Low-density Parity-Check Codes Over the Binary Erasure Channel

Introduction to Graphical Models and Inference for Communications

UC3M

March 3, 2018

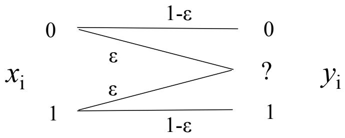
uc3m

Today

- Binary Erasure Channel, coding rate and channel capacity.
- Optimal decoding and the *Classical* coding approach.
- Modern coding theory in a nutshell.
- Introduction to Low-Density Parity Check codes.
- LDPC Asymptotic analysis for the $(3,6)$-LDPC ensemble.

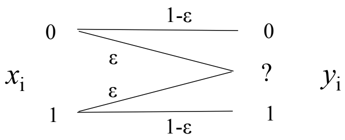
Index

The binary erasure channel (BEC)



Is not this channel model too simple??

- Yes! That's why we like it! We can make analytical predictions.
- *Quite surprisingly, most properties and statements that we encounter in our investigation of LDPC codes over the BEC hold in much greater generality.* (R. Urbanke and T. Richardson, Modern Coding Theory).
- Erasure correcting codes are used in the link layer of every communication stardard!



Uncoded transmission

- Average error probability: ε
- This is ok if $\varepsilon = 10^{-8}$ ... what if $\varepsilon = 0.5$?

Transmission of encoded bits

$$\underbrace{\boldsymbol{m}}_{m \text{ information bits}} \rightarrow \underbrace{\boldsymbol{x}}_{n \text{ encoded bits}} \rightarrow \underbrace{\boldsymbol{y}}_{n \text{ channel observations}} \rightarrow \underbrace{\hat{\boldsymbol{x}}}_{\text{decoded codeword}}$$

$$\mathbf{r} = \frac{m}{n} \qquad \text{(coding rate)}$$

Transmission of encoded bits

$$\underbrace{\boldsymbol{m}}_{m \text{ information bits}} \rightarrow \underbrace{\boldsymbol{x}}_{n \text{ encoded bits}} \rightarrow \underbrace{\boldsymbol{y}}_{n \text{ channel observations}} \rightarrow \underbrace{\hat{\boldsymbol{x}}}_{\text{decoded codeword}}$$

$$\mathbf{r} = \frac{m}{n} \qquad \text{coding rate}$$

- If we let $\mathbf{r} \rightarrow 0$, we can easily find a coding scheme such that $P(\hat{\boldsymbol{x}} \neq \boldsymbol{x}|\boldsymbol{y}) \rightarrow 0$. E.g. a repetition code

$$m_i \in \{0, 1\} \rightarrow \boldsymbol{x} = \underbrace{[m_i \; m_i \; \ldots m_i]}_{m_i \text{ is repeated } n \text{ times}}$$

- Problem solved?

BEC capacity

Channel Capacity

$$C = 1 - \varepsilon$$

- Assume $\varepsilon = 0.5$
- Theoretically, for any $\delta \in (0, 0.5)$ there exits a coding scheme of rate $\mathbf{r} = 0.5 - \delta$ for which $P(\hat{\boldsymbol{x}} \neq \boldsymbol{x} | \boldsymbol{y}) \rightarrow 0$ if we let $\mathbf{n} \rightarrow \infty$

We are wasting resources (information transmission rate, energy) if we use $\mathbf{r} \rightarrow 0$!!

Our goal is to design feasible encoding and decoding schemes that allow us to operate close to channel capacity.

Index

Linear block codes

- All codes used in practice (classic and modern) are linear.
- **Generator matrix:** $\boldsymbol{x} = \boldsymbol{m}\mathbf{G}$ where $\boldsymbol{m} \in \{0,1\}^m$.
- **Parity check matrix:** $\boldsymbol{x}\boldsymbol{H}^T = \mathbf{0} \;\; \forall \boldsymbol{x} \in \mathcal{C}$.
- Each row of the parity check matrix establishes a linear constraint between coded bits.

