LDPC Codes: A Group Algebra Formulation

Emina Soljanin
Bell Labs, Lucent
and
Elke Offer,
TU Munich

San Diego, January 2002

PRELIMINARIES
Questions

- How suboptimal algorithms approximate the optimal:
  θ approximates $\sin\theta$ for small θ
  $1+2x$ approximates $(1+x)^2$ for small x

- How the approximate solutions approach the optimal:
  gradient method
  greedy algorithm

- Why are codes and approximate algorithms good?

PRELIMINARIES
Notation

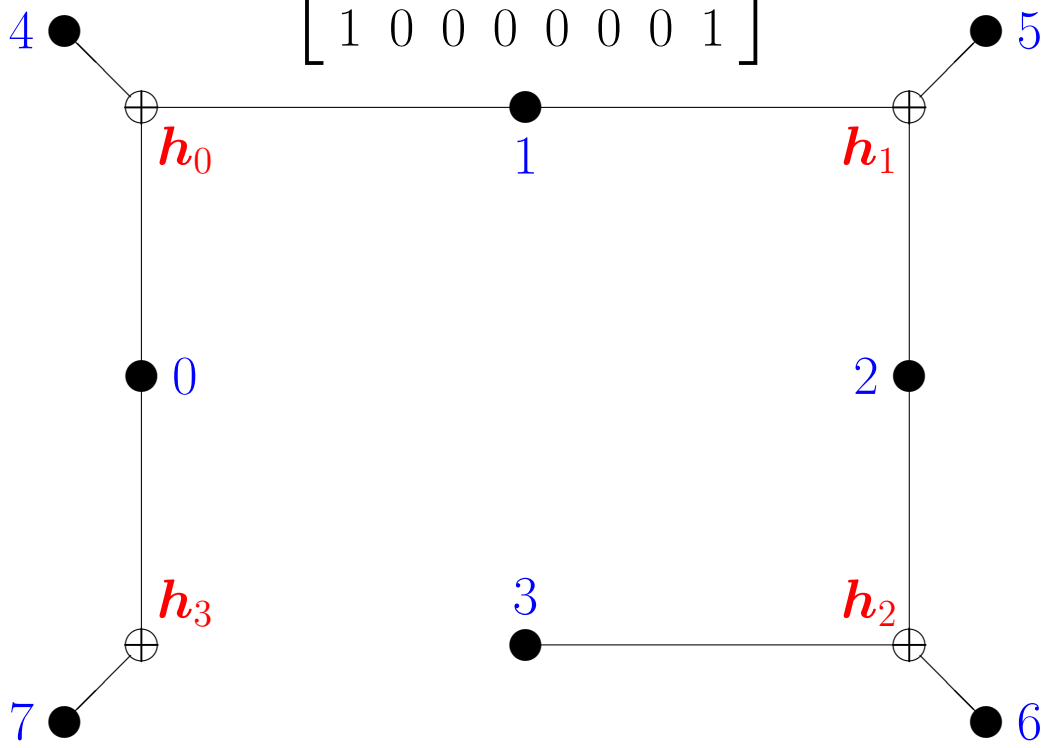
- (n, k) code $\mathcal{C}$ and its dual $(n, n-k)$ code $\mathcal{C}'$
- $\mathcal{C}$ and $\mathcal{C}'$ are subgroups of the additive group $\mathbb{F}_2^n$
- $\boldsymbol{h}_i$, $0 \leq i < n-k$: group generators and a basis for $\mathcal{C}'$
- $\boldsymbol{h}_i = \{h_{ij}\}_{0 \leq j < n}$: independent parity-checks for $\mathcal{C}$, rows of its parity check matrix
- $\mathcal{I} = \{0, 1, \ldots, n-k-1\}$: index set
- $\boldsymbol{r}$: the received word

PRELIMINARIES

Tanner Graph

- *Bipartite* graph: check and variable nodes.
- An $(8, 4)$ tree code:

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



ISOMORPHIC GROUPS
$\mathcal{C}' \cong \mathcal{G}$ and $\mathbb{F}_2^n \cong \mathcal{F}$

- Elements:

