E2 205 Error-Control Coding
Lecture 23

Kanishak Vaidya

October 30, 2019

LDPC Codes

Recap

- GDL Conclusion
  - Explaining BP Phase
  - GDL Complexity
  - Decoding Convolution Codes
  - Message Trellis
- LDPC Codes : Explaining the name

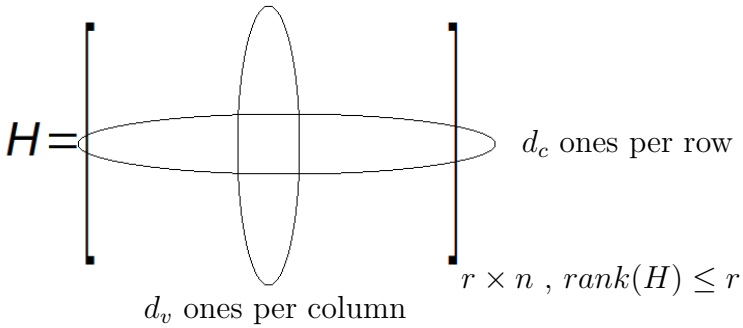
Today

- Rate
- Tanner Graphs
- Alphabets
- Symmetry Assumptions
- Gallager Evolution A : Density Evolution

1 Code Rate

- In a general linear $[n, k, d]$ code $\mathbb{C}$, the rows of parity check matrix are linearly independent.
- But for LDPC code, the rows could be linearly dependent. Although the rows should still span the dual code $\mathbb{C}^{\perp}$ and its null space should be the code $\mathbb{C}$.

Consider $r \times n$ Parity Check matrix H, $r \geq rank(H) = n - k$



- (d_v, d_c)-regular code :
  - d_v 1 in each column
  - d_c 1 in each row

To get bound on rate of the code i.e. k/n , compute number of ones in H first column-wise and then row-wise. We get :

$$nd_v = rd_c$$
$$\therefore \frac{r}{n} = \frac{d_v}{d_c}$$
$$\text{and as } n - k \leq r$$
$$\frac{n-k}{n} \leq \frac{d_v}{d_c} \implies \frac{k}{n} \leq 1 - \frac{d_v}{d_c}$$

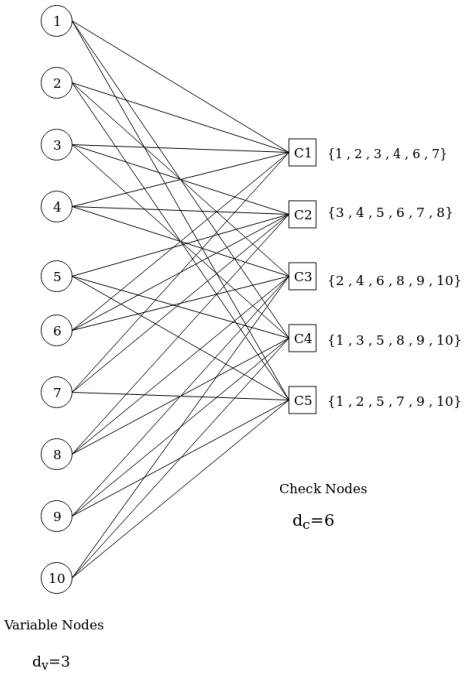
2 Tanner Graph

A graphical way to visualize Parity Check matrix of LDPC Codes.
Consider following Parity Check Matrix:

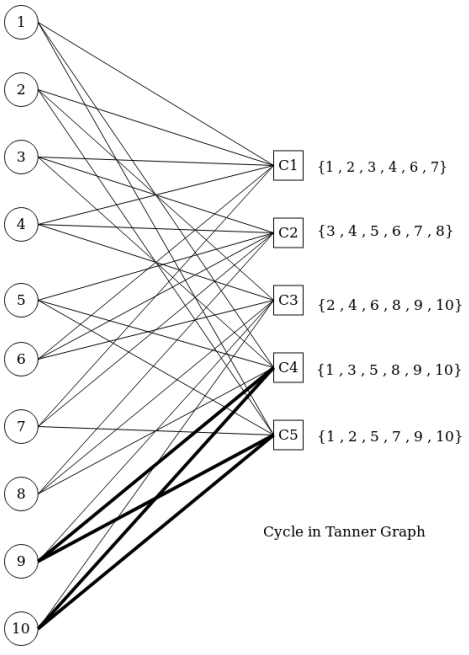
$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

which is $(3, 6)$-regular.