For a Hamming $(7,4)$ code

$$\boldsymbol{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

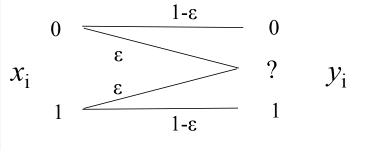
Therefore...

$$x_1 \oplus x_3 \oplus x_5 \oplus x_7 = 0$$
$$x_2 \oplus x_3 \oplus x_6 \oplus x_7 = 0$$
$$x_4 \oplus x_5 \oplus x_6 \oplus x_7 = 0$$

Transmission over the BEC

- Linear block code (m, n) with $\boldsymbol{G}$, $\boldsymbol{H}$ matrices.
- Codeword $\boldsymbol{x}$ is sent.

- Vector $\boldsymbol{y}$ is observed.
- $\mathcal{E}$ index set of erased bits.
- $\mathcal{R}$ index set of received bits.
- $\mathcal{E} \cup \mathcal{R} = \{1, \ldots, n\}$.



Thus, for the BEC

$$\boldsymbol{y}(\mathcal{E}) = ?, \qquad \boldsymbol{y}(\mathcal{R}) = \boldsymbol{x}(\mathcal{R})$$

Optimal decoding over the BEC

- Hamming $(7,4)$ code.
- $\boldsymbol{x} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ is sent.
- $\boldsymbol{y} = \begin{bmatrix} 1 & ? & 1 & 0 & ? & ? & 0 \end{bmatrix}$ is received.
- $\mathcal{E} = \{2,5,6\}$ and $\mathcal{R} = \{1,3,4,7\}$.

Thus, the system of equations

$$\begin{aligned} x_1 \oplus x_3 \oplus x_5 \oplus x_7 &= 0 \\ x_2 \oplus x_3 \oplus x_6 \oplus x_7 &= 0 \\ x_4 \oplus x_5 \oplus x_6 \oplus x_7 &= 0 \end{aligned}$$

can be simplified to

$$\begin{aligned} x_5 &= 0 \\ x_2 \oplus x_6 &= 1 \\ x_5 \oplus x_6 &= 0 \end{aligned}$$

By solving the system of binary equations we get a unique solution $\hat{\boldsymbol{x}} = [1110000] = \boldsymbol{x}$.

- Linear block code (m, n) with $\boldsymbol{G}$, $\boldsymbol{H}$ matrices.
- Codeword $\boldsymbol{x}$ is sent.
- Vector $\boldsymbol{y}$ is observed.
- $\boldsymbol{H}_{\mathcal{E}}$ submatrix of $\boldsymbol{H}$ by taking only those columns with column index in $\mathcal{E}$.

Optimal maximum a posteriori decoding

Find $\boldsymbol{x}(\mathcal{E})$ by solving the following system of equations:

$$\boldsymbol{x}(\mathcal{E})\boldsymbol{H}_{\mathcal{E}}^T = \boldsymbol{x}(\mathcal{R})\boldsymbol{H}_{\mathcal{R}}^T$$

In the former example

$$\begin{bmatrix} x_2 & x_5 & x_6 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Optimal maximum a posteriori (ML) decoding

Find $\boldsymbol{x}(\mathcal{E})$ by solving the following system of equations:

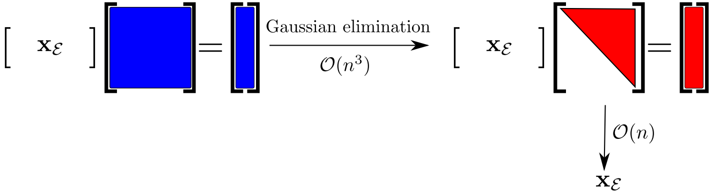
$$\boldsymbol{x}(\mathcal{E})\boldsymbol{H}_{\mathcal{E}}^{T} = \boldsymbol{x}(\mathcal{R})\boldsymbol{H}_{\mathcal{R}}^{T}$$

- If the system has unique solution, then $\hat{\boldsymbol{x}} = \boldsymbol{x}$. No decoding error is possible.
- If the system has multiple solutions, then all solutions are equally likely. We declare a decoding failure.

