CT111 Intro to Communication Systems
Lecture 13: LDPC Codes

Yash M. Vasavada

Associate Professor, DA-IICT, Gandhinagar

26th March 2018

Overview of Today's Talk

1. Tanner Graphs
2. LDPC Codes
3. EXIT Charts

Overview of Today's Talk

Overview of Today's Talk

The Parity Check Equations

Matrix Version

- The parity check equations can be written as a matrix product:

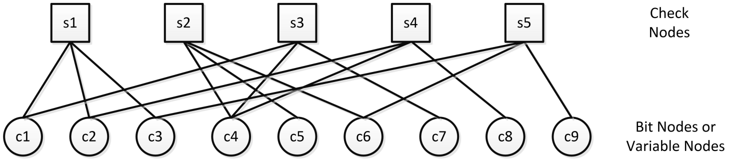
$$\mathbf{H}_{(N-K)\times N} \cdot \mathbf{c}_{N\times 1} = \mathbf{0}_{N-K\times 1}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

▷ Above is the parity check matrix equation for $(9, 4)$ product code

Tanner Graph

- This parity check matrix **H** can be represented by a Tanner Graph, which is a Bipartite Graph:
  - This has two types of nodes:
    1. A total of $N - K$ check nodes, and
    2. A total of N coded bit nodes
  - Nodes of one type are never connected to the nodes of the same type. Resulting binary partition of the nodes of graph is the reason why this is called Bipartite Graph



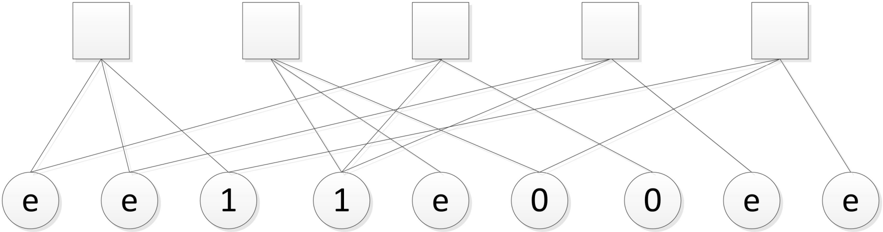
Decoding on Tanner Graph

Decoding can be better visualized and implemented using the Tanner graph framework:

1. Load the variable nodes with the observed code bits.
2. Each check node $1 \leq j \leq N$ sends a message to each of its connected variable nodes i.
   - ▷ The message is the modulo two sum of the bits associated with the connected variable nodes other than node i (if none are erased).
   - ▷ If a check node touches a single erasure, then it will become corrected.
3. Iterate Step 2 until all erasures corrected or no more corrections possible.