  $\mathcal{C}'$: $\boldsymbol{c}' = c'_0 c'_1 \dots c'_{n-1}$

  $\mathcal{G}$: $\boldsymbol{\lambda}^{\boldsymbol{c}'} = \lambda_0^{c'_0} \lambda_1^{c'_1} \dots \lambda_{n-1}^{c'_{n-1}}$

- Operations:

  $\mathcal{C}'$: $\boldsymbol{c}'_p \oplus \boldsymbol{c}'_q = \boldsymbol{c}'_i$

  $\mathcal{G}$: $\boldsymbol{\lambda}^{\boldsymbol{c}'_p} \otimes \boldsymbol{\lambda}^{\boldsymbol{c}_q} = \boldsymbol{\lambda}^{\boldsymbol{c}'_p \oplus \boldsymbol{c}'_q} = \boldsymbol{\lambda}^{\boldsymbol{c}'_i}$

- Note that $\lambda_i = \boldsymbol{\lambda}^{\boldsymbol{e}_i} \in \mathcal{F}$, and thus

$$\lambda_i \otimes \lambda_j = \boldsymbol{\lambda}^{\boldsymbol{e}_i \oplus \boldsymbol{e}_j} = \begin{cases} \lambda_i \cdot \lambda_j, & i \neq j, \\ 1, & i = j. \end{cases}$$

GROUP ALGEBRA $\boldsymbol{R}[\mathcal{G}]$

- Elements of the form $\sum_{g\in\mathcal{G}} \alpha_g g$, $\alpha_g \in \boldsymbol{R}$
- Addition and scalar multiplication in $\boldsymbol{R}[\mathcal{G}]$
- Multiplication in $\boldsymbol{R}[\mathcal{G}]$:

$$\Big(\sum_{f\in\mathcal{G}} \alpha_f f\Big) \otimes \Big(\sum_{g\in\mathcal{G}} \alpha_g g\Big) = \sum_{f,g\in G} \alpha_f \alpha_g (f \otimes g)$$

- Example:
  - multiplication in $\boldsymbol{R}[\mathcal{G}]$:

$$(1+\lambda_0\lambda_1) \otimes (1+\lambda_1\lambda_2) = 1 + \lambda_0\lambda_1 + \lambda_1\lambda_2 + \lambda_0\lambda_2$$

  - multiplication in $\boldsymbol{R}$:

$$(1+\lambda_0\lambda_1) \cdot (1+\lambda_1\lambda_2) = 1 + \lambda_0\lambda_1 + \lambda_1\lambda_2 + \lambda_1^2\lambda_0\lambda_2$$

GROUP ALGEBRA $\boldsymbol{R}[\mathcal{G}]$
Multiplication in $\boldsymbol{R}[\mathcal{G}]$ and $\boldsymbol{R}$

- If $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ have disjoint supports, then

$$\boldsymbol{\lambda}^{\boldsymbol{c}_1} \otimes \boldsymbol{\lambda}^{\boldsymbol{c}_2} = \boldsymbol{\lambda}^{\boldsymbol{c}_1 \oplus \boldsymbol{c}_2} = \boldsymbol{\lambda}^{\boldsymbol{c}_1 + \boldsymbol{c}_2} = \boldsymbol{\lambda}^{\boldsymbol{c}_1} \cdot \boldsymbol{\lambda}^{\boldsymbol{c}_2}.$$

- Let $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{C}$.

- If the supports of codewords in $\mathcal{S}_1$ and the supports of codewords in $\mathcal{S}_2$ are disjoint, then

$$\Big(\sum_{\boldsymbol{c} \in \mathcal{S}_1} \boldsymbol{\lambda}^{\boldsymbol{c}}\Big) \otimes \Big(\sum_{\boldsymbol{c} \in \mathcal{S}_2} \boldsymbol{\lambda}^{\boldsymbol{c}}\Big) = \Big(\sum_{\boldsymbol{c} \in \mathcal{S}_1} \boldsymbol{\lambda}^{\boldsymbol{c}}\Big) \cdot \Big(\sum_{\boldsymbol{c} \in \mathcal{S}_2} \boldsymbol{\lambda}^{\boldsymbol{c}}\Big).$$