Optimal decoding Complexity

- In average, there are εn variables erased.
- Gaussian elimination over a system of $\mathcal{O}(n)$ equations requires $\mathcal{O}(n^3)$ operations.
- After Gaussian elimination, $\boldsymbol{x}_{\mathcal{E}}$ is found in $\mathcal{O}(n)$ operations.

$$\boldsymbol{x}(\mathcal{E})\boldsymbol{H}_{\mathcal{E}}^T = \boldsymbol{x}(\mathcal{R})\boldsymbol{H}_{\mathcal{R}}^T$$



Classic channel coding theory

- Decoding via optimal Maximum Likelihood/Maximum a posteriori rules.
- Design the code such that the performance of optimal decoding is as good as possible.
- Strong algebraic structure so that optimal decoding can be solved more efficiently.
- Small sizes (otherwise decoding complexity becomes prohibitive).
- Linear Block codes (BCH codes, Reed Solomon Codes), Convolutional codes...

Index

Modern Coding Theory in a nutshell

- Optimal decoding restrains the codes we can use in practice.
- Our goal is to operate very close to capacity at vanishing error probability. **We need to use very long codes!**

Modern capacity-achieving codes

- Turbo Codes, LDPC codes, Polar Codes.
- Approximate decoding. **Worse than optimal decoding, but much less complex**.
- Approximate decoding complexity: $\mathcal{O}(n)$.
- Codes are optimized so that sub-optimal decoding is enhanced!

A suboptimal decoder for linear block codes over the BEC

Assume the system is already triangularized and reveal as much information you can.

- Hamming $(7,4)$ code.
- $\boldsymbol{x} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ is sent.
- $\boldsymbol{y} = \begin{bmatrix} 1 & ? & 1 & 0 & ? & ? & 0 \end{bmatrix}$ is received.

$$\begin{aligned} x_5 &= 0 \\ x_5 \oplus x_6 &= 0 \\ x_2 \oplus x_6 &= 1 \end{aligned}$$

$$\begin{array}{lcl} x_5 = 0 & & \\ x_5 \oplus x_6 = 0 & \longrightarrow & x_6 = 0 \\ x_2 \oplus x_6 = 1 & \longrightarrow & x_2 = 1 \end{array}$$

The complexity is $\mathcal{O}(n)$!!

- Hamming $(7,4)$ code.
- $\boldsymbol{x} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$ is sent.
- $\boldsymbol{y} = \begin{bmatrix} ? & 1 & ? & 0 & 0 & ? & ? \end{bmatrix}$ is received.

$$
\begin{aligned}
x_1 \oplus x_3 \oplus x_7 &= 0 \\
x_3 \oplus x_6 \oplus x_7 &= 0 \\
x_6 \oplus x_7 &= 0
\end{aligned}
$$

Approximate decoder

There are no equations with a single variable. No information can be revealed.

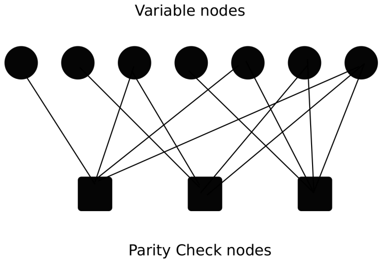
Optimal decoder

x_3 is revealed ($x_3 = 0$) by adding the last two equations.

A message-passing description

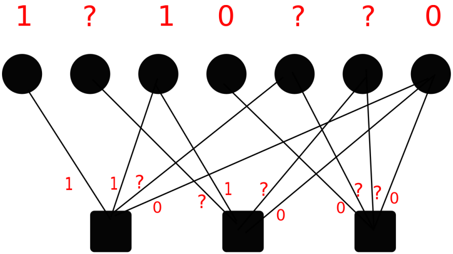
$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Tanner graph of the code:



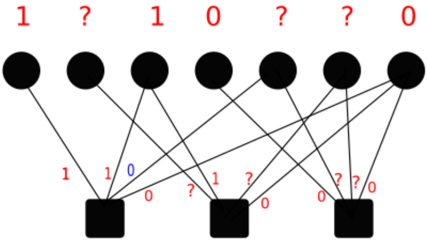
A message-passing description. The BP decoder.

