Lecture 13: How to construct LDPC codes

Course instructor: Alexey Frolov

al.frolov@skoltech.ru

Teaching Assistant: Stanislav Kruglik

stanislav.kruglik@skolkovotech.ru

March 3, 2017

Outline

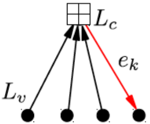
We know how to decode LDPC codes

- Messages are **log-likelihood ratios (LLRs)**:

$$L_{ch} = \log \frac{\mathbb{P}(r|v=0)}{\mathbb{P}(r|v=1)}$$

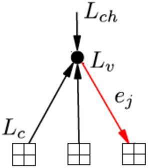
BSC: $r \in \{0,1\}$
AWGNC: $r \in \mathbb{R}$

Check node update:



$$L_c(e_k) = 2\,\mathrm{atanh}\left(\prod_{k' \neq k} \tanh\left(\frac{L_v(e_{k'})}{2}\right)\right)$$

Variable node update:



$$L_v(e_j) = L_{ch} + \sum_{j' \neq j} L_c(e_{j'})$$

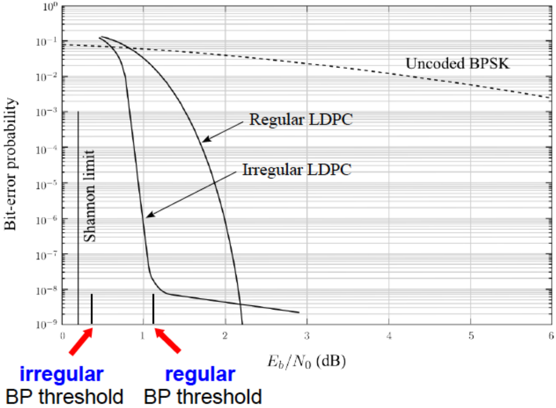
Problems

In what follows the decoding algorithm is fixed.

The decoding algorithm is suboptimal (there are cycles in the Tanner graph).

How to optimize LDPC parity-check matrices for this decoder?

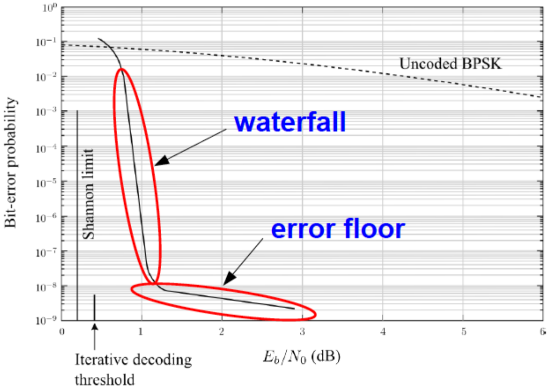
Regular vs Irregular LDPC codes



- **Irregular** LDPC code ensembles can have optimized thresholds **close to capacity**
- **Regular** LDPC code ensembles are asymptotically good and have good graph properties, resulting in a **low error floor**

Waterfall vs error floor

- The **Shannon limit** defines capacity and is a property of the physical channel.



- The **iterative decoding threshold** depends on the code structure and iterative decoding algorithm in use.
- Capacity-approaching LDPC codes typically display a **waterfall** (related to their threshold) and an **error-floor** (related to their graph/distance properties).

How to define the ensemble of irregular LDPC codes?

Degree (weight if we consider PCM) distribution polynomials

$$\Lambda(x) = \sum_{i=1}^{l_{\max}} \Lambda_i x^i \text{ (variable nodes)}$$

and

$$P(x) = \sum_{i=1}^{r_{\max}} P_i x^i \text{ (check nodes)},$$

where Λ_i and P_i are numbers of variable/check nodes of degree i.

Properties:

$$\Lambda(1) = n, P(1) = (1-R)n, R = 1 - \frac{P(1)}{\Lambda(1)}$$

How to define the ensemble of irregular LDPC codes?

Degree (weight if we consider PCM) distribution polynomials

$$L(x) = \frac{\Lambda(x)}{\Lambda(1)}$$

and

$$Q(x) = \frac{P(x)}{P(1)},$$

where L_i and Q_i are *fractions* of variable/check nodes of degree i.

Edge perspective

For the asymptotic analysis it is more convenient to take on an edge perspective. Define:

$$\lambda(x) = \sum_i \lambda_i x^{i-1}$$

and

$$\rho(x) = \sum_i \rho_i x^{i-1},$$

where λ_i and ρ_i are *fractions* of *edges* that connect to variable(check) nodes of degree i.

