

EECE 580B
Modern Coding Theory
Bitwise ML Decoder

Tomáš Filler and Jessica Fridrich

Dept. of Electrical and Computer Engineering, SUNY Binghamton



State University of New York

(Blockwise) ML Decoding

- C ... linear code
- $x = (x_1, \dots, x_n)$... sent codeword
- $y = (y_1, \dots, y_n)$... received word - constant

(Blockwise) Maximum Likelihood (ML) decoder:

- the best possible decoder
- minimizes the probability of block error P_B

$$\hat{x} = \arg \max_{x \in C} P(x|y)$$

(Blockwise) ML Decoding

- C ... linear code
- $x = (x_1, \dots, x_n)$... sent codeword
- $y = (y_1, \dots, y_n)$... received word - constant

(Blockwise) Maximum Likelihood (ML) decoder:

- the best possible decoder
- minimizes the probability of block error P_B

$$\hat{x} = \arg \max_{x \in C} P(x|y)$$

Calculation:

$$\hat{x} \stackrel{(1)}{=} \arg \max_{x \in C} \frac{P(y|x)P(x)}{P(y)} \stackrel{(2)}{=} \arg \max_{x \in C} P(y|x)P(x) \stackrel{(3)}{=} \arg \max_{x \in C} P(y|x)$$

Blockwise / Bitwise ML Decoding

(Blockwise) Maximum Likelihood (ML) decoder:

- the best possible decoder
- minimizes the probability of block error P_B

$$\hat{x} = \arg \max_{x \in C} P(x|y)$$

Find the most probable codeword, \hat{x} always $\in C$.

(Bitwise) Maximum Likelihood (ML) decoder:

- suboptimal decoder
- efficient to calculate (approximate) in practice

$$\hat{x}_i = \arg \max_{x_i \in \{0,1\}} P(x_i|y) \quad \forall i = 1, \dots, n$$

The most probable bit-by-bit decoding, \hat{x} does not need to $\in C$.

Bitwise ML Decoding (cont.)

$$\hat{x}_i = \arg \max_{x_i \in \{0,1\}} P(x_i|y)$$

$$\stackrel{(1)}{=} \arg \max_{x_i \in \{0,1\}} \sum_{\substack{x_j \\ j \neq i}} P(x|y)$$

$$\stackrel{(2)}{=} \arg \max_{x_i \in \{0,1\}} \sum_{\substack{x_j \\ j \neq i}} \frac{P(y|x)P(x)}{P(y)}$$

$$\stackrel{(3)}{=} \arg \max_{x_i \in \{0,1\}} \sum_{\substack{x_j \\ j \neq i}} P(y|x) \mathbb{1}_{\{x \in C\}}$$

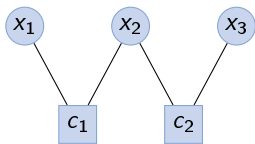
$$\stackrel{(4)}{=} \arg \max_{x_i \in \{0,1\}} \sum_{\substack{x_j \\ j \neq i}} \prod_{l=1}^n P(y_l|x_l) \mathbb{1}_{\{x \in C\}}$$

$\mathbb{1}_{\{x \in C\}} = 1$ if $x \in C$ and 0 otherwise - code membership fn.

Example 1: Repetition Code R_3 over BSC(0.1)

$$y = (1, 1, 0), P(y_l = x|x_l = x) = 0.9, P(y_l = 1 - x|x_l = x) = 0.1$$

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



$$f_1(x) = P(y_1|x), f_2(x) = P(y_2|x), f_3(x) = P(y_3|x)$$

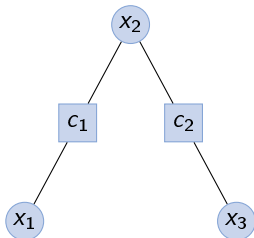
$$\begin{aligned} \hat{x}_2 &= \arg \max_{x_2 \in \{0,1\}} \sum_{x_1, x_3} f(x_1)f(x_2)f(x_3) \mathbb{1}_{\{x \in C\}} \\ &= \arg \max_{x_2 \in \{0,1\}} \sum_{x_1, x_3} f(x_1)f(x_2)f(x_3) \mathbb{1}_{\{x_1+x_2=0\}} \mathbb{1}_{\{x_2+x_3=0\}} \\ &= \arg \max_{x_2 \in \{0,1\}} f_2(x_2) \left(\sum_{x_1} f_1(x_1) \mathbb{1}_{\{x_1+x_2=0\}} \right) \left(\sum_{x_3} f_3(x_3) \mathbb{1}_{\{x_2+x_3=0\}} \right) \end{aligned}$$

Example 1: Repetition Code R_3 (cont.)