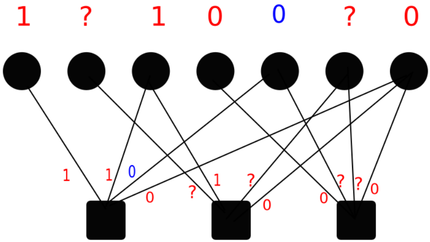
Initialization: Variable nodes send the channel observation to the parity check nodes they are connected:

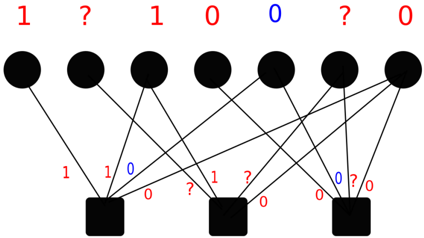


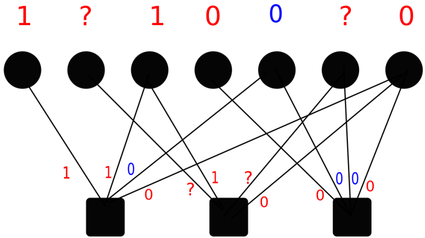
At each iteration:

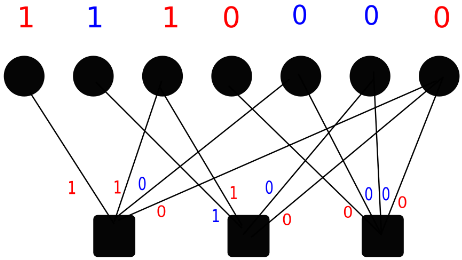
1. Using the received information, each factor tries to resolve the value of the variables that sent a "?" message. If so, they send the value obtained. Otherwise they send a "?" message.
2. **Only factor nodes with a single unknown can resolve a variable!**
3. Variable nodes send their new value or they resend a"?" message.


1 ? 1 0 ? ? 0
1 1 0 0 ? 1 ? 0 0 ? ? 0


1 ? 1 0 0 ? 0
1 1 0 0 ? 1 ? 0 0 ? ? 0


1 ? 1 0 0 ? 0
1 1 0 0
? 1 ? 0
0 0 ? 0


1 ? 1 0 0 ? 0
1 1 0 0
? 1 ? 0
0 0 0 0


1 1 1 0 0 0 0
1 1 0 0
1 1 0 0
0 0 0 0

Sub-optimal decoding for an arbitrary code

- Given a parity check matrix $\boldsymbol{H}$ of dimensions $(n-m)\times n$, the number ones per row can be as high as n.
- If a row has $\lfloor \alpha n \rfloor$ ones, where $\alpha \in (0,1)$, then the probability that $\lfloor \alpha n \rfloor - 1$ of the variables are correctly received and only one is unknown is

$$(\lfloor \alpha n \rfloor)\epsilon(1-\epsilon)^{(\lfloor \alpha n \rfloor - 1)}$$

  which tends to 0 as $\mathtt{n} \to \infty$.
- Matrix $\boldsymbol{H}$ has to be carefully designed!

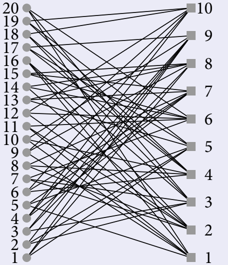
Index

Low-density Parity-Check codes

LDPC codes: linear block codes defined by sparse parity-check matrices.

LDPC (3,6) with $\mathtt{n} = 20$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	1	0	0	0	1
2	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1
4	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1	1	1	0
5	1	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0
6	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0	0	0	1
7	0	0	0	1	1	1	0	1	0	0	0	0	1	0	1	0	0	0	0	0
8	1	0	1	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0
9	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
10	0	0	0	1	0	1	0	0	1	0	0	1	0	0	0	0	0	1	1	0



The density of ones in the matrix $\mathbf{H}$ is $6/\mathtt{n}$ and the rate is $\mathtt{r} = 0.5$.