- The Tanner Graph for H will be a bipartite graph with Variable nodes on one side and Check nodes on other, representing columns and rows of H respectively. An edge between i^{th} check node and j^{th} variable node exist if and only if (i, j)th element of H is 1.



Tanner Graph for H



Cycle in a Tanner Graph

3 Definitions

Definition 1 (Path in a Tanner Graph) *A path in a Tanner Graph is a directed sequence* $\vec{e_1}, \vec{e_2} \dots \vec{e_l}$ *of directed edges satisfying* $\vec{e_i} = (u_i, u_i')$, $\vec{e}_{i+1} = (u_{i+1}, u_{i+1}') \implies u_i' = u_{i+1}$.

- Length of a path is number of directed edges along the path.
- Two nodes are at distance d in Tanner Graph if they are connected by a path of length d but not by any path of length less than d.

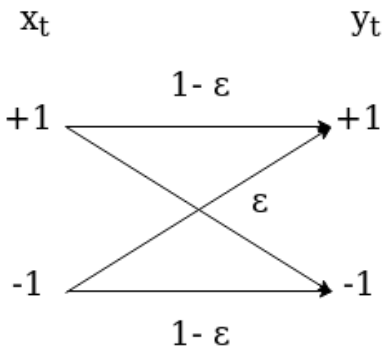
Definition 2 (Neighbourhood) N_u^d *: Neighbourhood of node* u *to depth* d*. The induced subgraph consisting of all edges traversed by paths of length at most* d *and starting from* u*.*

- If $\vec{e} = (v, c)$ then the undirected neighbourhood to depth d of $\vec{e} = N_v^d \cup N_c^d$.
- The directed neighbourhood to depth d of an edge $\vec{e} = (v, c)$ is $N_{\vec{e}}^d$: the induced subgraph containing all edges and nodes on paths $\vec{e_1}, \vec{e_2} \dots \vec{e_d}$ starting from v , $\vec{e_1} \neq \vec{e}$.

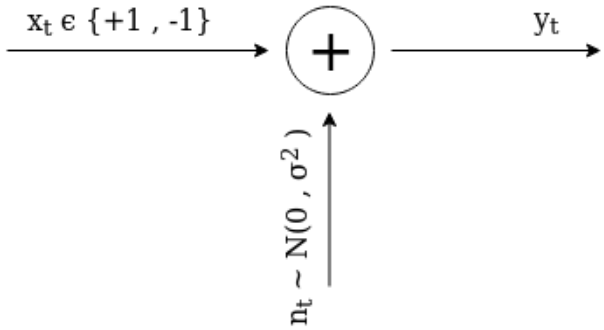
Two Principle Channel Models

Binary Symmetric Channel

- Instead of usual $\{0, 1\}$ input output, it will be convenient to think about input and outputs to be $\{+1, -1\}$.
  $x_t = (-1)^{u_t}, u_t \in \{0, 1\}$. $0 \equiv +1$ & $1 \equiv -1$
- Channel output $y_t = x_t z_t$ where $z_t \in \{+1, -1\}$ and $Pr(z_t = -1) = 1 - Pr(z_t = 1) = \epsilon$
- Channel Symmetry Condition : $p_{Y_t|X_t}(y|x) = p_{Z_t}(\frac{y}{x}) = p_{Y_t|X_t}(-y| - x)$