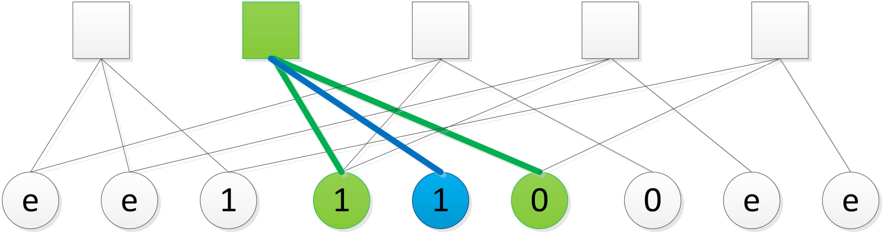
Decoding on Tanner Graph

Step 1: Load the Variable Nodes with the Received Codeword Bits



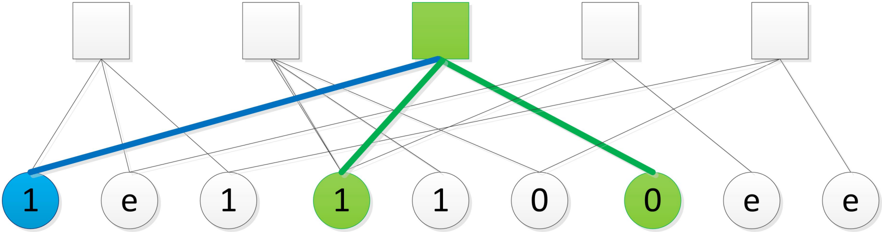
Decoding on Tanner Graph

Step 2: Iteration 1



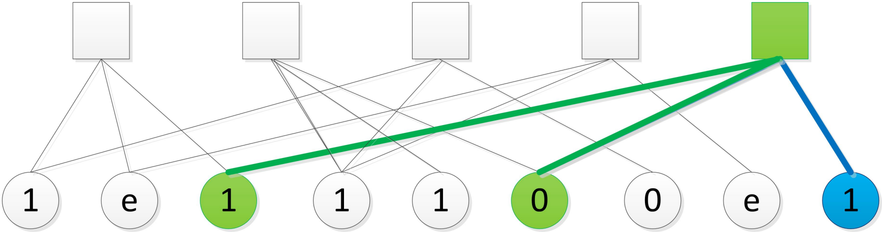
Decoding on Tanner Graph

Step 2: Iteration 2



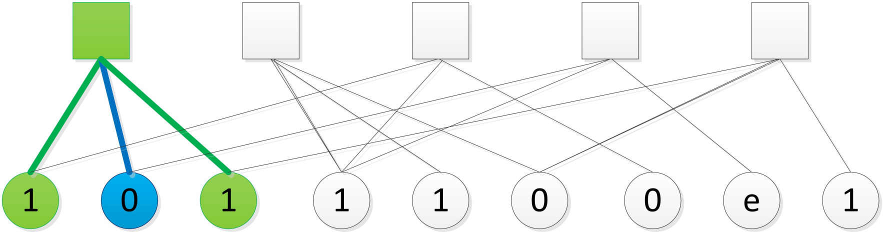
Decoding on Tanner Graph

Step 2: Iteration 3



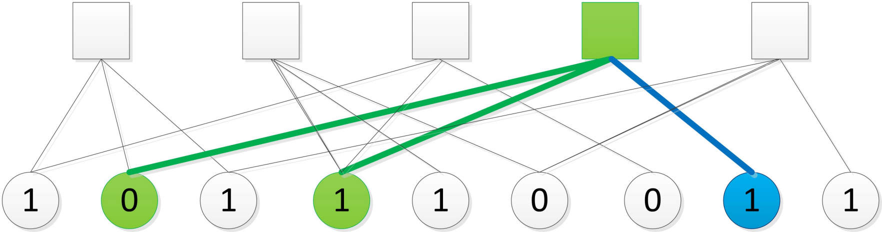
Decoding on Tanner Graph

Step 2: Iteration 4



Decoding on Tanner Graph

Step 2: Iteration 5



Observations

- The decoders complexity depends on the degree of the check nodes.
- The degree of a check node is equal to the Hamming weight of the corresponding row of the parity-check matrix.
- To achieve capacity, a long code is needed.
- Thus, it is desirable to have a code that is long, yet with small row weight.

Low Density Parity Check (LDPC) Codes

- An LDPC code is characterized by a sparse parity-check matrix $\mathbf{H}$.
- The row/column weights of this $\mathbf{H}$ are independent of length N of the codeword.
- Accordingly, the decoder complexity grows only linearly with block length.
- The LDPC codes were first proposed and developed by Robert Gallager in his 1960 PhD dissertation.
- Were forgotten because the decoder could not be implemented.
- Were rediscovered in the mid-1990s by MacKay and Neal after turbo codes were developed.

Low Density Parity Check (LDPC) Codes

An example LDPC code parity check matrix from MacKay and Neal (1996):

$$\mathbf{H} = \left[\begin{array}{cccc|cccc|cccc}
1 & & & & & 1 & & 1 & & 1 & & \\
1 & & & 1 & 1 & & & & & & 1 & \\
 & 1 & & & 1 & & 1 & & 1 & & & \\
 & & 1 & & & 1 & & & & & 1 & 1 \\
 & & 1 & & & & 1 & 1 & & & & 1 \\
 & 1 & & & 1 & & & & 1 & & 1 & \\
1 & & & 1 & & & 1 & & & 1 & & \\
 & 1 & & & & 1 & & 1 & & 1 & & \\
 & & 1 & 1 & & & & & 1 & & & 1
\end{array}\right]$$

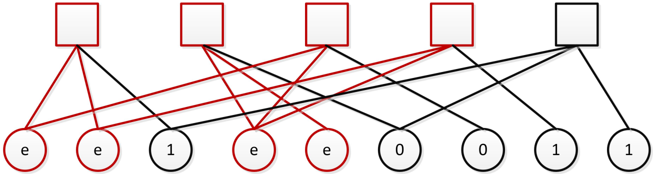
- Called a $(w_c, w_r) = (3, 4)$ regular code.
- For the regular codes,
  - rows of **H** have a constant Hamming weight of w_r; alternatively, the degree of the check nodes d_c is a constant
  - columns of **H** have a constant Hamming weight of w_c; alternatively, the degree of the variable nodes d_v is a constant

Appendix
For Information Only

- Remaining part is **not** in CT111 syllabus
- Included here for those of you interested

Stopping Sets

- A stopping set $\mathcal{V}$ is a set of erased variable nodes that cannot be corrected, regardless of the states of the other variable nodes