$y = (1, 1, 0)$, $P(y_l = x | x_l = x) = 0.9$, $P(y_l = 1 - x | x_l = x) = 0.1$
 $f_1(\cdot) = P(y_1 | \cdot) = f_2(\cdot) = P(y_2 | \cdot) = (.1, .9)$, $f_3(\cdot) = P(y_3 | \cdot) = (.9, .1)$

How to evaluate

$$\hat{x}_2 = \arg \max_{x_2 \in \{0,1\}} f_2(x_2) \left(\sum_{x_1} f_1(x_1) \mathbb{1}_{\{x_1+x_2=0\}} \right) \left(\sum_{x_3} f_3(x_3) \mathbb{1}_{\{x_2+x_3=0\}} \right)$$

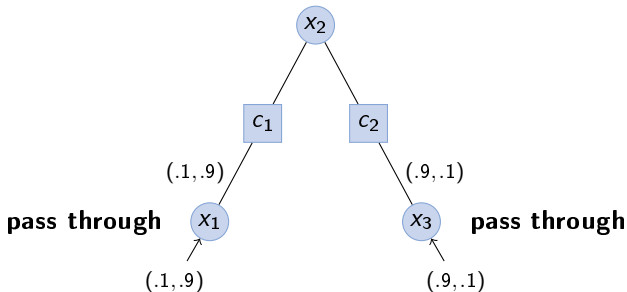


Example 1: Repetition Code R_3 (cont.)

$$f_1(\cdot) = P(y_1|\cdot) = f_2(\cdot) = P(y_2|\cdot) = (.1, .9) , f_3(\cdot) = P(y_3|\cdot) = (.9, .1)$$

Evaluating \hat{x}_2

$$\hat{x}_2 = \arg \max_{x_2 \in \{0,1\}} f_2(x_2) \left(\sum_{x_1} \underbrace{f_1(x_1)}_{(.1, .9)} \mathbb{1}_{\{x_1+x_2=0\}} \right) \left(\sum_{x_3} \underbrace{f_3(x_3)}_{(.9, .1)} \mathbb{1}_{\{x_2+x_3=0\}} \right)$$

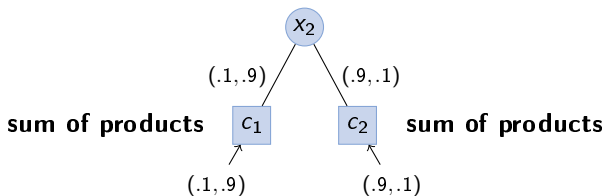


Example 1: Repetition Code R_3 (cont.)

$$f_1(\cdot) = P(y_1|\cdot) = f_2(\cdot) = P(y_2|\cdot) = (.1, .9) , f_3(\cdot) = P(y_3|\cdot) = (.9, .1)$$

Evaluating \hat{x}_2

$$\hat{x}_2 = \arg \max_{x_2 \in \{0,1\}} f_2(x_2) \left(\underbrace{\sum_{x_1} \underbrace{f_1(x_1)}_{(.1, .9)} \mathbb{1}_{\{x_1+x_2=0\}}}_{(.1, .9)} \right) \left(\underbrace{\sum_{x_3} \underbrace{f_3(x_3)}_{(.9, .1)} \mathbb{1}_{\{x_2+x_3=0\}}}_{(.9, .1)} \right)$$



Example 1: Repetition Code R_3 (cont.)

$$f_1(\cdot) = P(y_1|\cdot) = f_2(\cdot) = P(y_2|\cdot) = (.1, .9) , f_3(\cdot) = P(y_3|\cdot) = (.9, .1)$$

Evaluating \hat{x}_2

$$\hat{x}_2 = \arg \max_{x_2 \in \{0,1\}} \overbrace{f_2(x_2)}^{(.1, .9)} \overbrace{\left(\sum_{x_1} f_1(x_1) \mathbb{1}_{\{x_1+x_2=0\}} \right)}^{(.1, .9)} \overbrace{\left(\sum_{x_3} f_3(x_3) \mathbb{1}_{\{x_2+x_3=0\}} \right)}^{(.9, .1)}$$

(.009, .081)

(.1, .9)

↓

x_2 the final product

(.1, .9) (.9, .1)

Final result: $\hat{x}_2 = 1$

Example 1: Repetition Code R_3 (cont.)

$$f_1(\cdot) = P(y_1|\cdot) = f_2(\cdot) = P(y_2|\cdot) = (.1, .9) , f_3(\cdot) = P(y_3|\cdot) = (.9, .1)$$

How to evaluate

$$\hat{x}_1 = \arg \max_{x_1 \in \{0,1\}} \sum_{x_2, x_3} f_1(x_1) f_2(x_2) f_3(x_3) \mathbb{1}_{\{x_1+x_2=0\}} \mathbb{1}_{\{x_2+x_3=0\}}$$

?

