Chap. 5 Density Evolution and Gaussian Approximation

References:

- T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transations on Information Theory*, Feb. 2001.
- S.-Y. Chung, T. J. Richardson and R. L. Urbanke, "Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using a Gaussian Approximation," *IEEE Transations on Information Theory*, Feb. 2001.

• Gallager (1961)

Proposed the following *step-by-step program* to determine the *worst binary-symmetric channel* (BSC), i.e., the BSC with the largest crossover probability, over which an appropriately constructed (d_v, d_c)- regular LDPC code in conjunction with a given iterative decoding algorithm can be used to transmit information reliably.

1) *Code construction*: For increasing length n, construct a sequence of (d_v, d_c)-regular LDPC codes that do not contain cycles of length $\leq 2\,l(n)$, where

$$l(n) \triangleq \frac{\ln n - \ln \frac{d_v d_c - d_v - d_c}{2 d_c}}{\ln\left[(d_c - 1)(d_v - 1)\right]}.$$

2) *Density Evolution and Threshold Determination*

- $P_e(l) =$ Average fraction of incorrect messages passed at the lth iteration assuming that the graph does not contain cycles of length $2l$ or less.
- Threshold σ^*: $\forall \sigma < \sigma^*,\ \lim_{l\to\infty} P_e(l) = 0$
  (or *maximum channel parameter*)

- **T. J. Richardson and R. Urbanke**
  (The capacity of low-density parity-check codes under message-passing decoding, *IEEE Trans. Inform. Theory*, vol. 47, pp. 599-618, Feb. 2001.)

1) **Concentration**:

Let $P_e^n(l)$ be the *expected fraction of incorrect messages* passed in the lth iteration where the expectation is over all instances of the code, the choice of the message, and the realization of the noise.

Then, for any typical code $\mathcal{C}$ and any $\delta > 0$,

$$\Pr\left\{\left|P_{e,\mathcal{C}}^n(l) - P_e^n(l)\right| > \delta\right\} \xrightarrow[\text{(exponentially)}]{n \to \infty} 0.$$

2) **Convergence to cycle-free case**:

Let $P_e^\infty(l)$ be the expected fraction of incorrect messages passed in the lth decoding round assuming that the graph does not contain cycles of length $2l$ ore less. Then

$$\lim_{n\to\infty} P_e^n(l) = P_e^\infty(l).$$

3) **Density Evolution and Threshold Determination**:

- $P_e^\infty(l)$ is computable by a deterministic algorithm.
- There exists a channel parameter σ^* ($\simeq 0.88$ in BIAWGNC) such that

$$\lim_{l\to\infty} P_e^\infty(l) = \begin{cases} 0 & \text{if } \sigma < \sigma^* \\ \gamma(\sigma) > 0 & \text{if } \sigma > \sigma^*. \end{cases}$$

Remark:

1) Concentration $\Rightarrow$ *(Almost) all codes behave alike* and so the determination of the average behavior of the ensemble suffices to characterize the individual behavior of (almost) all codes

2) *For long codes this average behavior is equal to the behavior of cycle-free graphs.*

   For the cycle-free case this average behavior is computable by a deterministic algorithm.

3) *Long codes will exhibit a threshold phenomenon*, clearly separating the region where reliable transmission is possible from that where it is not.

Conjecture: If $\sigma > 0.88$, all codes in the (3.6)-regular LDPC ensemble have bit-error probability of at least 0.05 regardless of their length and regardless of how many iterations are performed.

- (d_v, d_c)-**regular LDPC codes**
  - Bipartite graph with n variable nodes and $m \triangleq \frac{nd_v}{d_c}$ check nodes
  - Parameters:
    d_v = degree of a variable node
    d_c = degree of a check node
  - Design rate:

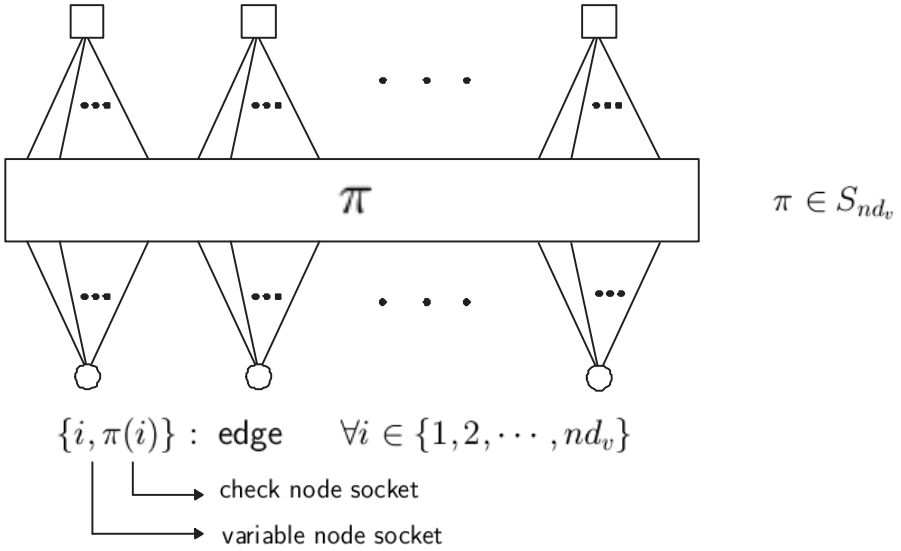
$$r \triangleq \frac{n-m}{n} = 1 - \frac{d_v}{d_c} \leq \text{Actual rate}$$

- **Ensemble of (d_v, d_c)-regular LDPC codes**

Let

$$\begin{aligned} C^n(d_v, d_c) &= \text{the set of labeled bipartite graphs} \\ S_{nd_v} &= \text{the set of all permutations on } \{1, 2, \ldots, nd_v\} \end{aligned}$$

Factor graph



- **Some Definitions and Notation**
  - $e = \{v, c\}$: edge between variable node v and check node c
  - $\overrightarrow{e}$: directed edge (v, c) or (c, v)
  - Path: a directed sequence of directed edges $\overrightarrow{e_1}, \cdots, \overrightarrow{e_k}$ such that if $\overrightarrow{e_i} = (u_i, u_i')$ then $u_i' = u_{i+1}$ for all $i = 1, \cdots, k-1$.
  - Length of the path = number of directed edges in it.
  - Two nodes have distance $d \leq \infty$ if they are connected by a path of length d but not by a path of length $< d$. If the nodes are not connected, then $d = \infty$.
  - $\mathcal{N}_u^d \triangleq$ the neighborhood of u of depth d.
    $=$ the induced subgraph consisting of all nodes reached and edges traversed by paths of length at most d starting from u (including u)

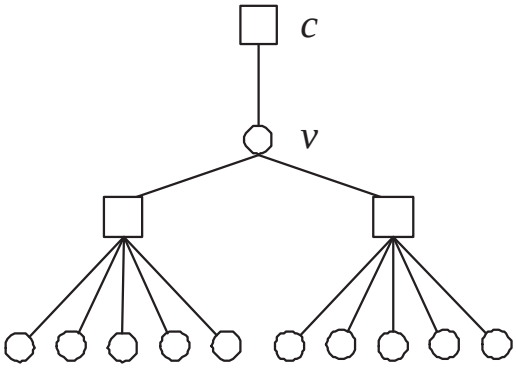
    Note that $u_1 \in \mathcal{N}_{u_2}^d \Leftrightarrow u_2 \in \mathcal{N}_{u_1}^d$ (by symmetry of the distance function)

    Exercise: Prove that $u_1 \in \mathcal{N}_{u_2}^d \Leftrightarrow u_2 \in \mathcal{N}_{u_1}^d$.
  - For an edge $e = \{v, c\}$,
    $$\mathcal{N}_e^d = \text{the undirected neighborhood of } e \text{ of depth } d \triangleq \mathcal{N}_v^d \cup \mathcal{N}_c^d$$
    Note that $e \in \mathcal{N}_{e'}^d \Leftrightarrow e' \in \mathcal{N}_e^d$

    Exercise: Prove that $e \in \mathcal{N}_{e'}^d \Leftrightarrow e' \in \mathcal{N}_e^d$.