SOME ELEMENTS OF $\boldsymbol{R}[\mathcal{G}]$ and $\boldsymbol{R}[\mathcal{F}]$

- $(1+\boldsymbol{\lambda}^{\boldsymbol{h_i}}) \otimes (1+\boldsymbol{\lambda}^{\boldsymbol{h_j}}) = 1+\boldsymbol{\lambda}^{\boldsymbol{h_i}}+\boldsymbol{\lambda}^{\boldsymbol{h_j}}+\boldsymbol{\lambda}^{(\boldsymbol{h_i}\oplus\boldsymbol{h_j})}$

- $(1+\boldsymbol{\lambda}^{\boldsymbol{h_i}}) \otimes (1+\boldsymbol{\lambda}^{\boldsymbol{h_i}}) = 2(1+\boldsymbol{\lambda}^{\boldsymbol{h_i}})$

- A generalized weight enumerator of $\mathcal{C}'$:

$$A_{\mathcal{C}'} = \sum_{\boldsymbol{c'}\in\mathcal{C}'} \boldsymbol{\lambda}^{\boldsymbol{c'}} = \prod_{i\in\mathcal{I}}{}_{\otimes}\,[1+\boldsymbol{\lambda}^{\boldsymbol{h_i}}] = \sum_{\boldsymbol{c'}\in\mathcal{C}'} \prod_{j=0}^{n-1} \lambda_j^{c'_{ij}}$$

- $\underbrace{A_{\mathcal{C}'} \otimes A_{\mathcal{C}'} \otimes \cdots \otimes A_{\mathcal{C}'}}_{\nu} = 2^{(\nu-1)\cdot|\mathcal{I}|} \cdot A_{\mathcal{C}'}$

- Weight enumerators of cosets of $\mathcal{C}'$: $\boldsymbol{\lambda}^{\boldsymbol{e_m}} \otimes A_{\mathcal{C}'}$

BIT-OPTIMAL DECODING

- Computes soft information for m-th bit:

$$L_m^{\mathcal{C}} = \log \frac{P(c_m = 0|\boldsymbol{r})}{P(c_m = 1|\boldsymbol{r})}$$

  where $P(c_m = b|\boldsymbol{r}) = \sum_{\boldsymbol{c}\in\mathcal{C}, c_m = b} P(\boldsymbol{c}|\boldsymbol{r})$.

- Under the usual assumptions,

$$L_m^{\mathcal{C}} = \log \frac{\displaystyle\sum_{\boldsymbol{c}\in\mathcal{C}, c_m=0} \prod_{j=0}^{n-1} p(r_j|c_j)}{\displaystyle\sum_{\boldsymbol{c}\in\mathcal{C}, c_m=1} \prod_{j=0}^{n-1} p(r_j|c_j)}.$$

- Poisson Summation Formula:

$$\frac{1}{|\mathcal{C}|} \sum_{\boldsymbol{c}\in\mathcal{C}} f(\boldsymbol{c}) = \frac{1}{|\mathbb{F}_2^n|} \sum_{\boldsymbol{c}'\in\mathcal{C}'} \hat{f}(\boldsymbol{c}').$$

BIT-OPTIMAL DECODING

- Computations can be done over the dual code:

$$L_m^{\mathcal{C}} = \underbrace{\log \frac{1+\lambda_m}{1-\lambda_m}}_{L_m} + \log \frac{\sum_{\boldsymbol{c}'_i \in \mathcal{C}'} \prod_{\substack{j=0 \\ j \neq m}}^{n-1} \lambda_j^{c'_{ij}}}{\sum_{\boldsymbol{c}'_i \in \mathcal{C}'} (-1)^{c'_{im}} \prod_{\substack{j=0 \\ j \neq m}}^{n-1} \lambda_j^{c'_{ij}}}$$

where

$$L_j = \log \frac{p(r_j|0)}{p(r_j|1)} \text{ and } \lambda_j = \frac{p(r_j|0) - p(r_j|1)}{p(r_j|0) + p(r_j|1)} = \tanh(L_j/2).$$

- We are looking for an expression for $L_m^{\mathcal{C}}$ and

$$\lambda_m^{\mathcal{C}} = \tanh(L_m^{\mathcal{C}}/2)$$

involving only $\boldsymbol{h}_i$, group generators and a basis for $\mathcal{C}'$.