Suboptimal decoder using $(3,6)$-LDPC codes. Initialization after BEC transmission

- If the number of ones per row is fixed to 6, then the probability that each row in $\mathbf{H}$ has a single unknown is

$$6\epsilon(1-\epsilon)^5$$

which does not vanish with $\mathtt{n}$!

- Let $r_1(0)$ be the fraction of rows in the system of equations with a single unknown.
- Note that $r_1(0) \sim \mathcal{B}(6\epsilon(1-\epsilon)^5, (1-\mathtt{r})\mathtt{n})$ and hence

$$\mathbb{E}[r_1(0)] = \frac{6\epsilon(1-\epsilon)^5(1-\mathtt{r})\mathtt{n}}{(1-\mathtt{r})\mathtt{n}} = 6\epsilon(1-\epsilon)^5(1-\mathtt{r})\mathtt{n}$$

$$\mathrm{Var}[r_1(0)] = \frac{6\epsilon(1-\epsilon)^5(1-6\epsilon(1-\epsilon)^5)(1-\mathtt{r})\mathtt{n}}{(1-\mathtt{r})^2\mathtt{n}^2} = \frac{6\epsilon(1-\epsilon)^5(1-6\epsilon(1-\epsilon)^5)}{(1-\mathtt{r})\mathtt{n}}$$

Density evolution for the $(3, 6)$ ensemble over the BEC

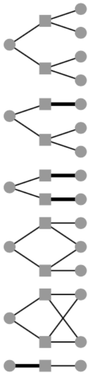
- Assume the code length $\mathtt{n} \to \infty$.
- All-zero codeword.

Asymptotic graph

- In the limit $\mathtt{n} \to \infty$, the graph looks like a tree! (the probability of any cycle of finite length in the code tends to zero).
- Messages received by each node per iteration are independent random variables.
- We can easily compute the asymptotic evolution of x^ℓ as the BP iterates.

Computation graph for the $(2, 4)$ LDPC ensemble

- Computation graph: unroll dependencies for a single variable up to a certain level of deepness.



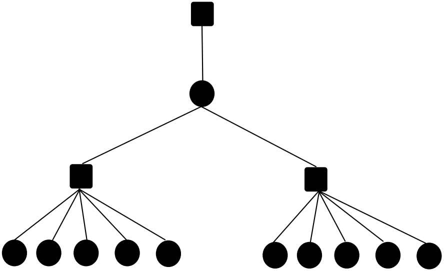
$\frac{(2n-6)(2n-8)}{(2n-1)(2n-5)}$	$\epsilon(1-(1-\epsilon)^2)^2$
$\frac{2(2n-6)}{(2n-1)(2n-5)}$	$\epsilon^2(1-(1-\epsilon)^2)$
$\frac{1}{(2n-1)(2n-5)}$	ϵ^3
$\frac{4(2n-6)}{(2n-1)(2n-5)}$	$\epsilon^2+\epsilon^3(1-\epsilon)$
$\frac{2}{(2n-1)(2n-5)}$	$\epsilon(1-(1-\epsilon)^2)$
$\frac{2}{2n-1}$	ϵ^2

Figure: Possible realizations of the depth-1 computation graph for the $(2, 4)$ LDPC ensemble, together with their probabilities for a graph generated at random and the probability that the root variable is erased after one iteration of the suboptimal decoder.

Density evolution for the $(3,6)$ ensemble over the BEC

- Initialization: variable nodes send an erasure message with probability ϵ.
- Let x^ℓ be the expected probability of an erasure message at iteration ℓ. Thus, $x^0 = \varepsilon$.
- Given $x^{\ell-1}$, using the message-passing update rules for the BEC, we get