Properties:

$$\lambda(x) = \frac{L'(x)}{L'(1)}, \rho(x) = \frac{Q'(x)}{Q'(1)}.$$

Example

Consider $[7,4]$ Hamming code

$$\mathbf{H}_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \Lambda(x) &= 3x + 3x^2 + x^3 \\ L(x) &= \frac{3}{7}x + \frac{3}{7}x^2 + \frac{1}{7}x^3 \\ \lambda(x) &= \frac{1}{4} + \frac{1}{2}x + \frac{1}{4}x^2 \end{aligned}$$

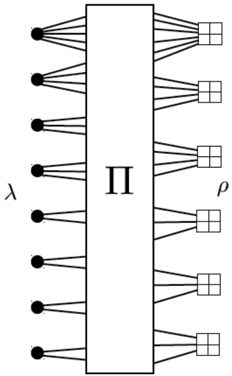
How to define the ensemble of irregular LDPC codes?

- **Node degrees**: random variables [Luby, et al., '97]

$$\lambda(x) = \sum_k \lambda_k x^{k-1}$$ ← variable node distribution

$$\rho(x) = \sum_k \lambda_k x^{k-1}$$ ← check node distribution

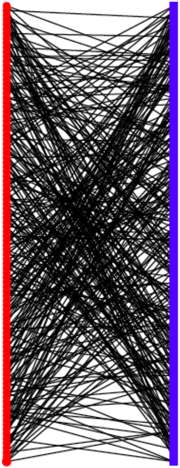
LDPC Ensemble:



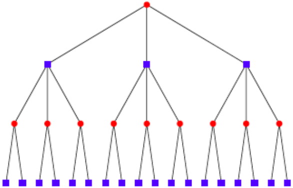
variable nodes **check nodes**

Outline

Computational graph



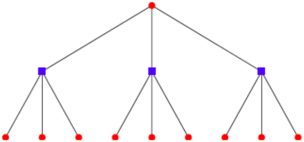
$$\lim_{n\to\infty} \mathbb{E}[P_b(\mathrm{G}, n, \ell)]$$



probability that computation graph of fixed depth becomes tree tends to 1 as n tends to infinity

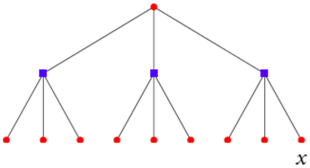
Density evolution, BEC

Luby, Mitzenmacher, Shokrollahi, Spielman, and Steman ‘97



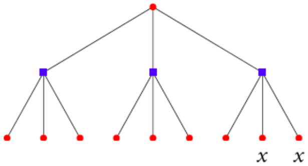
Density evolution, BEC

Luby, Mitzenmacher,
Shokrollahi, Spielman,
and Steman '97



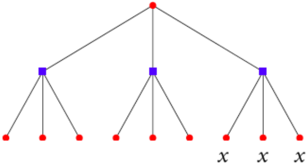
Density evolution, BEC

Luby, Mitzenmacher, Shokrollahi, Spielman, and Steman ‘97



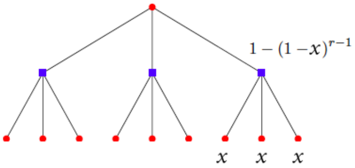
Density evolution, BEC

Luby, Mitzenmacher,
Shokrollahi, Spielman,
and Steman '97



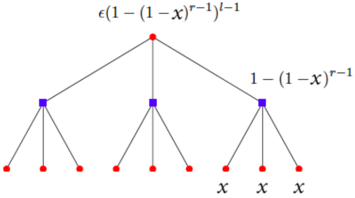
Density evolution, BEC

Luby, Mitzenmacher,
Shokrollahi, Spielman,
and Steman ‘97



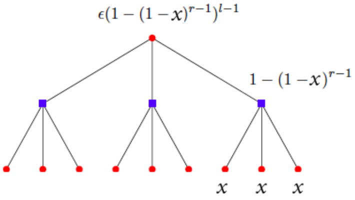
Density evolution, BEC

Luby, Mitzenmacher, Shokrollahi, Spielman, and Steman '97



Density evolution, BEC

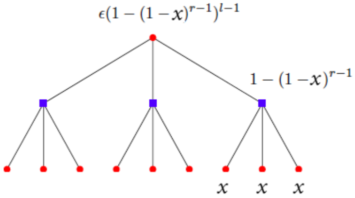
Luby, Mitzenmacher,
Shokrollahi, Spielman,
and Steman '97