BIT-OPTIMAL DECODING

Group Algebra Formulation

- Computing $L_m^{\mathcal{C}}$ over the dual code:

$$L_m^{\mathcal{C}} = \log \frac{1+\lambda_m}{1-\lambda_m} + \log \frac{\sum\limits_{\boldsymbol{c}'_i \in \mathcal{C}'} \prod\limits_{\substack{j=0 \\ j \neq m}}^{n-1} \lambda_j^{c'_{ij}}}{\sum\limits_{\boldsymbol{c}'_i \in \mathcal{C}'} (-1)^{c'_{im}} \prod\limits_{\substack{j=0 \\ j \neq m}}^{n-1} \lambda_j^{c'_{ij}}}$$

- An expression involving only $\boldsymbol{h}_i$:

$$L_m^{\mathcal{C}} = L_m + \log \frac{\prod\limits_{i=0}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{\substack{j=0 \\ j \neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}{\prod\limits_{i=0}^{n-k-1}{}_{\otimes} \left[1 + (-1)^{h_{im}} \prod\limits_{\substack{j=0 \\ j \neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}.$$

BIT-OPTIMAL DECODING

Computations over subcodes of $\mathcal{C}'$

- Claim: Soft output for the m-th bit:

$$\lambda_m^{\mathcal{C}} = \tanh(L_m^{\mathcal{C}}/2) = \frac{\boldsymbol{\lambda}^{\boldsymbol{e_m}} \otimes \prod_{i \in \mathcal{I}}{}_{\otimes} \left[1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}\right]}{\prod_{i \in \mathcal{I}}{}_{\otimes} \left[1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}\right]}.$$

- Index set partition: $\cup_{l=1}^{L} \mathcal{I}_l = \mathcal{I}$
- $\mathcal{C}'_l \subset \mathcal{C}'$ is defined by its set of generators $\boldsymbol{h}_{i_l}$, $i_l \in \mathcal{I}_l$
- Soft information:

$$\lambda_m^{\mathcal{C}} = \frac{\boldsymbol{\lambda}^{\boldsymbol{e_m}} \otimes \prod_{i \in \mathcal{I}_1}{}_{\otimes} \left[1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}\right] \otimes \cdots \otimes \prod_{i \in \mathcal{I}_L}{}_{\otimes} \left[1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}\right]}{\prod_{i \in \mathcal{I}_1}{}_{\otimes} \left[1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}\right] \otimes \cdots \otimes \prod_{i \in \mathcal{I}_L}{}_{\otimes} \left[1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}\right]}$$

MESSAGE PASSING ON TREES

$\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$

- Soft output for the m-th bit based on code $\mathcal{C}$:

$$\lambda_m^{\mathcal{C}} = \frac{\boldsymbol{\lambda}^{\boldsymbol{e_m}} \otimes \prod_{i \in \mathcal{I}_1}^{\otimes} [1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}] \otimes \prod_{i \in \mathcal{I}_2}^{\otimes} [1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}]}{\prod_{i \in \mathcal{I}_1}^{\otimes} [1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}] \otimes \prod_{i \in \mathcal{I}_2}^{\otimes} [1 + \boldsymbol{\lambda}^{\boldsymbol{h_i}}]} = \frac{\lambda_m \otimes A_{\mathcal{C}_1'} \otimes A_{\mathcal{C}_2'}}{A_{\mathcal{C}_1'} \otimes A_{\mathcal{C}_2'}}.$$

- $\mathcal{I}_1$ and $\mathcal{I}_2$ are obtained by cutting through the l-th variable node of the Tanner graph of $\mathcal{C}'$.

MESSAGE PASSING ON TREES

$\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$

- m-th soft output of the optimal decoder for code $\mathcal{C}_1$:

$$\lambda_m^{\mathcal{C}_1} = \frac{\lambda_m \otimes \prod\limits_{i \in \mathcal{I}_1 \otimes} \left[1 + \lambda_l^{h_{il}} \prod\limits_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{h_{ij}} \right]}{\prod\limits_{i \in \mathcal{I}_1 \otimes} \left[1 + \lambda_l^{h_{il}} \prod\limits_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{h_{ij}} \right]} = \frac{\lambda_m \otimes \sum\limits_{\boldsymbol{c'} \in \mathcal{C}_1'} \lambda_l^{c_l'} \prod\limits_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{c_j'}}{\sum\limits_{\boldsymbol{c'} \in \mathcal{C}_1'} \lambda_l^{c_l'} \prod\limits_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{c_j'}}$$

