $K = 4$

- Tanner graphs typically contain **cycles** (length 4 cycle highlighted above)
- The **girth** of a Tanner graph is the length of the shortest cycle (4 in this example)

Gallager's ensemble of LDPC codes

Ensemble $\mathcal{E}(N, K, J)$

$$\mathbf{H} = \begin{pmatrix} \pi_1(\mathbf{H}_b) \\ \pi_2(\mathbf{H}_b) \\ \vdots \\ \pi_J(\mathbf{H}_b) \end{pmatrix}_{\ell b \times b n_0}$$

where

$$\mathbf{H}_b = \begin{pmatrix} 11\ldots1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & 11\ldots1 & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & 11\ldots1 \end{pmatrix}_{b \times bK}$$

Lower bound on the minimum distance

Theorem (Gallager'62)

For any $J > 2$ there exists $\delta^*(K, J) > 0$ such that:

- *there are codes in the ensemble $\mathcal{E}(N, K, J)$ for which the following inequality holds*

$$d(\mathcal{C}) \geq (\delta^* - \varepsilon)N \tag{1}$$

- *the number of such codes $(G(N, K, J))$ satisfy the following relation*

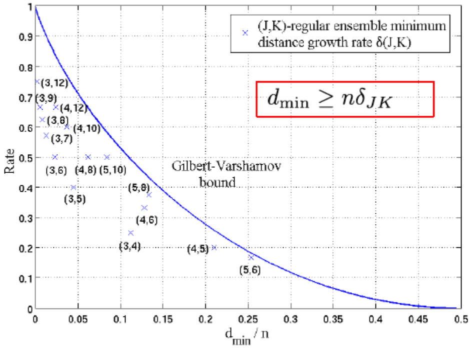
$$\lim_{N\to\infty} \frac{G(N, K, J)}{|\mathcal{E}(N, K, J)|} = 1.$$

The value δ^* is the smallest positive root of equation

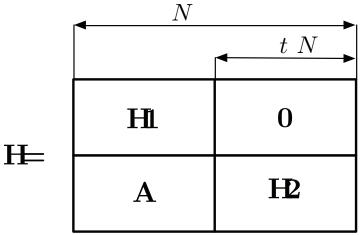
$$(J-1)h(\delta) + J \max_{s>0}\left(\delta \log \delta - \frac{1}{K} g_0(s, K)\right) = 0.$$

Lower bound on the minimum distance

- δ_{JK} is called the **typical minimum distance ratio**, or **minimum distance growth rate**, of a code ensemble



Upper bound on the minimum distance



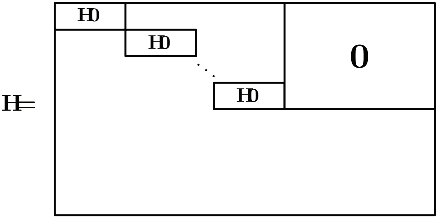
Lemma

Let the parity-check matrix $\boldsymbol{H}$ of the code $\mathcal{C}$ has a special form shown above, then

$$R(\mathcal{C}) \leq R(\mathcal{C}_1)(1-\tau) + R(\mathcal{C}_2)\tau,$$

where the codes C_1 and C_2 correspond to parity-check matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$.

Upper bound on the minimum distance



Theorem

Let $\mathcal{C}$ be a generalized LDPC code of length N, rate R, minimum distance δN, with constituent $[n_0, R_0, d_0]$ code $\mathcal{C}_0$. Then for sufficiently large N the following inequality holds

$$R(\mathcal{C}) \leq \min_{\frac{q}{q-1}\delta \leq \tau \leq 1} \left\{ R_0(1-\tau) + R^*\left(\frac{\delta}{\tau}\right)\tau \right\} + o(1),$$

where $R^*(\delta)$ is any upper bound on the code rate.

Bit-flipping decoding algorithm

- If there exists a variable node, such that the number of unsatisfied check nodes is bigger then the number of satisfied check nodes $\Rightarrow$ flip the bit.
- Continue until such variable nodes exist.