$$x_t = \epsilon(1-(1-x_{t-1})^{r-1})^{l-1}$$

Density evolution, BEC

Luby, Mitzenmacher, Shokrollahi, Spielman, and Steman '97



$$x_\ell = \epsilon\lambda\left(1-\rho\left(1-x_{\ell-1}\right)\right)$$

EXAMPLE 3.52 (DENSITY EVOLUTION FOR $(\lambda(x) = x^2, \rho(x) = x^5)$). For the degree distribution pair $(\lambda(x) = x^2, \rho(x) = x^5)$ we have $x_0 = \epsilon$ and for $\ell \geq 1$, $x_\ell = \epsilon(1 - (1 - x_{\ell-1})^5)^2$. For example, for $\epsilon = 0.4$ the sequence of values of x_ℓ is 0.4, 0.34, 0.306, 0.2818, 0.2617, 0.2438, and so forth. ◇

Density evolution, BEC

EXAMPLE 3.52 (DENSITY EVOLUTION FOR $(\lambda(x) = x^2, \rho(x) = x^5)$). For the degree distribution pair $(\lambda(x) = x^2, \rho(x) = x^5)$ we have $x_0 = \epsilon$ and for $\ell \geq 1$, $x_\ell = \epsilon(1 - (1 - x_{\ell-1})^5)^2$. For example, for $\epsilon = 0.4$ the sequence of values of x_ℓ is 0.4, 0.34, 0.306, 0.2818, 0.2617, 0.2438, and so forth. $\diamond$

Does this sequence converge to 0 ?

Density evolution, BEC

LEMMA 3.53 (MONOTONICITY OF $f(\cdot,\cdot)$). For a given degree distribution pair (λ, ρ) define $f(\epsilon, x) = \epsilon\lambda(1 - \rho(1 - x))$. Then $f(\epsilon, x)$ is increasing in both its arguments for $x, \epsilon \in [0, 1]$.

LEMMA 3.54 (MONOTONICITY WITH RESPECT TO CHANNEL). Let (λ, ρ) be a degree distribution pair and $\epsilon \in [0, 1]$. If $\mathrm{P}^{\mathrm{BP}}_{\mathcal{T}_\ell}(\epsilon) \xrightarrow{\ell\to\infty} 0$ then $\mathrm{P}^{\mathrm{BP}}_{\mathcal{T}_\ell}(\epsilon') \xrightarrow{\ell\to\infty} 0$ for all $0 \le \epsilon' \le \epsilon$.

phase transition: ϵ^{BP} so that

$$x_t \to 0 \text{ for } \epsilon < \epsilon^{\mathrm{BP}}$$

$$x_t \to x_\infty > 0 \text{ for } \epsilon > \epsilon^{\mathrm{BP}}$$

Threshold

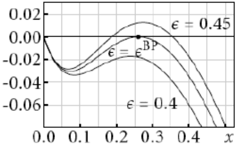
Definition 3.56 (Threshold of Degree Distribution Pair). The *threshold* associated with the degree distribution pair (λ, ρ), call it $\epsilon^{\mathrm{BP}}(\lambda, \rho)$, is defined as

$$\epsilon^{\mathrm{BP}}(\lambda, \rho) = \sup\{\epsilon \in [0,1] : \mathrm{P}^{\mathrm{BP}}_{\mathcal{T}_\ell(\lambda,\rho)}(\epsilon) \xrightarrow{\ell \to \infty} 0\}. \qquad \triangledown$$

Example 3.57 (Threshold of $(\lambda(x) = x^2, \rho = x^5)$). Numerical experiments show that $\epsilon^{\mathrm{BP}}(3,6) \approx 0.42944$. $\diamond$

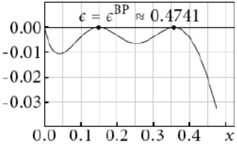
Fixed point characterization, BEC

$f(\epsilon, x) = \epsilon\lambda(1 - \rho(1 - x))$.



$f(\epsilon, x) - x = \epsilon(1 - (1 - x)^5)^2 - x$

$$(\lambda, \rho) = (x^2, x^5)$$



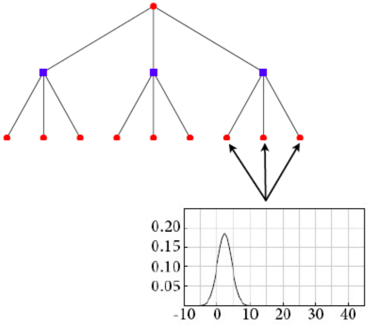
$$\lambda(x) = 0.106257x + 0.486659x^2 + 0.010390x^{10} + 0.396694x^{19}$$