– $\mathcal{N}_{\overrightarrow{e}}^{d}$ = the directed neighborhood of depth d of $\overrightarrow{e}(c, v)$

Note that $\mathcal{N}_{\overrightarrow{e}}^{d}$ is the induced subgraph containing all edges and nodes on paths starting from v such that $\overrightarrow{e_1} \neq \overrightarrow{e}$.



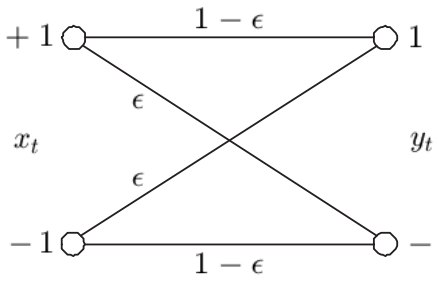
The neighborhood is *tree-like* iff all involved nodes are distinct.

• Binary-Input Memoryless Channels

1) Binary Symmetric Channels (BSCs):

$$y_t \;=\; (-1)^{w_t}\, x_t$$

where w_t is a sequence of i.i.d. Bernoulli RVs with $\Pr\{w_t = 0\} = 1 - \epsilon$.



$$C_{\mathsf{BSC}}(\epsilon) = 1 - h(\epsilon)$$

$$h(\;) = -x\log_2 x - (1-x)\log_2(1-x)$$

2) Continuous additive Gaussian channels:

$$y_t = x_t + z_t,$$

where $x_t \in \{\pm 1\}$ and z_t is an i.i.d. random variable with pdf $p(z)$

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-z^2/2\sigma^2}$$

The channel capacity is given by

$$C_{\mathsf{BIAWGN}}(\sigma) = -\int \phi_\sigma(x)\log_2\phi_\sigma(x)dx - \frac{1}{2}\log_2(2\pi e\sigma^2)$$

where

$$\phi_\sigma(x) = \frac{1}{\sqrt{8\pi\sigma^2}}\left(e^{-\frac{(x+1)^2}{2\sigma^2}} + e^{-\frac{(x-1)^2}{2\sigma^2}}\right).$$

3) Continuous additive Laplace channels:

$$p(z) \;=\; \frac{1}{2\lambda}\, e^{-\frac{|z|}{\lambda}}$$

$$C_{\mathsf{BIL}}(\lambda) \;=\; \frac{-\pi + 4\arctan(e^{-1/\lambda})}{2e^{1/\lambda}\log_e 2} \;-\; \log_2\left(\frac{1+e^{-2/\lambda}}{2}\right)$$

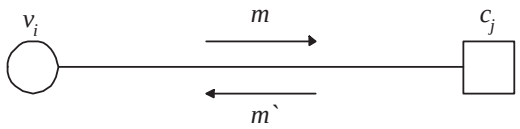
Exercise: Derive $C_{\mathsf{BSC}}(\epsilon)$, $C_{\mathsf{BIAWGN}}(\sigma)$, and $C_{\mathsf{BIL}}(\lambda)$.

• Message-Passing Decoders

1) Let

$\mathcal{O}$: channel output alphabet (= decoder input alphabet)

$\mathcal{M}$: message alphabet ($\mathcal{O} \subset \mathcal{M}$)



Condition on the processing: A message sent from a node u along an adjacent edge e may not depend on the message previously received along edge e.
"Only extrinsic information is passed along."

2) **Message map**

– Variable node message map $\Psi_v^{(l)}$

$$\Psi_v^{(l)} : \mathcal{O} \times \mathcal{M}^{d_v-1} \longrightarrow \mathcal{M}, \quad l \geq 1$$

– Check node message map $\Psi_c^{(l)}$

$$\Psi_c^{(l)} : \mathcal{M}^{d_c-1} \longrightarrow \mathcal{M}, \quad l \geq 0$$

3) **Distribution map**

– Let $\Pi_A \triangleq$ the space of probability distributions defined over A

– Distribution map ${}^*\Psi_v^{(l)}$ at variable node v

$${}^*\Psi_v^{(l)} : \Pi_{\mathcal{O}} \times \Pi_{\mathcal{M}}^{d_v-1} \longrightarrow \Pi_{\mathcal{M}}$$

If m_0: a RV with $P_0 \in \Pi_{\mathcal{O}}$;
$m_i,\ i = 1, \cdots, d_v - 1$: RVs with $P \in \Pi_{\mathcal{M}}$; and
$m_0, m_1, \cdots, m_{d_v-1}$ are independent,
then the distribution of $\Psi_v^{(l)}(m_0, \cdots, m_{d_v-1})$ is ${}^*\Psi_v^{(l)}(P_0, P, \cdots, P)$.

– Distribution map ${}^*\Psi_c$ at check node c

$${}^*\Psi_c^{(l)} : \Pi_{\mathcal{M}}^{d_c-1} \longrightarrow \Pi_{\mathcal{M}}$$

- **Symmetry Assumptions: Restriction to All-One Codeword**
  - Alphabet
    1) Discrete case:

$$\begin{aligned}\mathcal{O} &\triangleq \{-q_0, -(q_0-1), \cdots, -1, 0, 1, \cdots, (q_0-1), q_0\} \\ \mathcal{M} &\triangleq \{-q, -(q-1), \cdots, -1, 0, 1, \cdots, (q-1), q\}\end{aligned}$$

    2) Continuous case

$$\mathcal{O} = \mathcal{M} = \mathbb{R} \qquad \text{or} \qquad \mathcal{O} = \mathcal{M} = [-\infty, +\infty]$$

  - Sign and Absolute Value of the Message
    1) The sign of the message indicates whether the transmitted bit is estimated to be $+1$ or -1; and
    2) The absolute value of the message is a measure of the *reliability of this estimate*.

Definition (Symmetry Conditions)

1) *Channel Symmetry*: The channel is output-symmetric, i.e.,

$$p(y_t = q \,|\, x_t = 1) \;=\; p(y_t = -q | x_t = -1).$$

2) *Check node symmetry*: Signs factor out of check node message maps

$$\Psi_c^{(l)}(b_1 m_1, \cdots, b_{d_c-1} m_{d_c-1}) = \Psi_c^{(l)}(m_1, \cdots, m_{d_c-1}) \cdot \prod_{i=1}^{d_c-1} b_i$$

   for any $\pm$ sequence $(b_1, b_2, \ldots, b_{d_c-1})$.

3) *Variable node symmetry*: Sign inversion invariance of variable node message maps holds

$$\begin{aligned}\Psi_v^{(l)}(-m_0, -m_1, \cdots, -m_{d_v-1}) &= -\Psi_v^{(l)}(m_0, m_1, \cdots, m_{d_v-1}), \quad l \geq 1 \\ \text{and} \quad \Psi_v^{(0)}(-m_0) &= -\Psi_v^{(0)}(m_0).\end{aligned}$$

Lemma 1 (Conditional Independence of Error Probability under Symmetry)
Let $\mathcal{G}$ be the bipartite graph representing a given binary linear code (not necessarily an LDPC code) and for a given message-passing algorithm let $P_e^{(l)}(\mathbf{x})$ be the conditional (bit or block) probability of error after the lth decoding iteration, assuming that the codeword $\mathbf{x}$ was sent. If the channel and the decoder fulfill the symmetry conditions, then $P_e^{(l)}(\mathbf{x})$ is independent of $\mathbf{x}$.

Proof)

- Let $p(q) = p(y = q \,|\, x = +1)$ be the channel transition probability.
- Any binary-input memoryless output-symmetric channel can be modeled multiplicatively as $y_t = x_t z_t$ where x_t is the input bit, y_t is the channel output, and z_t are i.i.d random variables with distribution $\Pr(z_t = q) = p(q)$.
- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, an arbitrary codeword
  $\mathbf{z} = (z_1, z_2, \ldots, z_n)$, the channel realization
  $\mathbf{y} = \mathbf{xz}$, an observation from the channel,
  where multiplication is componentwise.
- $m_{ij}^{(l)}(\mathbf{w}) \triangleq$ message sent from variable node v_i to check node c_j in iteration l assuming that $\mathbf{w}$ was received.

  $m_{ji}^{(l)}(\mathbf{w}) \triangleq$ message sent from c_j to v_i assuming that $\mathbf{w}$ was received.
- Use induction.

  At $l = 0$, $m_{ij}^{(0)}(\mathbf{y}) = x_i m_{ij}^{(0)}(\mathbf{z})$ from the variable node symmetry.