- l-th soft output of the optimal decoder for code $\mathcal{C}_2$:

$$\lambda_l^{\mathcal{C}_2'} = \frac{\lambda_l \otimes \prod\limits_{i \in \mathcal{I}_2 \otimes} \left[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}} \right]}{\prod\limits_{i \in \mathcal{I}_2 \otimes} \left[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}} \right]} = \frac{\lambda_l \otimes A_{\mathcal{C}_2'}}{A_{\mathcal{C}_2'}}.$$

PEARL'S BELIEF PROPAGATION RESULT
A Simple Proof

- Claim: If we replace (update) λ_l in the equation for $\lambda_m^{\mathcal{C}_1}$ by $\lambda_l^{\mathcal{C}_2'}$, we obtain $\lambda_m^{\mathcal{C}}$.
- Proof: By the substitution, we obtain

$$\sum_{\boldsymbol{c'}\in\mathcal{C}_1'} \lambda_l^{\mathcal{C}_2'} \prod_{\substack{j=0\\ j\neq l}}^{n-1} \lambda_j^{c_j'} = \sum_{\boldsymbol{c'}\in\mathcal{C}_1'} \Big(\frac{\lambda_l \otimes A_{\mathcal{C}_2'}}{A_{\mathcal{C}_2'}}\Big)^{c_l'} \prod_{\substack{j=0\\ j\neq l}}^{n-1} \lambda_j^{c_j'}$$

$$= \sum_{\boldsymbol{c'}\in\mathcal{C}_{1,0}'} \prod_{j=0}^{n-1} \lambda_j^{c_j'} + \sum_{\boldsymbol{c'}\in\mathcal{C}_{1,1}'} \Big(\frac{\lambda_l \otimes A_{\mathcal{C}_2'}}{A_{\mathcal{C}_2'}}\Big) \prod_{\substack{j=0\\ j\neq l}}^{n-1} \lambda_j^{c_j'}$$

$$= \frac{1}{A_{\mathcal{C}_2'}} \Big[A_{\mathcal{C}_2'} \cdot \sum_{\boldsymbol{c'}\in\mathcal{C}_{1,0}'} \prod_{j=0}^{n-1} \lambda_j^{c_j'} + (\lambda_l \otimes A'_{\mathcal{C}_2}) \sum_{\boldsymbol{c'}\in\mathcal{C}_{1,0}'} \prod_{\substack{j=0\\ j\neq l}}^{n-1} \lambda_j^{c_j'} \Big]$$

$$= \frac{1}{A'_{\mathcal{C}_2}} \cdot A_{\mathcal{C}_2'} \otimes \Big[\sum_{\boldsymbol{c'}\in\mathcal{C}_{1,0}'} \prod_{j=0}^{n-1} \lambda_j^{c_j'} + \sum_{\boldsymbol{c'}\in\mathcal{C}_{1,1}'} \prod_{j=0}^{n-1} \lambda_j^{c_j'} \Big]$$

$$= \frac{1}{A_{\mathcal{C}_2'}} (A_{\mathcal{C}_2'} \otimes A_{\mathcal{C}_1'}).$$

LDPC

- The optimal log-likelihood for bit m:

$$L_m^{\mathcal{C}} = L_m + \log \frac{\prod\limits_{i=0}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}{\prod\limits_{i=0}^{n-k-1}{}_{\otimes} \left[1 + (-1)^{h_{im}} \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}.$$

- Computing over subcodes:

$$L_m + \log \frac{\prod\limits_{\substack{i=0 \\ h_{im}=0}}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}}\right] \otimes \prod\limits_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}{\prod\limits_{\substack{i=0 \\ h_{im}=0}}^{n-k-1}{}_{\otimes} \left[1 + \prod\limits_{j=0}^{n-1} \lambda_j^{h_{ij}}\right] \otimes \prod\limits_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left[1 - \prod\limits_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}$$

COMPUTING THE LOG-LIKELIHOOD FOR BIT m

Gallager's Message Passing

The initial log-likelihood for bit m

$$L_m + \log \frac{\prod_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left(1 + \prod_{\substack{l=0 \\ l \neq m}}^{n-1} \lambda_l^{h_{il}}\right)}{\prod_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left(1 - \prod_{\substack{l=0 \\ l \neq m}}^{n-1} \lambda_l^{h_{il}}\right)}.$$

is based on L_m and the parity check equations in which m participates.