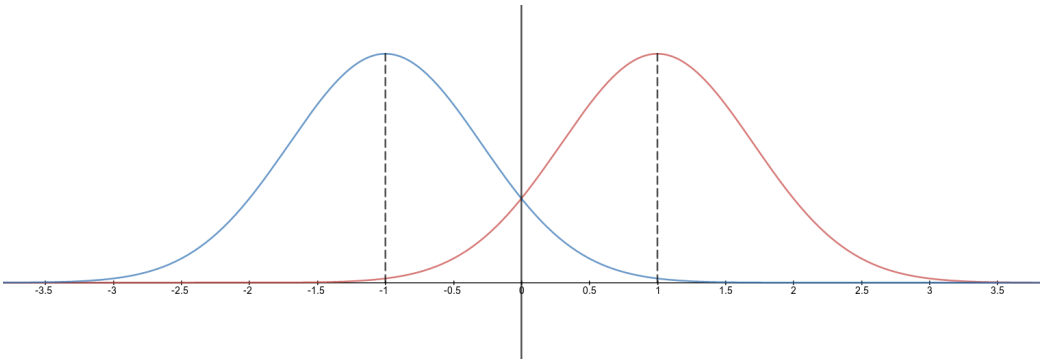
Binary Symmetric Channel



Binary AWGN Channel

Binary Input AWGN Channel

- In this channel $x_t \in \{+1, -1\}$ and $y_t = x_t + n_t$ where $n_t \sim \mathcal{N}(0, \sigma^2)$.
- But the channel can also be viewed as a multiplicative channel where $y_t = x_t z_t$ & $z_t \sim \mathcal{N}(1, \sigma^2)$.
- Channel Symmetry Condition still holds.



Distribution of channel output, given input $x = +1$ (red bell curve) and $x = -1$ (blue bell curve). Note that $f_{Y|X}(y|x = +1) = f_{Y|X}(-y|x = -1)$ as per channel symmetry condition

4 Message Passing

Alphabets

- $\mathcal{O}$: Channel Output Alphabet
- $\mathcal{M}$: The common alphabet employed to pass messages from variable nodes to check nodes and vice versa.
- For discrete case
  - $\mathcal{O} = \{-q_0, -q_0+1, \ldots, 0, 1, \ldots, q_0\}$
  - $\mathcal{M} = \{-q, -q+1, \ldots, 0, 1, \ldots, q\}$
- For continuous case
  - $\mathcal{O} = \mathcal{M} = \mathbb{R}$

Message Passed

- 0^{th} iteration : $\textcircled{v} \longrightarrow \boxed{c}$: Message is passed from variable node to check node. As initially only channel output is present, the function that maps channel output to message passed from variable to check node is $\psi_v^{(0)} : \mathcal{O} \longrightarrow \mathcal{M}$.
- l^{th} iteration : It is completed in 2 steps
  - (i) $\textcircled{v} \longleftarrow \boxed{c}$: Message is passed from check nodes to variable node. Corresponding function is $\psi_c^{(l)} : \mathcal{M}^{d_c-1} \longrightarrow \mathcal{M}$
  - (ii) $\textcircled{v} \longrightarrow \boxed{c}$: Message is passed from variable nodes to check nodes. Corresponding function is $\psi_v^{(l)} : \mathcal{O} \times \mathcal{M}^{d_v-1} \longrightarrow \mathcal{M}$

Aside

Suppose $Y = g(X)$ where X & Y are random variable. Then

$$f_Y(y) = \frac{f_X(g^{-1}(y))}{\frac{dy}{dx}|_{x=g^{-1}(y)}}$$

(g could also be thought of as mapping distribution $g : f_X \to f_Y$)

So assuming $\Pi_{\mathcal{O}}$ as density function over alphabet $\mathcal{O}$ and $\Pi_{\mathcal{M}}$ as density over $\mathcal{M}$ we have

- $\psi_v^{(0)} : \Pi_{\mathcal{O}} \to \Pi_{\mathcal{M}}$
- $\psi_v^{(l)} : \Pi_{\mathcal{O}\times\mathcal{M}^{d_v-1}} \to \Pi_{\mathcal{M}}$, and because of independence as $\Pi_{\mathcal{O}} \times \Pi_{\mathcal{M}}^{d_v-1} \to \Pi_{\mathcal{M}}$
- Similarly $\psi_c^{(l)} : \Pi_{\mathcal{M}}^{d_v-1} \to \Pi_{\mathcal{M}}$

Symmetry Assumptions Pertaining to Message Passing

- At a variable node we assume

$$\psi_v^{(0)}(bm) = b\psi_v^{(0)}(m), b \in \{+1, -1\}$$

$$\psi_v^{(l)}(bm_0, bm_1 \dots bm_{d_v-1}) = b\psi_v^{(0)}(m_0, m_1 \dots m_{d_v-1}), b \in \{+1, -1\}$$