Outline

Expander graph

Let $G = (V, E)$ be a graph with n vertixes. Let us denote by

$$\Gamma(v) = \{u : (u, v) \in E\}$$

the neighbors of the vertex v and by

$$\Gamma(S) = \bigcup_{v \in S} \Gamma(v)$$

the neighbors of the vertex set S.

Expander graph

Definition

A graph G is called an (ω, α)-expander, if

$$\forall S \subset V, |S| \leq \omega n \Rightarrow \Gamma(S) > \alpha |S|.$$

Expander graph

In what follows we consider the following graph types:

- G – bipartite ($V = V_L \sqcup V_R$) (J, K)–regular, $\deg(v) = J, v \in V_L \;\; \deg(v) = K, v \in V_R$.
- G – Δ-regular, $\deg(v) = \Delta$;

Expansion of random graph

Lemma

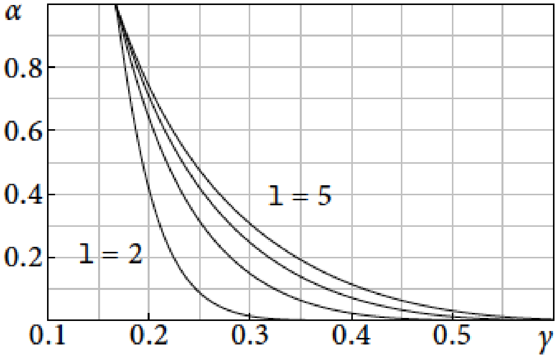
Let G be a graph chosen uniformly from the ensemble of (J, K)-regular bipartite graphs and let $n \to \infty$. For a given $\gamma \in [\frac{1}{K}, 1 - \frac{1}{J})$ let ω be the positive solution of the equation

$$\begin{aligned} \frac{J-1}{J} h(\omega) &- \frac{1}{K} h(\omega\alpha K) \\ &- \omega\alpha K h\left(\frac{1}{\alpha K}\right) = 0. \end{aligned}$$

Then for $0 < \omega' < \omega$ *and* $\beta = J(1-\alpha) - 1$

$$\Pr(\{G \textit{ is an } (J, K, \omega', \alpha) \textit{ expander}\}) \geq 1 - O(n^{-\beta}).$$

Expansion of random graph



Bit-flipping

Theorem

Let G be (J, K) regular bipatrite graph, which is also $\left(\omega, \alpha = \frac{3}{4}J\right)$-expander. The algorithm is able to correct up to $\frac{\omega N}{2}$ errors.

Sipser-Spielman construction

Consider a regular graph. Associate check codes with vertexes and codeword bits with edges.

Expander mixing lemma

Lemma

Let $G = (V, E)$ be a Δ-regular graph with n vertexes and the second largest eigenvalue λ. Let $S \subset V$, $|S| = \gamma n$ then

$$\left| e(S) - \frac{1}{2}\Delta\gamma^2 n \right| \leq \frac{1}{2}\lambda\gamma(1-\gamma)n.$$

Proof

Consider $\mathbf{f}$:

$$\mathbf{f}(i) = -\frac{1}{|S|}, \quad i \in S$$
$$\mathbf{f}(i) = \frac{1}{n-|S|}, \quad \text{otherwise}$$

Let A be an adjacency matrix of a graph G.

$$|(Af,f)| \le \lambda(f,f).$$

$$(Af,f) = 2\sum_{(i,j)\in E} f(i)f(j) = \Delta\sum_{i=1}^{n} f^2(i) - \sum_{(i,j)\in E}(f(i)-f(j))^2.$$

$$\sum_{i=1}^{n} f^2(i) = 1/|S| + 1/(n-|S|).$$

Proof

$$\sum_{(i,j)\in E} (f(i) - f(j))^2 = e(S, \bar{S})(1/|S| + 1/(n - |S|))^2.$$

$$2e(S) + e(S, \bar{S}) = \Delta|S|.$$

Distance bound

Theorem

$$\delta(\mathcal{C}) \geq \left(\frac{\delta_0 - \frac{\lambda}{\Delta}}{1 - \frac{\lambda}{\Delta}} \right) \delta_0.$$

Zemor's construction

Choose underlying regular graph to be bipartite.

Thank you for your attention!