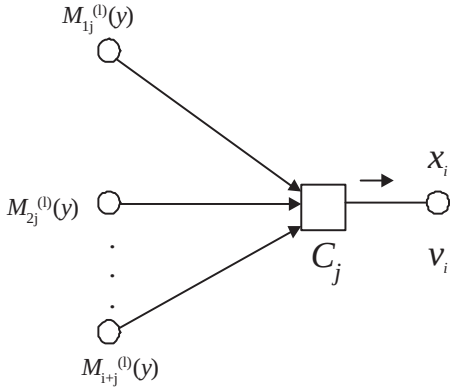
  Assume that $m_{ij}^{(l)}(\mathbf{y}) = x_i m_{ij}^{(l)}(\mathbf{z})$ in iteration l.

  Since x is a codeword, $\prod_{k:\, \exists_{e=(v_k, c_j)}} x_k = 1$.

  From the check node symmetry,

$$m_{ji}^{(l+1)}(\mathbf{y}) \;=\; x_i \, m_{ji}^{(l+1)}(\mathbf{z}).$$

Exercise: Show this.
(Hint: First, show $m_{ji}^{(l+1)}(\mathbf{y}) = \left(\prod_{t=1}^{i-1} x_t\right) m_{ji}^{(l+1)}(\mathbf{z})$ and use $\left(\prod_{t=1}^{i-1} x_t\right) = x_i$ in the following figure)

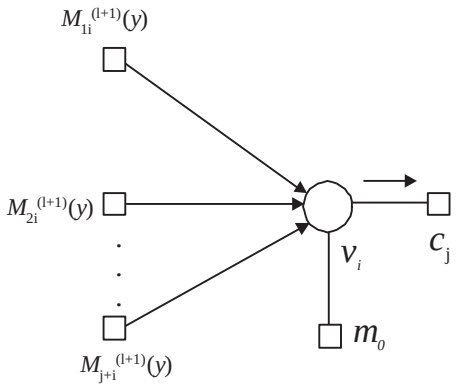


From the variable node symmetry,

$$m_{ij}^{(l+1)}(\mathbf{y}) \;=\; x_i\, m_{ij}^{(l+1)}(\mathbf{z}).$$

Exercise: Show this.
(Hint: First, show

$$\begin{aligned} m_{ij}^{(l+1)}(\mathbf{y}) &= \Psi_v^{(l+1)}\left(m_{ij}^{(0)}(\mathbf{y}), m_{1i}^{(l+1)}(\mathbf{y}), \cdots, m_{j-1,i}^{(l+1)}(\mathbf{y})\right) \\ &= x_i\, m_{ij}^{(l+1)}(\mathbf{z}) \end{aligned}$$



– By induction, all messages to and from variable node x_i when $\mathbf{y}$ is received are equal to the product x_i and the corresponding message when $\mathbf{z}$ is received

– Hence, both decoders commit exactly the same number of errors (if any), which proves the claim.

Remark:

The entire behavior of the decoder can be predicted from its behavior *assuming transmission of the all-one codeword*.

□ Density Evolution and Threshold Determination

- **Assumption**

  1) The decoding neighborhood of depth $2l$ is tree-like.

  2) The all-one codeword was transmitted.

Note:

1) The number of incorrect messages is equal to the number of messages with nonpositive sign (by Assumption 2)

2) A message passed along an edge is *correct* if the sign of the message agrees with the transmitted bit. Otherwise, it is *incorrect*.

- **Discrete Alphabets**: $\mathcal{M} = \{-q, \cdots, -1, 0, 1, \cdots, q\}$

  $p_k^{(l)}$ = Probability that the message sent from variable nodes at iteration l is equal to k

  $q_k^{(l)}$ = Probability that the message sent from check nodes to variable nodes in the lth iteration is equal to k

- **Gallager's Decoding Algorithm A**

  – Ensemble of (d_v, d_c)-regular graphs

  – BSC with $\mathcal{O} = \mathcal{M} = \{\pm 1\}$ and crossover probability ϵ. Then

  $$p_1^{(0)} = 1 - \epsilon, \quad p_{-1}^{(0)} = \epsilon.$$

  – *The message maps are time-invariant*, i.e., they do not depend on the iteration number and are given by

  $$\Psi_v(m_0) = m_0,$$

  $$\Psi_v(m_0, m_1, \cdots, m_{d_v-1}) = \begin{cases} -m_0 & \text{if } m_1 = m_2 = \cdots = m_{d_v-1} = -m_0 \\ m_0 & \text{otherwise} \end{cases}$$

  $$\Psi_c(m_1, \cdots, m_{d_c-1}) = \prod_{i=1}^{d_c-1} m_i$$

Note:

1) The check nodes send a message indicating the modulo-two sum of the other neighboring variables.
2) The variable nodes send their received value unless the incoming messages are unanimous.

– Update for $p_k^{(l)}$ and $q_k^{(l)}$:

$$\begin{aligned}(q_{-1}^{(l)}, q_1^{(l)}) &= {}^*\Psi_c\left((p_{-1}^{(l-1)}, p_1^{(l-1)}), \cdots, (p_{-1}^{(l-1)}, p_1^{(l-1)})\right) \\ &= \frac{1}{2}\Big(\underbrace{1-(1-2p_{-1}^{(l-1)})^{d_c-1}}_{\text{odd}}, \underbrace{1+(1-2p_{-1}^{(l-1)})^{d_c-1}}_{\text{even}}\Big)\end{aligned}$$

$$\begin{aligned}(p_{-1}^{(l)}, p_1^{(l)}) &= {}^*\Psi_v\left((p_{-1}^{(0)}, p_1^{(0)}), (q_{-1}^{(l)}, q_1^{(l)}), \cdots, (q_{-1}^{(l)}, q_1^{(l)})\right) \\ &= \Big(p_1^{(0)}(q_{-1}^{(l)})^{d_v-1} + p_{-1}^{(0)}(1-(q_1^{(l)})^{d_v-1}), \\ &\qquad p_{-1}^{(0)}(q_1^{(l)})^{d_v-1} + p_1^{(0)}(1-(q_{-1}^{(l)})^{d_v-1}\Big).\end{aligned}$$

– Therefore,

$$\begin{aligned}p_{-1}^{(l)} &= p_{-1}^{(0)} - p_{-1}^{(0)}\left[\frac{1+(1-2p_{-1}^{(l-1)})^{d_c-1}}{2}\right]^{d_v-1} \\ &\quad + (1-p_{-1}^{(0)})\left[\frac{1-(1-2p_{-1}^{(l-1)})^{d_c-1}}{2}\right]^{d_v-1}.\end{aligned}$$

Note:

1) For a fixed value of $p_{-1}^{(l-1)}$, $p_{-1}^{(l)}$ *is an increasing function of* $p_{-1}^{(0)}$.
2) For a fixed value of $p_{-1}^{(0)}$, $p_{-1}^{(l)}$ *is an increasing function of* $p_{-1}^{(l-1)}$.
3) $p_{-1}^{(l)}$ *is an increasing function of* $p_{-1}^{(0)}$ by induction.

– Threshold:

ϵ^* = the supremum of all values of $p_{-1}^{(0)} \in [0,1]$ such that

$$\lim_{l\to\infty} p_{-1}^{(l)} = 0.$$

Note that

$$\forall\ p_{-1}^{(0)} < \epsilon^*, \qquad \lim_{l\to\infty} p_{-1}^{(l)} = 0.$$

– Maximal parameter ϵ^* for the BSC and Gallager's decoding algorithms A and B, algorithm E(Erasures in the decoder), and belief propagation (BP).

d_v	v_c	Rate	$\epsilon^*(A)$	$\epsilon^*(B)$	$\epsilon^*(E)$	$\epsilon^*(BP)$	ϵ_{opt}
3	6	0.5	0.04	0.04	0.07	0.084	0.11
4	8	0.5	0.047	0.051	0.059	0.076	0.11
5	10	0.5	0.027	0.041	0.055	0.068	0.11
3	5	0.4	0.061	0.061	0.096	0.113	0.146
4	6	0.333	0.066	0.074	0.09	0.116	0.174
3	4	0.25	0.106	0.106	0.143	0.167	0.215

– Note that algorithms A and B only differ for $d_v > 3$. Also, listed is the maximum allowed value ϵ_{opt} corresponding to the channel capacity.