- The minimum stopping set $\mathcal{V}_{min}$ is the stopping set containing the fewest variable nodes
- Let $d_{min} = |\mathcal{V}_{min}|$ be the size of the minimum stopping set
  - ▷ There is at least one pattern with d_{min} erasures that cannot be corrected
  - ▷ The erasure correction capability of the code is $d_{min} - 1$

Decoding Probability
Check to Variable Node Message

- Decoding involves the exchange of messages between variable nodes and check nodes.
- Consider the degree d_c check node.
  - → An outgoing message sent over an edge of the Tanner graph (by a particular check node toward a particular variable node) is a function of the incoming messages arriving over the other $d_c - 1$ edges.
  - → For the outgoing message to be correct, all $d_c 1$ incoming messages must be correct. Probability of this is $(1 - e_\ell)^{d_c - 1}$.
    - ▷ e_ℓ is the probability that the message going from the variable node to the check node is an erasure at ℓ^{th} iteration
    - ▷ $e_0 = e$ (recall, e is the original BEC-induced erasure probability)
  - → Therefore, the probability of the check node sending an erasure is

$$e_\ell^c = 1 - (1 - e_\ell)^{d_c - 1}. \quad (1)$$

Decoding Probability
Variable to Check Node Message

- Next consider the degree d_v variable node.
  - → An outgoing message sent by a particular variable over a particular edge is a function of the incoming messages arriving over the other $d_v - 1$ edges.
  - → For the outgoing message to be an erasure at ℓ^{th} iteration, the variable node has to be erased over the channel (this occurs with a probability of $e_0 = e$, *and* all $d_v - 1$ messages have all to be erased at ℓ^{th} iteration. This occurs with a probability of

$$e_{\ell+1} = e_0 (e_\ell^c)^{d_v - 1}. \tag{2}$$

- Combining Equations 1 and 2, we obtain

$$e_{\ell+1} = e_0 \left(1 - (1 - e_\ell)^{d_c - 1}\right)^{d_v - 1}. \tag{3}$$

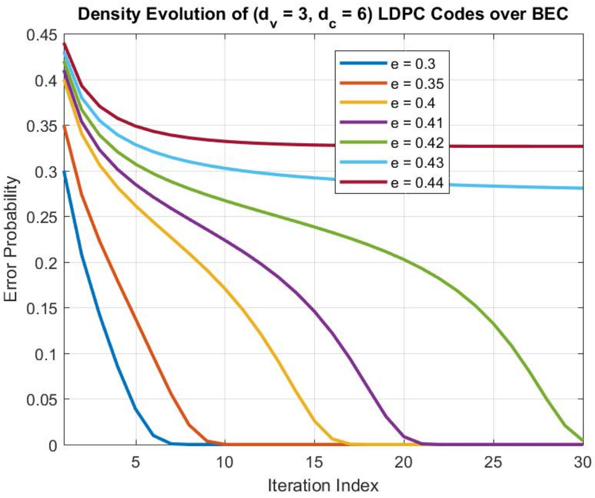
Decoding Probability

using Density Evolution Method

- The method described earlier is called the Density Evolution (DE) method of evaluating the performance of regular LDPC codes
- This DE method assumes independent messages, which is achieved when the girth (i.e., length of the shortest cycle) of the Tanner graph is sufficiently large.
- If $e_\ell \to 0$ as $\ell \to \infty$ for a particular channel erasure probability e, a code drawn from the ensemble of all such (d_v, d_c) regular LDPC codes will be able to correctly decode.
- The threshold e^* is the maximum e for which $e_\ell \to 0$ as $\ell \to \infty$.
- For the $(d_v = 3, d_c = 6)$ regular code, the threshold e^* is 0.4294.

Decoding on Tanner Graph

Step 2: Iteration 5



Summary of Density Evolution Method

- Density Evolution describes only an asymptotic performance of the ensemble of LDPC codes.
- Implementation requires that an matrix be generated by drawing from the ensemble of all (d_v, d_c) LDPC codes.
- Goals of good **H** design:
  1. High girth (a small value of girth, i.e., short cycles, can cause loss of independent information to be fed back in the iterative decoder).
  2. Full rank (if **H** is not full rank, then the rate of the LDPC code reduces below the desired rate)
  3. Large minimum stopping set (small stopping sets give rise to an error floor).
- A database of good regular LDPC codes is available on David MacKay's website

Extrinsic Information Transfer (EXIT) Chart

for Convergence Analysis of LDPC codes

- An observation about the LDPC codes:
  - ▷ Check nodes are essentially the SPC codes
  - ▷ Variable nodes are essentially the repetition codes
- How do they work in tandem?