$$x^\ell = \varepsilon(1 - (1 - x^{\ell-1})^5)^2$$



$(3,6)$ LDPC ensemble's threshold

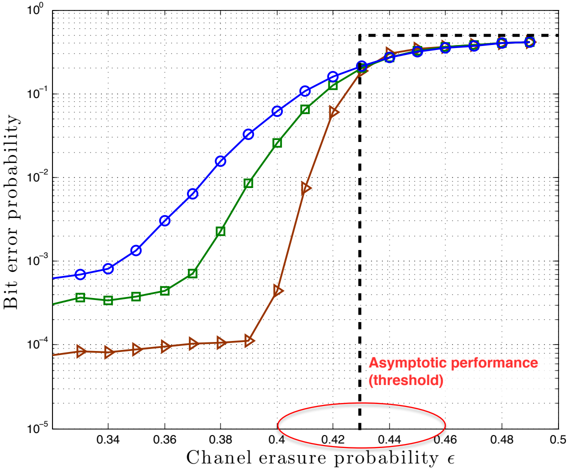
- Using the DE analysis, we can predict if in the asymptotic limit $\mathtt{n} \to \infty$ the BP is able to successfully decode.
- Successful decoding: $\lim_{\ell\to\infty} x^{\ell} = 0$.

$$\lim_{\ell\to\infty} x^{\ell} \to 0 \qquad \varepsilon < 0.4294$$
$$\lim_{\ell\to\infty} x^{\ell} \to \delta \qquad \varepsilon \geq 0.4294$$

where $\delta > 0$.

The $(3,6)$ LDPC ensemble cannot operate close to capacity! The threshold is at 0.4294 while the Shannon limit is at $\epsilon = 0.5$.

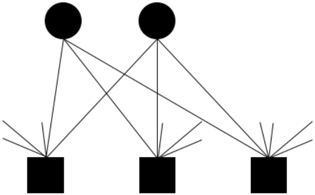
Bit-error rate of the $(3,6)$ ensemble over the BEC $\mathtt{n} = 2^8$ ($\circ$), $\mathtt{n} = 2^9$ ($\square$) and $\mathtt{n} = 2^{11}$ ($\triangleright$).



The threshold ε^* can be computed analytically. **It only depends on the connectivity pattern in the matrix H!**

What about the error floor?

- Error floor is caused by short cycles in the graph.
- As an example, let's analyze the error floor caused by codewords of weight 2 in the (3,6)-LDPC ensemble (generated at random).
- **A weight-2 codeword exists if the LDPC graph contains the following cycle**



- $(3,6)$-LDPC ensemble.
- Code sample $\mathcal{C}$ is chosen randomly with a uniform probability from the ensemble.
- Let N_2 be the number of codewords with Hamming weight 2.

$$N_2 \sim \text{Poisson}(\lambda), \qquad \lambda = \frac{375/9}{\mathtt{n}}$$

- The fraction of codes in the ensemble with $N_2 = 0$ is $\exp^{-\lambda}$.
- The average bit error rate caused by codewords with Hamming weight 2 is

$$P_b \approx 2\epsilon^2 \frac{375/9}{\mathtt{n}}$$

Proof: For a given pair of variables, the probability that they are connected to the same 3 parity check nodes is

$$p_2 = 3\frac{5}{3n-3}\frac{5}{3\mathtt{n}-4}\frac{5}{3\mathtt{n}-5} \approx \frac{375/9}{\mathtt{n}^3}$$

Since there are $\mathtt{n}^2$ pairs of variables, then $\mathbb{E}[N_2] = \frac{375/9}{\mathtt{n}}$. Then, $N_2 \sim \mathcal{B}(p_2, \mathtt{n}^2)$, which tends with $\mathtt{n}$ to $N_2 \sim \text{Poisson}(\frac{375/9}{\mathtt{n}})$. Therefore, $P(N_2 == 0) = \exp(-\lambda)$.

Index

Regular LDPC codes of rate 1/2

l	r	$\mathtt{r}$	ε^*
3	6	1/2	0.43
4	8	1/2	0.38
5	10	1/2	0.34

Increasing the density does not help...

- Improving the solution in the asymptotic limit $\mathtt{n} \to \infty$ by **optimizing the LDPC graph degree distribution (DD).**
- LDPC ensemble: set of codes of length $\mathtt{n}$ that exhibit the same DD in the Tanner graph.
- **Variable perspective DD:**
  - L_i Fraction of variables with degree i.
  - R_j Fraction of check nodes with degree j.
- **Edge perspective DD:**
  - λ_i Fraction of edges with left degree i.
  - ρ_j Fraction of edges with right degree j.