– Note that this decoding algorithm is *universal* in that it does not require knowledge of the channel parameter.

- **Gallager's Decoding Algorithm B**
  - For $d_v > 3$, it is more efficient.
  - The message along edge $\overrightarrow{e} = (v_i, c_j)$ equals the received value r_i unless at least b incoming check messages (excluding the message along edge $\overrightarrow{e}$) disagree with the received value, in which case the opposite of r_i is sent.
  - In general, $b = b^{(l)}$ is a function of d_c, d_v, and l
  - Message maps

$$\Psi_v(m_0) = m_0$$

$$\Psi_v(m_0, m_1, \cdots, m_{d_v-1}) = \begin{cases} -m_0 & \text{if } |\{i : m_i = -m_0\}| \geq b_l \\ m_0 & \text{otherwise} \end{cases}$$

$$\Psi_c(m_1, \cdots, m_{d_c-1}) = \prod_{i=1}^{d_c-1} m_i$$

  - Evolution of $p_{-1}^{(l)}$

$$\begin{aligned} p_{-1}^{(l)} = p_{-1}^{(0)} - p_{-1}^{(0)} \sum_{k=b_l}^{d_v-1} \binom{d_v-1}{k} \left[\frac{1+(1-2p_{-1}^{(l-1)})^{d_c-1}}{2}\right]^k \\ \cdot \left[\frac{1-(1-2p_{-1}^{(l-1)})^{d_c-1}}{2}\right]^{d_v-1-k} \\ + \left(1-p_{-1}^{(0)}\right) \sum_{k=b_l}^{d_v-1} \binom{d_v-1}{k} \left[\frac{1-(1-2p_{-1}^{(l-1)})^{d_c-1}}{2}\right]^k \\ \cdot \left[\frac{1+(1-2p_{-1}^{(l-1)})^{d_c-1}}{2}\right]^{d_v-1-k} \end{aligned}$$

Note: The optimal choice of b_l is given by the smallest integer b for which

$$\frac{1-p_{-1}^{(0)}}{p_{-1}^{(0)}} \leq \left[\frac{1+(1-2p_{-1}^{(l-1)})^{d_c-1}}{1-(1-2p_{-1}^{(l-1)})^{d_c-1}}\right]^{2b-d_v+1}.$$

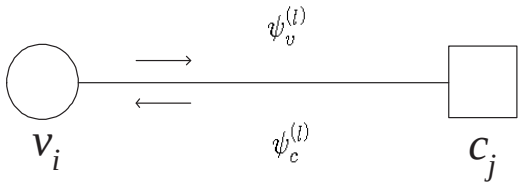
- **BSC with Erasures in the Decoder : Algorithm E**
  - Alphabet: $\mathcal{M} = \{-1, 0, 1\}$
  - *Decoding rule*:

$$\Psi_v^{(0)}(m_0) = m_0;$$

$$\Psi_v^{(l)}(m_0, m_1, \cdots, m_{d_v-1}) = \mathrm{sgn}\left(w^{(l)} m_0 + \sum_{i=1}^{d_v-1} m_i\right)$$

where $w^{(l)}$ is an appropriately chosen weight; and

$$\Psi_c(m_1 \cdots, m_{d_c-1}) = \prod_{i=1}^{d_c-1} m_i$$



  - *Evolution of* $Q^{(l)} \triangleq (q_1^{(l)}, q_{-1}^{(l)}, q_0^{(l)})$:

$$Q^{(l)} = {}^*\Psi_c^{(l)}(P^{(l-1)}, \cdots, P^{(l-1)})$$

where $P^{(l)} \triangleq (p_1^{(l)}, p_{-1}^{(l)}, p_0^{(l)})$.

Therefore,

$$q_1^{(l)} = \frac{1}{2}\left[(p_1^{(l-1)} + p_{-1}^{(l-1)})^{d_c-1} + (p_1^{(l-1)} - p_{-1}^{(l-1)})^{d_c-1}\right] : \quad \text{Even}$$

$$q_{-1}^{(l)} = \frac{1}{2}\left[(p_1^{(l-1)} + p_{-1}^{(l-1)})^{d_c-1} - (p_1^{(l-1)} - p_{-1}^{(l-1)})^{d_c-1}\right] : \quad \text{Odd}$$

$$q_0^{(l)} = 1 - (1 - p_0^{(l-1)})^{d_c-1} \qquad ((m_1, \cdots, m_{d_c-1}) \text{ are all nonzero})$$

- *Evolution of* $P^{(l)} \triangleq (p_1^{(l)}, p_{-1}^{(l)}, p_0^{(l)})$:

$$P^{(l)} = \Psi_v^{(l)}(P^{(c)}, Q^{(l)}, \cdots, Q^{(l)})$$

Therefore,

$$\begin{aligned}
p_0^{(l)} &= p_0^{(0)} \sum_{(i,j):\ i-j=0} \binom{d_v-1}{i,j,d_v-1-2i} \left(q_1^{(l)}\right)^i \left(q_{-1}^{(l)}\right)^j \left(q_0^{(l)}\right)^{d_v-1-2i} \\
&\quad + p_1^{(0)} \sum_{(i,j):\ i-j=-w_l} \binom{d_v-1}{i,j,d_v-1-i-j} \left(q_1^{(l)}\right)^i \left(q_{-1}^{(l)}\right)^j \left(q_0^{(l)}\right)^{d_v-1-i-j} \\
&\quad + p_{-1}^{(0)} \sum_{(i,j):\ i-j=w_l} \binom{d_v-1}{i,j,d_v-1-i-j} \left(q_1^{(l)}\right)^i \left(q_{-1}^{(l)}\right)^j \left(q_0^{(l)}\right)^{d_v-1-i-j} \\
p_1^{(l)} &= p_0^{(0)} \sum_{(i,j):\ i-j>0} \binom{d_v-1}{i,j,d_v-1-i-j} \left(q_1^{(l)}\right)^i \left(q_{-1}^{(l)}\right)^j \left(q_0^{(l)}\right)^{d_v-1-i-j} \\
&\quad + p_1^{(0)} \sum_{(i,j):\ i-j>-w_l} \binom{d_v-1}{i,j,d_v-1-i-j} \left(q_1^{(l)}\right)^i \left(q_{-1}^{(l)}\right)^j \left(q_0^{(l)}\right)^{d_v-1-i-j} \\
&\quad + p_{-1}^{(0)} \sum_{(i,j):\ i-j>w_l} \binom{d_v-1}{i,j,d_v-1-i-j} \left(q_1^{(l)}\right)^i \left(q_{-1}^{(l)}\right)^j \left(q_0^{(l)}\right)^{d_v-1-i-j} \\
p_{-1}^{(l)} &= 1 - p_1^{(l)} - p_0^{(l)}
\end{aligned}$$

Remark:

1) At any given step l, different choices of the weight $w^{(l)}$ will result in different densities $(p_{-1}^{(l)}, p_0^{(l)}, p_1^{(l)})$.
There is no clear (linear) ordering among those alternatives.

2) Minimize $p_{-1}^{(l)} + \alpha p_0^{(l)}$, where $\alpha > 0$ (say $\alpha = 1/2$)

$\Rightarrow$ Find the optimum weights $w^{(1)}, w^{(2)}, \cdots, w^{(l)}$ by dynamic programming.

Example: For a (3.6) regular LDPC code, $w^{(1)} = 2$ and $w^{(l)} = 1,\ \forall\, l \geq 2$.

The advantage of such an approach is that it is widely applicable regardless of how many alternative maps there are and regardless of how many levels of

quantization we have.

The major drawback is that this scheme is computationally intensive and that it quickly becomes infeasible if the size of the message alphabet becomes large.

3) Sensible heuristics:
Will assume that the weight $w^{(l)}, l \in N$, maximizes

$$1 - p_0^{(l)} + h(p_0^{(l)}) - h(p_0^{(l)}, p_{-1}^{(l)})$$

which is the capacity of a memoryless symmetric channel with binary input and ternary output with a crossover probability of $p_{-1}^{(l)}$ and an erasure probability of $p_0^{(l)}$.

4) The largest achievable parameter ϵ^* for Algorithm E is significantly larger than the corresponding entries for Gallager A and B. (Table I)

The simple decoder with erasures performs almost as well as a belief-propagation decoder.