$$\rho(x) = 0.5x^7 + 0.5x^8$$

Density evolution, BEC

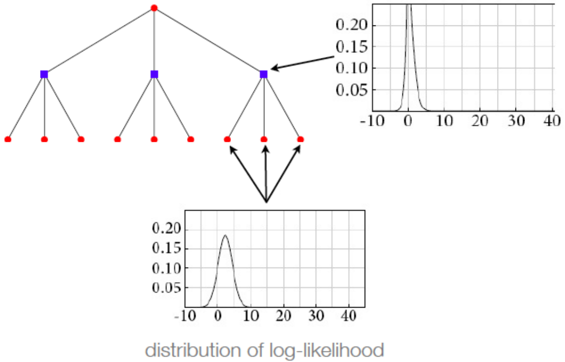
$\mathtt{l}$	$\mathtt{r}$	$r(\mathtt{l},\mathtt{r})$	$\epsilon^{\text{Sha}}(\mathtt{l},\mathtt{r})$	$\epsilon^{\text{BP}}(\mathtt{l},\mathtt{r})$
3	6	$\frac{1}{2}$	$\frac{1}{2} = 0.5$	≈ 0.4294
4	8	$\frac{1}{2}$	$\frac{1}{2} = 0.5$	≈ 0.3834
3	5	$\frac{2}{5}$	$\frac{3}{5} = 0.6$	≈ 0.5176
4	6	$\frac{1}{3}$	$\frac{2}{3} \approx 0.667$	≈ 0.5061
3	4	$\frac{1}{4}$	$\frac{3}{4} = 0.75$	≈ 0.6474

Density evolution, AWGNC

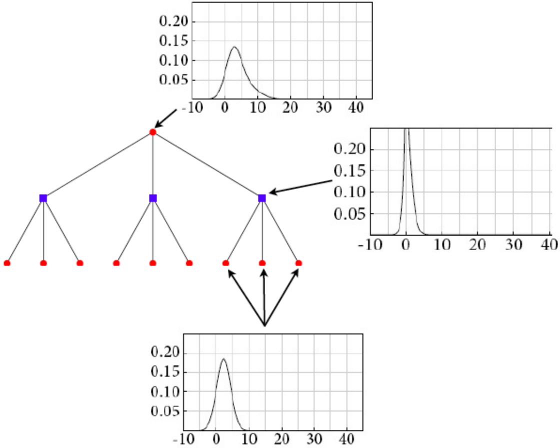


distribution of log-likelihood

Density evolution, AWGNC



Density evolution, AWGNC



distribution of log-likelihood

Outline

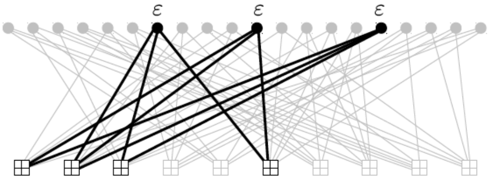
Error floor

- ➡ Error events in the **waterfall** typically result from **large** decoding failures (a large number of symbols decoded incorrectly)
- ➡ Error events in the **error floor** typically result from **small** decoding failures (only a few symbols decoded incorrectly)
- The minimum distance is a **code property**; under ML decoding, a large minimum distance results in a low error floor
- Under sub-optimal **iterative BP decoding**, the error floor is also affected by small failures arising due to **weaknesses in the Tanner graph**
- ➡ These graphical weaknesses have been studied extensively for a variety of channels and are known collectively as **pseudocodewords** [Frey et al '98], **stopping sets** [Di et al '02], **near-codewords** [MacKay & Postol '03], **trapping sets** [Richardson '03], **elementary trapping sets** [Laendner & Milenkovic '05], and **absorbing sets** [Dolecek et al '07].

Error floor, BEC

- On the BEC, the cause of failures is **stopping sets** [Di, et al. '02].

Definition: A stopping set is a subset S of the variable nodes such that all neighboring check nodes are connected to S at least twice



Example stopping set in a $(3,6)$-regular Tanner graph

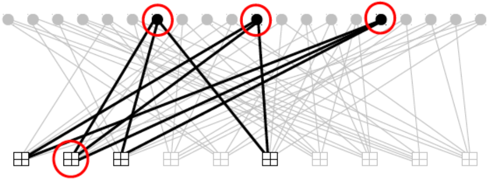
- If the highlighted nodes are all erasures then the BP decoder will fail to correct them

➡ Message-passing decoding is suboptimal! The MAP decoder fails if and only if the set of erasures contains the set of all non-zero positions in the codeword.

Error floor, AWGNC

- On the AWGNC, failures are attributed to **trapping sets** [Richardson '03].