COMPUTING THE LOG-LIKELIHOOD FOR BIT m

Gallager's Message Passing

$\lambda_{ml} = \tanh(L_{ml}/2)$ is the Gallager's message passed from variable node l to variable node m through their unique common check node $c(m,l)$:

$$L_{ml} = \log \frac{1+\lambda_l}{1-\lambda_l} + \log \frac{\prod_{\substack{i=0 \\ h_{il}=1 \\ i \neq c(m,l)}}^{n-k-1} \left(1 + \prod_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{h_{ij}}\right)}{\prod_{\substack{i=0 \\ h_{il}=1 \\ i \neq c(m,l)}}^{n-k-1} \left(1 - \prod_{\substack{j=0 \\ j \neq l}}^{n-1} \lambda_j^{h_{ij}}\right)}.$$

LDPC
Gallager's Message Passing

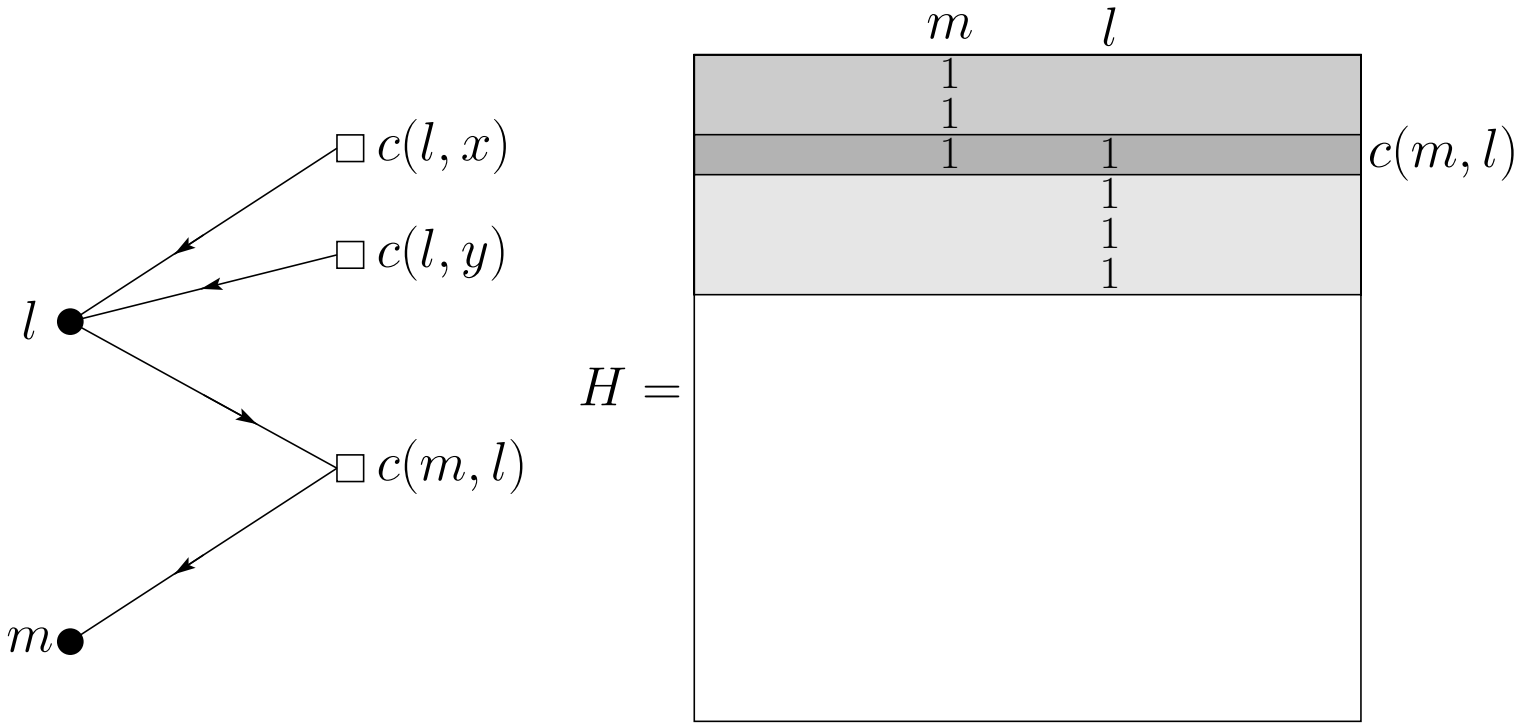
- The updated log-likelihood for bit m:

$$L_m + \log \frac{\prod_{\substack{i=0 \\ h_{im}=0,h_{il}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\right] \otimes \prod_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}{\prod_{\substack{i=0 \\ h_{im}=0,h_{il}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\right] \otimes \prod_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left[1 - \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}.$$

- Compare with the optimal:

$$L_m + \log \frac{\prod_{\substack{i=0 \\ h_{im}=0}}^{n-k-1}{}_{\otimes} \left[1 + \prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\right] \otimes \prod_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left[1 + \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}{\prod_{\substack{i=0 \\ h_{im}=0}}^{n-k-1}{}_{\otimes} \left[1 + \prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\right] \otimes \prod_{\substack{i=0 \\ h_{im}=1}}^{n-k-1}{}_{\otimes} \left[1 - \prod_{\substack{j=0 \\ j\neq m}}^{n-1} \lambda_j^{h_{ij}}\right]}$$

LDPC
Gallager's Message Passing



COMPUTING THE LOG-LIKELIHOOD FOR BIT m
Each Neighbor Sends a Message

- $\mathcal{N}(m)$, the set of $neighbors$ of m:

$$\mathcal{N}(m) = \{l \mid \exists i \in \mathcal{I} \text{ s.t. } h_{im} = 1 \text{ and } h_{il} = 1\}.$$

- Consider the subcodes of $\mathcal{C}'$ defined by the following index sets:

$$\mathcal{I}_m = \{i \in \mathcal{I} \mid h_{im} = 1\}$$
$$\mathcal{I}_{ml} = \{i \in \mathcal{I} \mid h_{il} = 1 \text{ and } h_{im} = 0\}, \quad l \in \mathcal{N}(m)$$
$$\mathcal{I}_{\mathcal{N}(m),m} = \left(\cup_{l \in \mathcal{N}(m)} \mathcal{I}_{ml}\right) \cup \mathcal{I}_m$$

- These index sets are disjoint if there are no loops of length 4, 6, and 8 in the Tanner graph.

COMPUTING THE LOG-LIKELIHOOD FOR BIT m

Gallager's Message Passing

- Initial soft information for bit m:

$$\lambda_m^{\mathcal{C}_m} = \frac{\lambda_m \otimes \prod_{i \in \mathcal{I}_m} \left[1 + \prod_{l \in \mathcal{N}(m)} \lambda_l^{h_{il}}\right]}{\prod_{i \in \mathcal{I}_m} \left[1 + \prod_{l \in \mathcal{N}(m)} \lambda_l^{h_{il}}\right]} = \frac{\lambda_m \otimes A_{\mathcal{C}'_m}}{A_{\mathcal{C}'_m}}$$

- Message to m passed from its neighbor $l \in \mathcal{N}(m)$:

$$\lambda_l^{\mathcal{C}_{ml}} = \frac{\lambda_l \otimes \prod_{i \in \mathcal{I}_{ml}} \left[1 + \prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\right]}{\prod_{i \in \mathcal{I}_{ml}} \left[1 + \prod_{j=0}^{n-1} \lambda_j^{h_{ij}}\right]} = \frac{\lambda_l \otimes A_{\mathcal{C}'_{ml}}}{A_{\mathcal{C}'_{ml}}}$$

- Optimal since there are no loops of length 4.