- **Quantized Continuous Channels with Erasures in the Decoder**

1) Map continuous channels to BEC
   - Pick a symmetric threshold τ around zero.
   - Negative values, Erasures, Positive values: $r \leq -\tau, \ -\tau < r < \tau, \ r \geq \tau$.
2) Apply decoding algorithm E.
3) Optimize the threshold σ^* over τ.
4) Adaptive quantization
   - Pick a suitable threshold value τ_l at every step of the decoding algorithm
   - Pick $w^{(l)} \ \ l \geq 0$

Example: For the BIAWGN channel and the $(3, 6)$-regular code,

$$\sigma^* = 0.743 \quad \rightarrow \quad \epsilon = 8.9\%$$

• Quantized Gaussian Channel, 3-bit Messages

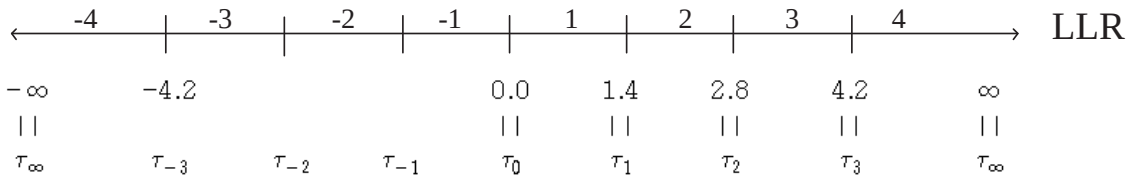
1) $(3, 6)$-Regular codes

2) $\mathcal{O} = \mathcal{M} = \{-4,\ -3,\ -2,\ -1,\ 1,\ 2,\ 3,\ 4\}$

3) Output of the Gaussian channel at time t: $y_t \triangleq x_t + z_t$

$\Rightarrow\ LLR \triangleq \log\frac{P_1}{P_{-1}} = \frac{2}{\sigma^2}y_t = \frac{2}{\sigma^2}(x_t + z_t)$

4) Threshold set τ



threshold

5) Check message map: $\Psi_c(m_1, m_2, \cdots, m_{d_c-1}) = sM$

- $s \triangleq$ sign $\triangleq \prod_{i=1}^{d_c-1}\mathrm{sgn}(m_i)$
- $M \triangleq$ reliability
- $n_q \triangleq$ the number of received messages out of all $(d_c - 1)$ messages under consideration with reliability (absolute value) $q, q \in \{1, 2, 3, 4\}$
- *Decision tree to determine M*:
  - · if $n_1 > 0$ or $n_2 > 3$ then M =1.
  - · if $n_2 > 1$ then M =2.
  - · if $n_2 = 1$ and $n_3 > 1$ then $M = 2$.
  - · if $n_2 = 1$ or $n_3 > 2$ then $M = 3$.
  - · $M = 4$

  Note:
  - · STOP the tests as soon as M is determined.
  - · $n_1 + n_2 + n_3 + n_4 = d_c - 1 = 5$.

6) Variable message map $\Psi_v(m_\prime, m_1, m_2)$

- $\phi_{\mathcal{O}}: \mathcal{O} \to \{-21, -15, -9, -3, 3, 9, 15, 21\}$

$$\phi_{\mathcal{O}}(m) = 6m - 3\ \mathrm{sgn}(m)$$

- $\phi_{\mathcal{M}}: \mathcal{M} \to \{-12, -8, -6, -2, 2, 6, 8, 12\}$

$$\phi_{\mathcal{M}}(m) = 2\ \mathrm{sgn}(m)\left(|m| + \left\lfloor \frac{|m|}{2} \right\rfloor\right)$$

- $\Psi_v(m_0, m_1, m_2)$ is determined by
  - $\phi_{\mathcal{O}}(m_0) + \phi_{\mathcal{M}}(m_1) + \phi_{\mathcal{M}}(m_2)$
  - the threshold set: $\{-\infty, -18, -12, -6, 0, 6, 12, 18, \infty\}$

Example:

$m_0 = 1, m_1 = -4, m_2 = 2$

$\phi_{\mathcal{O}}(1) + \phi_{\mathcal{M}}(-4) + \phi_{\mathcal{M}}(2) = 3 - 12 + 6 = -3 \Rightarrow -1 \in \mathcal{M}$

Note: For the $(3, 6)$-regular code

1) $\sigma^* = 0.847$ with this decoder

2) $\sigma^* = 0.88$ with belief propagation

☐ Continuous Message Alphabets: Belief Propagation

- **Assumptions**

  1) The message alphabet $\mathcal{M}$ is continuous.

  2) The output alphabet $\mathcal{O}$ is discrete or continuous

  3) The messages are the LLR given by $\log \frac{p_1}{p_{-1}}$, where (p_1, p_{-1}) denotes a (conditional) probability density on a bit, satisfying $p_1 + p_{-1} = 1$.

  4) Assume that there are no cycles of length $2l$ or less.
  (Each message is conditionally independent of all others)

- **Density of the message map $\Psi_v(m_0, m_1, \cdots, m_{d_v-1})$ at variable node v**

  1) Message map $\Psi_v(m_0, m_1, \cdots, m_{d_v-1})$:

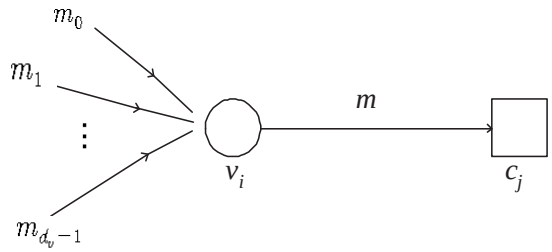
$$\Psi_v(m_0, m_1, \cdots, m_{d_v-1}) = \sum_{i=0}^{d_v-1} m_i$$

  2) Density of $\Psi_v(m_0, m_1, \cdots, m_{d_v-1})$:

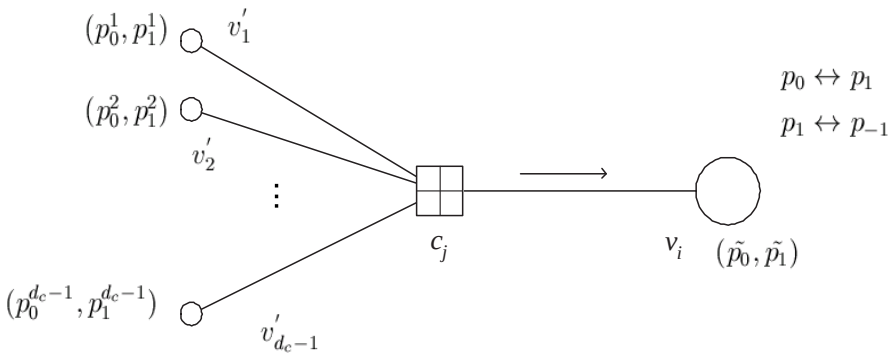
$$\begin{aligned} {}^{*}\Psi_v(P_0, P_1, \cdots, P_{d_v-1}) &= \bigotimes_{i=0}^{d_v-1} P_i \qquad \text{(convolution)} \\ &= \mathcal{F}^{-1}\left(\prod_{i=0}^{d_v-1} \mathcal{F}(P_i) \right) \end{aligned}$$

  where $\mathcal{F}$ is the Fourier transform and P_i denotes the density of m_i.

  Note: In our case, $P_1 = P_2 = \cdots = P_{d_v-1}$



- **Message map at check node**



1) Message along edge k: $m_k \triangleq \log \frac{p_0^k}{p_1^k}$ for $k = 1, 2, \cdots, d_c - 1$.

2) Relation at check node:

$$v_i = v_1' + v_2' + \cdots + v_{d_c-1}' \mod 2$$

3) $\tilde{p}_0 =$ probability that the modulo-2 sum of the $d_c - 1$ independent $\{0, 1\}$-valued random variables is equal to zero.

4) The probability vector $(\tilde{p}_0, \tilde{p}_1)$ is given by the cyclic convolution of the $d_c - 1$ probability vectors (p_0^k, p_1^k), that is,

$$(\tilde{p}_0, \tilde{p}_1) = \bigotimes_{k=1}^{d_c-1} (p_0^k, p_1^k)$$

where $(p_0^1, p_1^2) \bigotimes (p_0^1, p_1^2) \triangleq (p_0^1 p_0^2 + p_1^1 p_1^2, p_0^1 p_1^2 + p_1^1 p_0^2)$.