Extrinsic Information Transfer (EXIT) Chart

for Convergence Analysis of LDPC codes

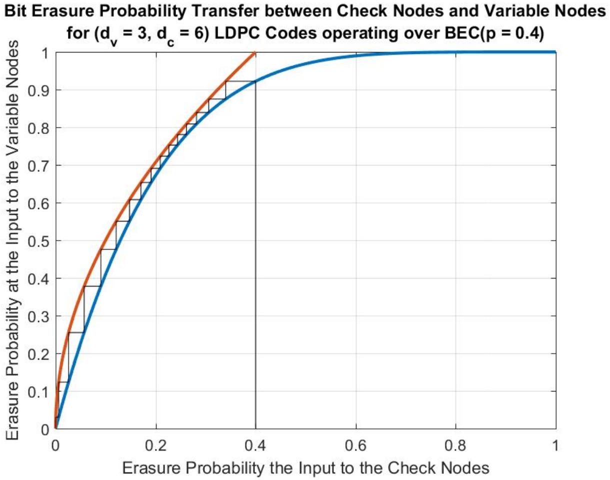
- How do these two types of codes operate in tandem?
- Let p_{SPC}^{in} and p_{SPC}^{out} denote the BEC error probabilities at the input and the output to the SPC codes (i.e., the check nodes), and
- Let p_{RC}^{in} and p_{RC}^{out} denote the BEC error probabilities at the input and the output to the repetition codes (i.e., the variable nodes)
- In LDPC codes, $p_{SPC}^{in} = p_{RC}^{out}$, and $p_{RC}^{in} = p_{SPC}^{out}$
- Following holds:

$$p_{SPC}^{out} = 1 - \left(1 - p_{SPC}^{in}\right)^{d_c - 1}$$

$$p_{RC}^{out} = e_0 \left(p_{RC}^{in}\right)^{d_v - 1}$$

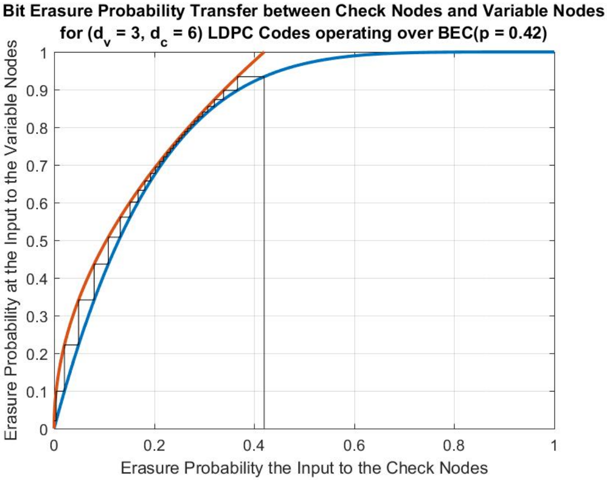
Erasure Probability Transfer Chart
for Convergence Analysis of LDPC codes

- Channel erasure probability $e_0 = 0.4$



Erasure Probability Transfer Chart
for Convergence Analysis of LDPC codes

- Channel erasure probability $e_0 = 0.42$



Extrinsic Information Transfer (EXIT) Chart

for Convergence Analysis of LDPC codes

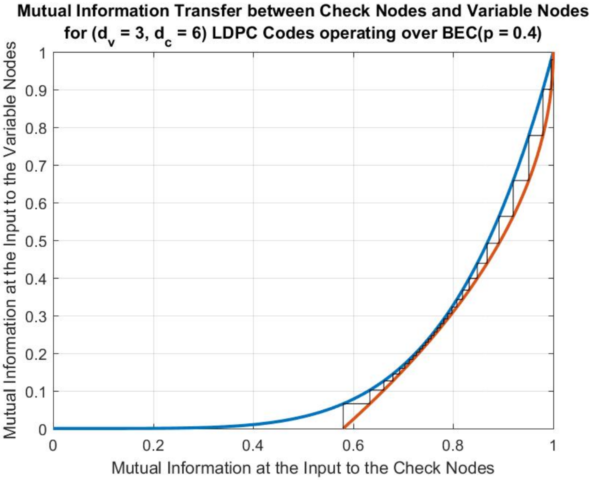
- Recall that the mutual information (the channel capacity) for the BEC(e) with erasure probability of e is given as $C = I = 1 - e$.
- The equalities in the prior chart can be translated to the mutual information I.

$$I_{SPC}^{out} = {I_{SPC}^{in}}^{d_c - 1}$$

$$I_{RC}^{out} = e_0 \left(1 - I_{RC}^{in}\right)^{d_v - 1}$$

EXIT Chart
for Convergence Analysis of LDPC codes

- Channel erasure probability $e_0 = 0.4$



EXIT Chart
for Convergence Analysis of LDPC codes

- Channel erasure probability $e_0 = 0.42$