- DD is typically given in polynomial form

$$\lambda(x) = \sum_{i=1}^{l_{\max}} \lambda_i x^{i-1}, \;\; \rho(x) = \sum_{j=1}^{r_{\max}} \rho_j x^{j-1}.$$

- The *design rate* of the code is

$$\mathtt{r} = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}$$

- The density of the matrix

$$\Delta = \frac{1}{\int \lambda(x)} \frac{1}{\mathtt{r}}$$

measures the complexity of the ensemble (edges per information bit).

An example

Consider the DD:

$$\lambda(x) = x, \qquad\qquad \rho(x) = \frac{x^3}{3} + \frac{2\ x^4}{3}$$

- All edges have degree 2 $\Rightarrow$ all variable nodes (n) have degree 2.
- The graph contains $\frac{1}{3}\frac{2n}{4} = \frac{n}{6}$ check nodes of degree 4 and $\frac{2}{3}\frac{2n}{5} = \frac{4n}{15}$ check nodes of degree 5.
- The average check node degree is

$$R_{\text{avg}} = 4\frac{n/6}{n/6 + 4n/15} + 5\frac{4n/15}{n/6 + 4n/15} \approx 4.6153$$

  and can be computed as $\left(\int_0^1 \rho(x)dx\right)^{-1}$.
- The rate of the code is

$$\mathtt{r} = 1 - \frac{\#\text{ rows in } \boldsymbol{H}}{\#\text{ columns in } \boldsymbol{H}} = 1 - \frac{\frac{2n}{R_{\text{avg}}}}{n} \approx 0.5667$$

Density evolution over the BEC

- For any given ensemble DD defined by its degree distribution pair $\lambda(x)$ and $\rho(x)$, we can easily generalized the DE recursion.

$$\textbf{Density Evolution (DE):}\ x_\ell = \epsilon\lambda(1-\rho(1-x_{\ell-1})),$$

Degree distribution LDPC optimization

For a fixed rate $\mathtt{r}$, optimize the coefficients of $(\lambda(x), \rho(x))$ to obtain a threshold close to channel capacity.

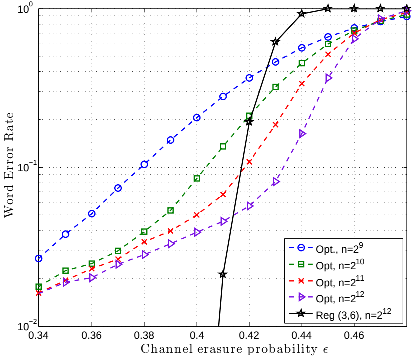
If we fix the maximum left degree to $l_{max} = 100$, the rate $\mathtt{r} = 1/2$ and we set a constant check node degree $\rho(x) = x^{10}$.

$$\begin{aligned}\lambda(x) = {} & 0.169010x + 0.162144x^2 + 0.005938x^4 + 0.016799x^5 + 0.186455x^6 \\ & + 0.006864x^{13} + 0.025890x^{16} + 0.096393x^{18} + 0.010531x^{26} \\ & + 0.004678x^{27} + 0.079616x^{28} + 0.011885x^{38} + 0.224691x^{99}.\end{aligned}$$

- For the BEC, the BP threshold is $\varepsilon^* \approx 0.485$.
- In the AWGN channel, the gap to capacity is only 0.02370 dB.

Irregular LDPC codes in practice. Drawbacks.

- Gap to capacity only vanishes in the limit $l_{max} \to \infty$.
- Irreducible error floor (even for the optimal decoder). Stopping sets!
- **Strong trade-off between performance in the waterfall region and error floor.**



The $(3,6)$ regular code has $\varepsilon^* = 0.4294$ but no error floor....

$$\rho(x) = x^5$$

$$\begin{aligned}\lambda(x) &= 0.416x + 0.166x^2 + 0.1x^3 + 0.07x^4 + 0.053x^5 \\ &+ 0.042x^6 + 0.035x^7 + 0.03x^8 + 0.026x^9 \\ &+ 0.023x^{10} + 0.02x^{11} + 0.0183x^{12},\end{aligned}$$

$$\varepsilon^* = 0.4808, \qquad \mathbf{r} = 1/2$$

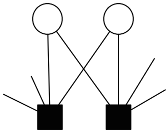


Figure: SS of weight two

.