5) An efficient way of performing these convolutions is by means of a Fourier transform.

Note: 2-point Fourier transform

- f is a function over $\mathrm{GF}(2) = \mathbb{Z}/2\mathbb{Z}$. The Fourier transform $\mathcal{F}(f)$ of f over $\{0, 1\}$ is given by

$$\mathcal{F}(f)(s) = \sum_{i=0}^{1} f(i)(-1)^{is}, \quad s \in \{0, 1\}$$

Note that $\mathcal{F}(f)(0) = f(0) + f(1)$ and $\mathcal{F}(f)(1) = f(0) - f(1)$.

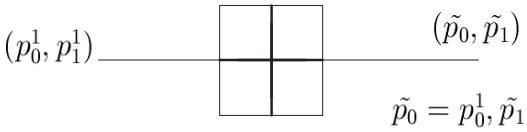
– If f is a probability mass function, then $\mathcal{F}(f)(0) = 1$.
  $\Rightarrow$ The value $\mathcal{F}(f)(1)$ gives all information on f.

Exercise: Show that

$$\tilde{p_0} - \tilde{p_1} = \prod_{k=1}^{d_c-1} (p_0^k - p_1^k).$$

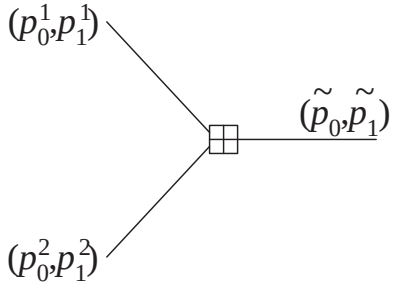
Sketch of direct proof) Use induction on d_c.

$d_c = 2$:



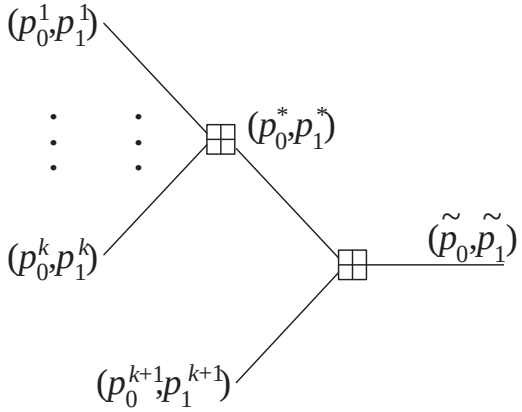
$$\tilde{p_0} = p_0^1, \tilde{p_1} = p_1^1$$
$$\therefore \tilde{p_0} - \tilde{p_1} = p_0^1 - p_1^1$$

Case $d_c = 3$



$$\tilde{p_0} = p_0^1 p_0^2 + p_1^1 p_1^2$$
$$\tilde{p_1} = p_0^1 p_1^2 + p_1^1 p_0^2$$
$$\Rightarrow \tilde{p_0} - \tilde{p_1} = (p_0^1 - p_1^1)(p_0^2 - p_1^2)$$

Case $d_c = k + 1$



$$\tilde{p_0} - \tilde{p_1} = \underbrace{(p_0^* - p_1^*)}_{=\prod_{i=1}^{k}(p_0^i - p_1^i)} (p_0^{k+1} - p_1^{k+1})$$

Note: Transformation from $q \triangleq p_0 - p_1$ to the LLR given by $m \triangleq \log \frac{p_0}{p_1}$

$$\begin{aligned} q &= \frac{e^m - 1}{e^m + 1} = \tanh\left(\frac{m}{2}\right) \\ m &= \log \frac{1+q}{1-q} \end{aligned}$$

The check message map $\Psi_c(m_1, \cdots, m_{d_c-1})$ is given by

$$\Psi_c(m_1, \cdots, m_{d_c-1}) = \log\left(\frac{1 + \prod_{i=1}^{d_c-1} \tanh\frac{m_i}{2}}{1 - \prod_{i=1}^{d_c-1} \tanh\frac{m_i}{2}}\right),$$

that is,

$$\tanh\frac{m}{2} = \prod_{k=1}^{d_c-1} \tanh\frac{m_k}{2} \qquad \text{(tanh rule)}$$

since

$$\tilde{p}_0 - \tilde{p}_1 = \prod_{k=1}^{d_c-1} (p_0^k - p_1^k)$$

- **Evolution of densities of the messages at check nodes**

1) *Representation of a probability density* (p_0, p_1)

$$(p_0, p_1) \quad \begin{matrix} \nearrow \\ \searrow \end{matrix} \quad \begin{matrix} \log \frac{p_0}{p_1} \\ p_0 - p_1 \end{matrix} \quad \longleftrightarrow \quad \begin{matrix} (\text{lg sgn}(p_0 - p_1), & -\log|p_0 - p_1|) \\ \uparrow & \uparrow \\ \text{GF(2)} & [0, \infty) \end{matrix}$$

where

$$\text{lg sgn} q \triangleq \begin{cases} 1 & q < 0 \\ 0 & q > 0 \end{cases}$$

Note that under this representation, the check node message map in the given space $\text{GF}(2) \times [0, \infty)$ is simply addition.

2) Find a pdf under the change of variables:

Let

$$P(m) = \text{the density of} \quad m \triangleq \log \frac{p_0}{p_1}$$
$$\tilde{P}(a, y) = \text{the density of} \quad (a, y) \triangleq \left(\text{lg sgn } m, -\log \left|\tanh \tfrac{m}{2}\right|\right)$$

Question: Given a density P of log-likelihoods, how can we compute the equivalent density $\tilde{P}$ over $\text{GF}(2) \times [0, \infty)$?

Note that

$$\text{lg sgn } \tanh\left(\frac{m}{2}\right) = \text{lg sgn } m$$

since $\tanh$ is odd.

3) Define a new variable y:

If $m > 0$, $\quad y \triangleq -\log \tanh\left(\frac{m}{2}\right)$
Symmetrical relation: $\quad m = -\log \tanh\left(\frac{y}{2}\right)$

If $m < 0$, $\quad y \triangleq -\log\left(-\tanh\left(\frac{m}{2}\right)\right) = -\log \tanh\left(-\frac{m}{2}\right)$
Symmetrical relation: $\quad -m = -\log \tanh\left(\frac{y}{2}\right)$

4) *Compute the density $\tilde{P}(a,y)$ from a given $P(m)$*:

For $y \in [0,\infty)$, the components of $\tilde{P}(a,y)$ with respect to a are defined by

$$\begin{aligned} \tilde{P}^0(y) &\triangleq \tilde{P}(0,y) \\ \tilde{P}^1(y) &\triangleq \tilde{P}(1,y) \end{aligned}$$

Then

$$\begin{aligned} \tilde{P}^0(y) &= \frac{1}{\sinh(y)} P\left(-\log\tanh\frac{y}{2}\right) \\ \tilde{P}^1(y) &= \frac{1}{\sinh(y)} P\left(\log\tanh\frac{y}{2}\right) \end{aligned}$$

Exercise: Show that the above relation holds.

Aside:

1) $\sinh x = \frac{e^x - e^{-x}}{2}$
2) $\cosh x = \frac{e^x + e^{-x}}{2}$
3) $\tanh' x = \left(\frac{\sinh x}{\cosh x}\right)' = \frac{1}{\cosh^2 x}$
4) $2\sinh x \cosh x = \sinh(2x)$
5) $P(y) = |\frac{dx}{dy}| P(x)$ $\quad (y = f(x)$: one-to-one)

5) *Compute the density $P(m)$ from $\tilde{P}(a,y)$* (or $\tilde{P}^0(y)$ and $\tilde{P}^1(y)$):

$$P(m) = \begin{cases} \frac{1}{\sinh(m)} \tilde{P}^0\left(-\log\tanh\frac{m}{2}\right) & m > 0 \\ \frac{1}{\sinh(m)} \tilde{P}^1\left(\log\tanh\left(\frac{-m}{2}\right)\right) & m < 0 \end{cases}$$

Exercise: Show that the above relation holds.