COMPUTING THE LOG-LIKELIHOOD FOR BIT m

- Update after each neighbor sends a message:

$$\lambda_m^{\mathcal{C}_{\mathcal{N}(m),m}} = \frac{\frac{\lambda_m \otimes A_{\mathcal{C}'_m}}{A_{\mathcal{C}'_m}} \otimes \prod_{i \in \mathcal{I}_m} \left[1 + \prod_{l \in \mathcal{N}(m)} \left(\frac{\lambda_l \otimes A_{\mathcal{C}'_{ml}}}{A_{\mathcal{C}'_{ml}}}\right)^{h_{il}}\right]}{\prod_{i \in \mathcal{I}_m} \left[1 + \prod_{l \in \mathcal{N}(m)} \left(\frac{\lambda_l \otimes A_{\mathcal{C}'_{ml}}}{A_{\mathcal{C}'_{ml}}}\right)^{h_{il}}\right]}$$

- Optimal since there are no loops of length 4 and no loops of length 6 and 8.

- What about long loops?

EXAMPLE: OUR $(8,4)$ TREE CODE

iteration #1

$$L_0^{(1)} = L_0 + \log \frac{(1+\lambda_7)\cdot(1+\lambda_1\lambda_4)}{(1-\lambda_7)\cdot(1-\lambda_1\lambda_4)} \tag{1}$$

$$L_{01}^{(1)} = L_1 + \log \frac{1+\lambda_2\lambda_5}{1-\lambda_2\lambda_5} \text{ and } \lambda_{01}^{(1)} = \frac{\lambda_1+\lambda_2\lambda_5}{1+\lambda_1\lambda_2\lambda_5} \tag{2}$$

$$L_{12}^{(1)} = L_2 + \log \frac{1+\lambda_3\lambda_6}{1-\lambda_3\lambda_6} \text{ and } \lambda_{12}^{(1)} = \frac{\lambda_2+\lambda_3\lambda_6}{1+\lambda_2\lambda_3\lambda_6}$$

iteration #2 – Substituting $\lambda_{01}^{(1)}$ for λ_1 in (1), and $\lambda_{12}^{(1)}$ for λ_2 in (2):

$$L_0^{(2)} = L_0 + \log \frac{(1+\lambda_7)\cdot(1+\lambda_1\lambda_2\lambda_5+\lambda_1\lambda_4+\lambda_2\lambda_4\lambda_5)}{(1-\lambda_7)\cdot(1+\lambda_1\lambda_2\lambda_5-\lambda_1\lambda_4-\lambda_2\lambda_4\lambda_5)}$$

$$= L_0 + \log \frac{(1+\lambda_7)\otimes(1+\lambda_1\lambda_4)\otimes(1+\lambda_1\lambda_2\lambda_5)}{(1-\lambda_7)\otimes(1-\lambda_1\lambda_4)\otimes(1+\lambda_1\lambda_2\lambda_5)}$$

$$L_{01}^{(2)} = L_1 + \log \frac{1+\lambda_2\lambda_3\lambda_6+\lambda_2\lambda_5+\lambda_3\lambda_5\lambda_6}{1+\lambda_2\lambda_3\lambda_6-\lambda_2\lambda_5-\lambda_3\lambda_5\lambda_6}$$

$$= L_0 + \log \frac{(1+\lambda_2\lambda_5)\otimes(1+\lambda_2\lambda_3\lambda_6)}{(1-\lambda_2\lambda_5)\otimes(1+\lambda_2\lambda_3\lambda_6)}$$

$$\lambda_{01}^{(2)} = \frac{\lambda_1+\lambda_2\lambda_5+\lambda_1\lambda_2\lambda_3\lambda_6+\lambda_3\lambda_5\lambda_6}{1+\lambda_1\lambda_2\lambda_5+\lambda_2\lambda_3\lambda_6+\lambda_1\lambda_3\lambda_5\lambda_6}$$

iteration #3 – Substituting $\lambda_{01}^{(2)}$ for λ_1 in (1):

$$L_0^{(3)} = L_0 + \log \frac{(1+\lambda_7)\otimes(1+\lambda_1\lambda_4)\otimes(1+\lambda_1\lambda_2\lambda_5)\otimes(1+\lambda_2\lambda_3\lambda_6)}{(1-\lambda_7)\otimes(1-\lambda_1\lambda_4)\otimes(1+\lambda_1\lambda_2\lambda_5)\otimes(1+\lambda_2\lambda_3\lambda_6)}$$