Index

Capacity-achieving codes

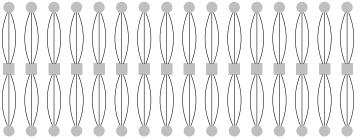
- Polar Codes.
- LDPC Convolutional codes (spatially-coupled LDPC codes).

Irregular LDPC codes theoretically achieve channel capacity but they are very hard to implement in VLSI circuits.

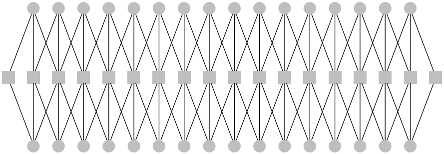
- Most of actual and next generation communication standards only consider quasi-regular LDPC codes.
- Regular LDPC codes can be efficiently implemented (area, energy, throughput,...).

LDPCC based on the $(3,6)$-regular LDPC codes

- L independent $(3,6)$-regular LDPC codes of length M bits are linked in a deterministic way.

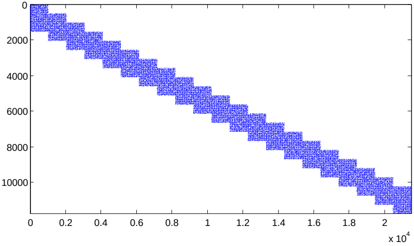


We connect each variable node to a random check node of the code in the left and the code in the right



- L is the chain length. ML is the total codelength.
- For $L \to \infty$ the code looks like a $(3,6)$ ensemble....
- Low-degree check nodes in the boundaries, higher protection!

H matrix for $L = 20$ and $M = 1024$



The rate of the code is

$$\mathtt{r}(3, 6, L) = \frac{1}{2} - \underbrace{\frac{1}{L}}_{\text{rate loss!!}}$$

We need L to be very large!

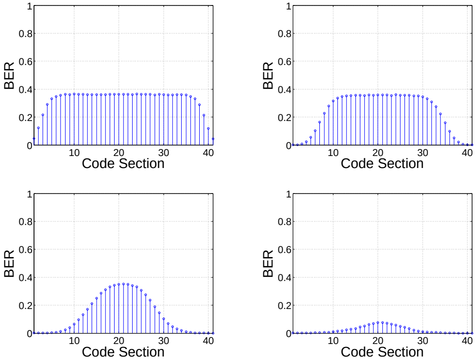
Density Evolution

- DE recursion can be generalized to the convolutional ensemble .
- Surprisingly, the BP threshold $\varepsilon^*(3,6,L)$ outperforms the BP limit for the $(3,6)$ code and tends to the optimal threshold $\varepsilon^{opt}(3,6) \approx 0.4881$.
- Much closer to capacity!
- $\varepsilon^*(5,10,L=50) = 0.499486$**!**

l	r	$\mathtt{r}$	ε^*	$\varepsilon^*(l,r,L)$
3	6	1/2	0.43	0.48815
4	8	1/2	0.38	0.4947
5	10	1/2	0.34	0.4994

Decoding the $(3,6)$ convolutional ensemble

Above the uncoupled $(3,6)$ code threshold, $\varepsilon \geq 0.4294$, the code is peeled off from the boundaries towards the center of the code



Windowed decoding

- LDPCC codes are in general quite large because both M and L are as big as possible.
- One solution to reduce complexity further is to use a fixed sized window (of small size compared to L) to perform decoding.
- This also results in a reduced decoding latency and could be used in latency constrained applications.
- Moreover, spatially coupled codes with $L = \infty$ (convolutional-like codes) can only be decoded through such a windowed decoder.

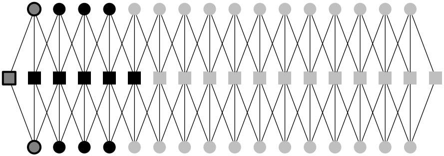


Figure: To decode variables in the first section, we use a subgraph involving sections $1, \ldots, W$

.