6) *Fourier transform (or Laplace transform) of* $\tilde{P}(a,y)$:

$$\begin{aligned}\mathcal{F}(\tilde{P})(b,s) &\triangleq \sum_{a\in \mathsf{GF}(2)} \int_0^\infty \tilde{P}(a,y)(-1)^{ab}e^{-sy}dy \\ &= \sum_{a\in \mathsf{GF}(2)} (-1)^{ab} \int_0^\infty \tilde{P}^a(y)e^{-sy}dy\end{aligned}$$

In other words,

$$\begin{aligned}\mathcal{F}(\tilde{P})(0,s) &= \int_0^\infty \tilde{P}^0(y)e^{-sy}dy + \int_0^\infty \tilde{P}^1(y)e^{-sy}dy \\ &= \hat{\tilde{P}}^0(s) + \hat{\tilde{P}}^1(s) \\ \mathcal{F}(\tilde{P})(1,s) &= \hat{\tilde{P}}^0(s) - \hat{\tilde{P}}^1(s)\end{aligned}$$

where $\hat{\tilde{P}}^a(s)$ is the Laplace transform of $\tilde{P}^a(y) \triangleq \tilde{P}(a,y)$, i.e.,

$$\hat{\tilde{P}}^a(s) = \int_0^\infty \tilde{P}^a(y)e^{-sy}dy$$

for $a \in \mathsf{GF}(2)$.

7) Transform-domain update at the check node:

– One-to-one correspondence between two representations:

$$m_i \in [-\infty,\infty] \quad \overset{1:1}{\longleftrightarrow} \quad \begin{array}{l}\tilde{m}_i \in \mathsf{GF}(2)\times[0,\infty) \\ \| \\ (\log \mathsf{sgn}\ m, -\log\tanh\left(\frac{m}{2}\right))\end{array}$$

– Update at check node

$$\prod_{i=1}^{k} \tanh\frac{m_i}{2} = \tanh\frac{m}{2} \quad \longleftrightarrow \quad \sum_{i=1}^{k}\tilde{m}_i = \tilde{m}$$

multiplicative additive

where $\tilde{m}$ is the sum of putatively independent randem variables.

– Computation of the dentsity $\tilde{Q}$ of $\tilde{m}$ using the Fourier transform:

$$\mathcal{F}(\tilde{Q})(0,s) = \prod_{i=1}^{k} \mathcal{F}(\tilde{P}_i)(0,s) = \prod_{i=1}^{k} (\hat{\tilde{P}}_i^0(s) + \hat{\tilde{P}}_i^1(s))$$
$$\mathcal{F}(\tilde{Q})(1,s) = \prod_{i=1}^{k} \mathcal{F}(\tilde{P}_i)(1,s) = \prod_{i=1}^{k} (\hat{\tilde{P}}_i^0(s) - \hat{\tilde{P}}_i^1(s))$$

– $\tilde{Q}$ may be obtained by performing the inverse Fourier transform.

– Compute the density $Q \triangleq {}^*\Psi_c(P_1, P_2, \cdots, P_{d_c-1})$ by a change of variables.

- **Density evolution for belief propagation for (d_v, d_c)-regular graphs**
  - Notation

    $P^{(l)}$: the common density associated with the messages from variable nodes to check nodes in the lth round

    P_0: the density of the received values

  - *Procedure of density evolution*

    1) Find the density $\tilde{P}^{(l)}$ corresponding to $P^{(l)}$ (by change of variables).

    2) Determine the density $\tilde{Q}^{(l)}$.

$$\hat{\tilde{Q}}^{(l),0} - \hat{\tilde{Q}}^{(l),1} = \left(\hat{\tilde{P}}^{(l-1),0} - \hat{\tilde{P}}^{(l-1),1}\right)^{d_c-1}$$
$$\hat{\tilde{Q}}^{(l),0} + \hat{\tilde{Q}}^{(l),1} = \left(\hat{\tilde{P}}^{(l-1),0} + \hat{\tilde{P}}^{(l-1),1}\right)^{d_c-1}$$

    where $\hat{\tilde{Q}}^{(l),a}$ is the Laplace transform of $\tilde{Q}^{(l),a}$.

    3) Obtain $Q^{(l)}$ from $\tilde{Q}^{(l)}$ (by performing the appropriate change of measure)

    4) $\mathcal{F}(P^{(l+1)}) = \mathcal{F}(P_0)\left(\mathcal{F}(Q^{(l)})\right)^{d_v-1}$

    5) Compute $P^{(l+1)}$ by IFFT.

- **Monotonicity - Threshold Effects**

1) A class of channels
   - *fulfill the required symmetry condition*; and
   - *is parameterized by* α.

   Example
   - crossover probability ϵ for the BSC
   - standard deviation σ for the BIAWGNC
   - λ for the BILC

2) The parameter α reflects a natural ordering of the channels.
   - The capacity decreases with increasing parameter α
   - Convergence for a parameter α

     $\Rightarrow$ Convergence for every parameter α' such that $\alpha' \leq \alpha$.

3) A channel W is represented by its *transition probability* $P_w(y|x)$.

   A channel W' is *physically degraded* with respect to W if

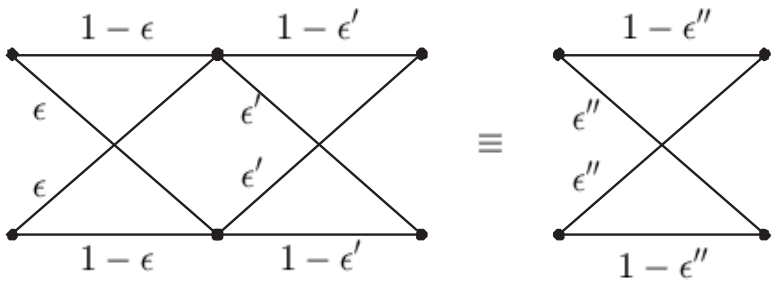
$$P_{W'}(y'|x) = P_Q(y'|y)P_W(y|x)$$

   for some auxiliary channel Q.

Theorem 2 (Monotonicity for physically degraded channels) *Let W and W' be two given memoryless channels that fulfill the required channel symmetry conditions. Assume that W' is physically degraded with respect to W. For a given code and a belief-propagation decoder, let p be the expected fraction of incorrect messages passed at the lth decoding iteration assuming tree-like neighborhoods and transmission over the channel W, and let p' denote the equivalent quantity for transmission over channel W'. Then $p \leq p'$.*

A class of channels parameterized by α is *monotone with respect to a particular decoder* if the convergence for a parameter α implies the convergence for every parameter α' such that $\alpha' < \alpha$.

Example: [BSC: Monotonicity by Self-Concatenation]



- The concatenation of two BSCs (in any order) is again a BSC.
- The parameter of the concatenated channel is

$$\epsilon'' = (1-\epsilon')\epsilon + (1-\epsilon)\epsilon' \qquad (*)$$

- For any $\epsilon'' \geq 0$ and any $\epsilon' \leq \epsilon''$, there exists $\epsilon \geq 0$ such that (*) is fulfilled.

  $\Rightarrow$ The class of BSC is monotone with respect to a belief propagation decoder. □

4) *Monotonicity holds* even for

- the class of BECs
- the class of BIAWGNCs
- the class of Cauchy channels with

$$P_{\text{Cauchy}}(z) = \frac{\lambda}{\pi(\lambda^2 + z^2)}$$

- the class of BILCs with

$$P_{\text{BIL}}(z) = \frac{1}{2\lambda}\, e^{-\frac{|z|}{\lambda}}$$

Exercise: Show that the above statements hold.