Definition: An (a,b) general trapping set $\tau_{a,b}$ of a bipartite graph is a set of a variable nodes which induce a subgraph with exactly b odd-degree check nodes.



A $(3,1)$ trapping set in a $(3,6)$-regular Tanner graph

- Low connectivity outside the set causes the iterative decoder to become trapped and fail to correct the symbols in the set
- Certain types of trapping sets with small a and b, such as **elementary trapping sets** and **absorbing sets**, are known to be particularly harmful

Progressive-edge grows algorithm

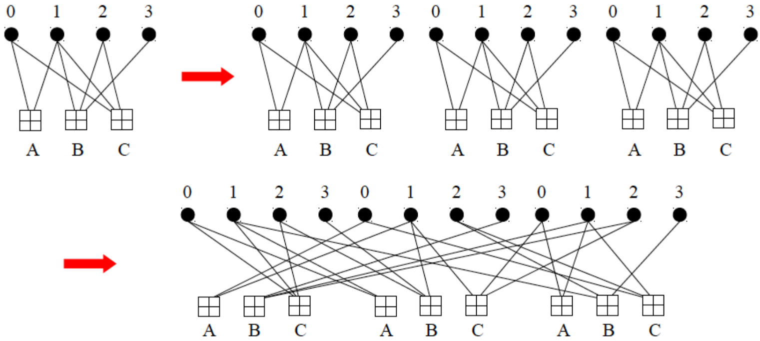
Input: $L(x)$
Output: PCM with "big" girth

Main idea: greedy algorithm, add edges to the graph in a sequential manner. Each time choose the connection, that maximizes the girth.

Outline

Protograph-based LDPC codes

- Codes can be constructed from a **protograph** using a **copy-and-permute** operation



[Tho05] J. Thorpe, “Low-Density Parity-Check (LDPC) codes constructed from protographs”, *Jet Propulsion Laboratory INP Progress Report*, Vol. 42-154 Aug. 2003.

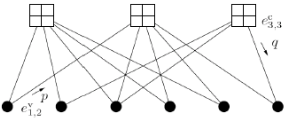
Protograph-based LDPC codes

- **Compact representation** of a permutation matrix based ensemble by a base matrix:

$$\mathbf{H} = \begin{bmatrix} \mathbf{\Pi}_{1,1} & \mathbf{\Pi}_{1,2} & \mathbf{\Pi}_{1,3} & \mathbf{\Pi}_{1,4} & \mathbf{\Pi}_{1,5} & \mathbf{0} \\ \mathbf{\Pi}_{2,1} & \mathbf{0} & \mathbf{\Pi}_{2,3} & \mathbf{\Pi}_{2,4} & \mathbf{\Pi}_{2,5} & \mathbf{\Pi}_{2,6} \\ \mathbf{0} & \mathbf{\Pi}_{3,2} & \mathbf{\Pi}_{3,3} & \mathbf{0} & \mathbf{0} & \mathbf{\Pi}_{3,6} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

base matrix



protograph

[Tho05] J. Thorpe, “Low-Density Parity-Check (LDPC) codes constructed from protographs”, *Jet Propulsion Laboratory INP Progress Report*, Vol. 42-154 Aug. 2003.

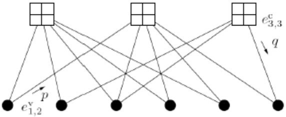
Protograph-based LDPC codes

- **Compact representation** of a permutation matrix based ensemble by a base matrix:

$$\underbrace{\mathbf{H} = \begin{bmatrix} \mathbf{\Pi}_{1,1} & \mathbf{\Pi}_{1,2} & \mathbf{\Pi}_{1,3} & \mathbf{\Pi}_{1,4} & \mathbf{\Pi}_{1,5} & \mathbf{0} \\ \mathbf{\Pi}_{2,1} & \mathbf{0} & \mathbf{\Pi}_{2,3} & \mathbf{\Pi}_{2,4} & \mathbf{\Pi}_{2,5} & \mathbf{\Pi}_{2,6} \\ \mathbf{0} & \mathbf{\Pi}_{3,2} & \mathbf{\Pi}_{3,3} & \mathbf{0} & \mathbf{0} & \mathbf{\Pi}_{3,6} \end{bmatrix}}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

base matrix



protograph

[Tho05] J. Thorpe, "Low-Density Parity-Check (LDPC) codes constructed from protographs", *Jet Propulsion Laboratory INP Progress Report*, Vol. 42-154 Aug. 2003.

Quasi-cyclic LDPC codes

Replace permutation matrices with circulant matrices (usually of weight 1).

Why this code is quasi-cyclic?

Thank you for your attention!