- At a check node

$$\psi_c^{(l)}(b_1m_1, b_2m_2 \dots b_{d_c-1}m_{d_c-1}) = \left(\prod_{j=1}^{d_c-1} b_j\right) \psi_c^{(l)}(m_1, m_2 \dots m_{d_c-1}),$$

$$b_j \in \{+1, -1\} \forall j$$

Performance Evaluation

- We will evaluate performance by carrying out density evolution i.e. estimate the number of incorrect messages passed during each iteration.
- We will assume that $\underline{1}$ codeword was transmitted.

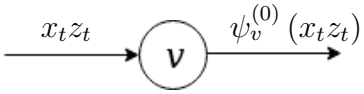
Claim *Under channel symmetry assumption X_t and Z_t are independent.*
Proof Let $Z \equiv Z_t$, $Z \equiv X_t$, $Y \equiv Y_t$.

$$p_{Z|X}(z|x) = p_{Y|X}(y|x) = p_{Y|X}(xz|x) = p_{Y|X}(-xz|-x) = p_{Z|X}(z|-x)$$

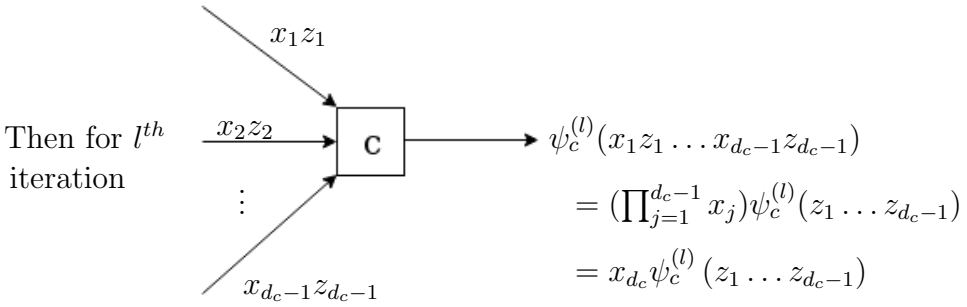
Distribution of Z is same for X and $-X$ (and $X \in \{+1, -1\}$)
$\therefore X$ and Z are independent

Claim *Under channel symmetry and message passing assumptions the expected number of incorrect messages passed on from the variable nodes to check nodes is same on every iteration.*

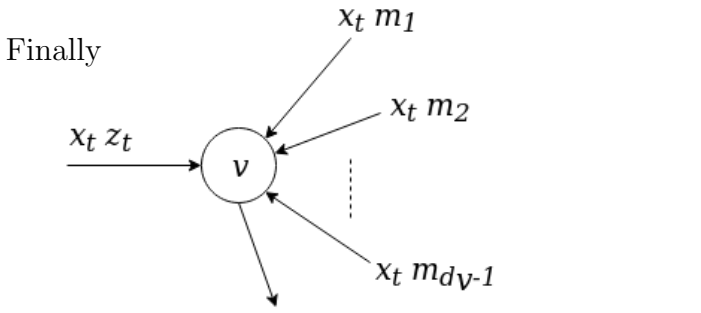
Proof Let $(X_1 \dots X_n), X_j \in \{+1, -1\}$ be a codeword in LDPC code.



Initial map. Here $\psi_v^{(0)}(x_t z_t) = x_t \psi_v^{(0)}(z_t)$



$\prod_{j=1}^{d_c-1} x_j = x_{d_c}$($\because$ messages satisfy parity check)
So if incoming messages are multiplied by corresponding variable, then output variable is also multiplied by corresponding variable.



Now, as the messages are always x_t times whatever messages would have been had the corresponding $x_t = 1$. So we are actually comparing true value of x_t to sign of $x_t \psi_v^{(l)}(z_t, m_1, m_2 \dots m_{d_v-1})$, which again is same as checking sign of $\psi_v^{(l)}(z_t, m_1, m_2 \dots m_{d_v-1})$. Thus the expected number of incorrect

messages is independent of transmitted codeword and hence, is same on every iteration.