- **Thresholds for Belief Propagation**

1) BSC under belief propagation

$$P_0(x) = \epsilon\,\delta\left(x+\log\frac{1-\epsilon}{\epsilon}\right) + (1-\epsilon)\,\delta\left(x-\log\frac{1-\epsilon}{\epsilon}\right)$$

2) BIAWGN channel under belief propagation

$$P_0\left(\frac{2}{\sigma^2}x\right) = \frac{\sigma}{2\sqrt{2\pi}}\exp\left[-\frac{(x-1)^2}{2\sigma^2}\right]$$

3) BILC under belief propagation

$$P_0(x) = \frac{1}{2}\delta\left(x+\frac{2}{\lambda}\right) + \frac{1}{2}e^{-\frac{2}{\lambda}}\delta\left(x-\frac{2}{\lambda}\right) + \frac{1}{4}e^{-x+\frac{2}{\lambda}}\chi_{|x|\le\frac{2}{\lambda}}$$

where

$$\chi_{|x|\le\frac{2}{\lambda}} \triangleq \begin{cases} 1 & |x| \le \frac{2}{\lambda} \\ 0 & |x| > \frac{2}{\lambda}. \end{cases}$$

4) Threshold value σ^* for the BIAWGNC under belief propagation for various code parameters. Also listed is the maximum allowed value σ_{opt}.

d_v	d_c	Rate	σ^*	σ_{opt}
3	6	0.5	0.88	0.979
4	8	0.5	0.83	0.979
5	10	0.5	0.79	0.979
3	5	0.4	1.0	1.148
4	6	0.333	1.01	1.295
3	4	0.25	1.26	1.549

5) Threshold value λ^* for the BILC under belief propagation for various code parameters. Also listed is the maximum allowed value λ_{opt}.

d_v	d_c	Rate	λ^*	λ_{opt}
3	6	0.5	0.65	0.752
4	8	0.5	0.62	0.752
5	10	0.5	0.58	0.752
3	5	0.4	0.77	0.914
4	6	0.333	0.78	1.055
3	4	0.25	1.02	1.298

□ Concentration and Convergence to the Cycle-free Case

- **Concentration**

  Assuming that *the codeword length n is large enough*, for almost all codes in the ensemble $\mathcal{C}^n(d_v, d_c)$ transmission will be reliable if and only if the parameter of the channel is below the calculated threshold value.

- **Setup for Analysis**

  1) Assume we are in the lth iteration for a fixed l.
  2) The message passed from variable node v to check node c is a function of the chosen graph and the input to the decoder.
  3) $Z =$ number of incorrect messages among all $d_v n$ variable-to-check messages sent out in the lth iteration
  4) $E(Z) =$ the expected value of Z over all graphs and all decoder inputs
  5) For a given edge $\overrightarrow{e}$ whose directed neighborhood of depth $2l$ is tree-like,
     $p =$ the expected number of incorrect messages (including half the number of erasures) passed along this edge at the lth iteration, averaged over all inputs.

  Example: In the case of continuous message alphabets,

$$p = \int_{-\infty}^{0^-} P(x)dx + \frac{1}{2}\int_{0^-}^{0^+} P(x)dx$$

  where $P(x)$ is the density of the messages at the lth iteration.

Theorem 3 *There exists positive constants* $\beta = \beta(d_v, d_c, l)$ *and* $\gamma = \gamma(d_v, d_c, l)$ *such that*

[Concentration around expected value] For any $\epsilon > 0$,

$$\Pr\left\{\left|Z - E[Z]\right| > nd_v\epsilon/2\right\} \leq 2e^{-\beta\epsilon^2 n}.$$

[Convergence to cycle-free case] For any $\epsilon > 0$ and $n > \frac{2\gamma}{\epsilon}$,

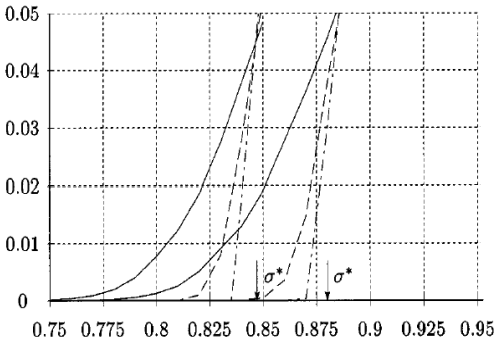
$$\left|E[Z] - nd_v p\right| < nd_v\epsilon/2.$$

[Concentration around cycle-free case] For any $\epsilon > 0$ and $n > \frac{2\gamma}{\epsilon}$, we have

$$\Pr\left\{\left|Z - nd_v p\right| > nd_v\epsilon\right\} \leq 2e^{-\beta\epsilon^2 n}.$$

Note: From the concentration theorem, (for sufficiently large n) almost all codes can transmit reliably up to the threshold value, but they have an error probability bounded away from zero above the threshold value.

- *Bit-error probability versus parameter* σ for the $(3, 6)$-regular code used over the BIAWGN channel. The left curves correspond to the message passing algorithm with eight messages, whereas the right curves correspond to belief propagation. The solid curves correspond to a codeword length of 1000, whereas the dashed and the dotted-dashed curves correspond to codeword lengths of 10000 and 100000, respectively. The arrows indicate the corresponding threshold values $\epsilon^* = 0.847$ and $\epsilon^* = 0.88$. Observe how the lines move closer to this threshold values for increasing codeword lengths.



□ Extensions of Density Evolution

- **Extension to Irregular graphs**

1) *Variable node degree distribution* $\lambda(x)$:

$$\lambda(x) = \sum_{i \geq 2}^{d_v^{\max}} \lambda_i x^{i-1}$$

where λ_i is the fraction of edges incident to variable nodes with degree i.

2) *Check node degree distribution* $\rho(x)$:

$$\rho(x) = \sum_{i \geq 2}^{d_c^{\max}} \rho_i x^{i-1}$$

where ρ_i is the fraction of edges incident to check nodes with degree i.

3) *Density evolution for irregular LDPC codes*:

– Density Q of the message at check node in the lth iteration

$$\begin{aligned} \hat{\tilde{Q}}^{(l),0} - \hat{\tilde{Q}}^{(l),1} &= \rho\left(\hat{\tilde{P}}^{(l-1),0} - \hat{\tilde{P}}^{(l-1),1}\right) \\ \hat{\tilde{Q}}^{(l),0} + \hat{\tilde{Q}}^{(l),1} &= \rho\left(\hat{\tilde{P}}^{(l-1),0} + \hat{\tilde{P}}^{(l-1),1}\right) \end{aligned}$$

– Density P of the message at variable node in the lth iteration

$$\mathcal{F}(P^{(l)}) = \mathcal{F}\left(P^{(0)}\right) \lambda\left(\mathcal{F}(Q^{(l)})\right)$$

- **Extension to turbo codes under turbo decoding**

1) notion of a neighborhood of a variable node

2) notion of a tree-like neighborhood

- **Overall structure of LDPC codes and belief-propagation decoders**

1) Structure of the code

   – The variables v_i take values r_i in some ring (example: $\mathrm{GF}(2^m)$); and

   – Check nodes represent *weighted sum constraints*. For each check node c_j,

$$0 = \sum_{i:\ \exists e=(v_i,c_j)} w_i r_i$$

   where the weights w_i are elements of the ring.

2) Meaning of the messages

   – *A message is a (representation of a) conditional distribution of the variable associated with the variable node*.

   – All messages arriving at a node at a particular stage (including the received value in the case of a variable node) are assumed to be *conditional distributions of the associated variable, each independent of the others*, i.e., each conditioned on independent random variables.

   – At the variable nodes *the outgoing messages are pointwise products of the densities from incoming messages*.

   – *The outgoing messages from the check nodes represent the distribution of the (additive group) inverse of some weighted sum of variables.*

   ⇒ The outgoing message represents the convolution of the participating incoming messages (suitably adjusted for any multiplicative factor).

Note: In the case of $\mathrm{GF}(2^n)$,

   – The additive group operative at the check nodes is $\mathrm{GF}(2)^n$.

   – The appropriate Fourier transform is just the multidimensional version of the one used for binary parity-check codes.

- **Problem of computing average asymptotic performance**

  Here, we interpret the messages themselves as random variables for which we desire the density.

  1) In each case, i.e., at both sets of nodes, we move to *a log domain so that pointwise products become pointwise sums*.

  2) Once in the log domain, *the density of an outgoing message is the convolution of the density of incoming messages*. Typically, the log domain can be suitably represented so that a Fourier transform exists over the space on which the densities are defined.

  3) The update of the density can then be computed efficiently in this Fourier domain. This applies both at the variable nodes and at the check nodes, but the spaces are different in both cases.

  Note: In the case of GF(2^m)

  - The Fourier transform of the densities can be viewed as a real function over F_2^m, taking values in the interval $[-1, 1]$.

  - *The appropriate log domain is a multidimensional version of the one appearing in the binary case*.

  - Unfortunately, the dimensionality of the space renders computation of the densities infeasible except for quite small m (e.g., $m = 2$).

  Example:

  - Extension to codes over $\mathbb{Z}/(q)$

  - Extension to codes over $\mathbb{Z}^2/(q_1, q_2)$: QAM-type signaling schemes

  - It is the existence of the Fourier transform over these spaces that renders the necessary computation at the check nodes efficient.