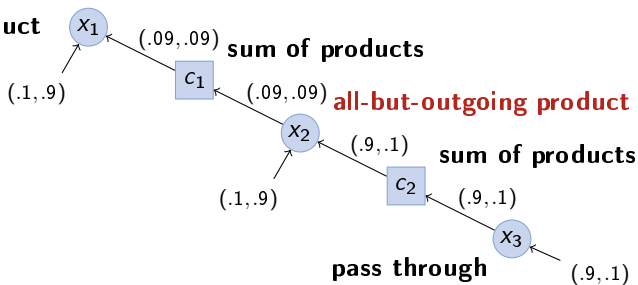
Example 1: Repetition Code R_3 (cont.)

$$f_1(\cdot) = P(y_1|\cdot) = f_2(\cdot) = P(y_2|\cdot) = (.1, .9) , f_3(\cdot) = P(y_3|\cdot) = (.9, .1)$$

Evaluating \hat{x}_1 : (overlines denote the order of evaluation)

$$\hat{x}_1 = \arg \max_{x_1 \in \{0,1\}} f_1(x_1) \underbrace{\left[\sum_{x_2} f_2(x_2) \left(\overline{\overline{\sum_{x_3} f_3(x_3) \mathbb{1}_{\{x_2+x_3=0\}}}} \right) \mathbb{1}_{\{x_1+x_2=0\}} \right]}_{(.009, .081)}$$

the final product



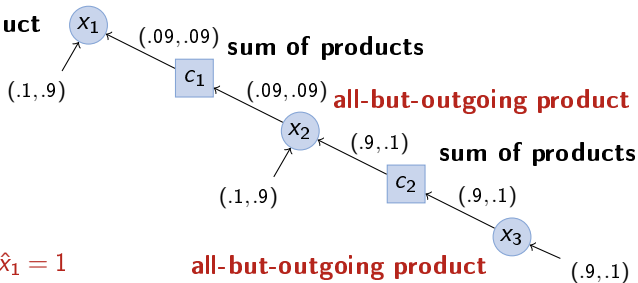
Example 1: Repetition Code R_3 (cont.)

$$f_1(\cdot) = P(y_1|\cdot) = f_2(\cdot) = P(y_2|\cdot) = (.1, .9) , f_3(\cdot) = P(y_3|\cdot) = (.9, .1)$$

Evaluating \hat{x}_1 : (overlines denote the order of evaluation)

$$\hat{x}_1 = \arg \max_{x_1 \in \{0,1\}} f_1(x_1) \underbrace{\left[\sum_{x_2} \overline{f_2(x_2)} \left(\overline{\sum_{x_3} \overline{f_3(x_3)} \mathbb{1}_{\{x_2+x_3=0\}} \right)} \mathbb{1}_{\{x_1+x_2=0\}} \right]}_{(.009, .081)}$$

the final product



Final result: $\hat{x}_1 = 1$

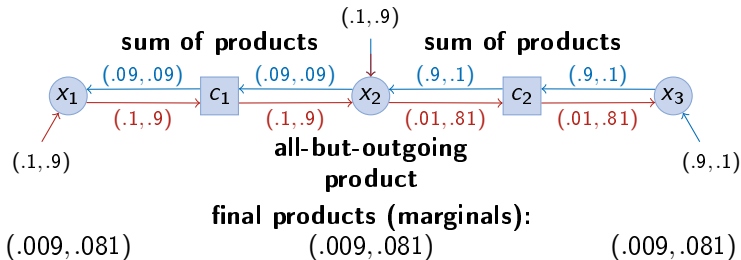
all-but-outgoing product

Example 1: Evaluating All \hat{x}_i s in Parallel

$$\hat{x}_1 = \arg \max_{x_1 \in \{0,1\}} f_1(x_1) \left[\sum_{x_2} f_2(x_2) \left(\overline{\sum_{x_3} f_3(x_3) \mathbb{1}_{\{x_2+x_3=0\}}} \right) \mathbb{1}_{\{x_1+x_2=0\}} \right]$$

$$\hat{x}_2 = \arg \max_{x_2 \in \{0,1\}} f_2(x_2) \left(\sum_{x_1} f_1(x_1) \mathbb{1}_{\{x_1+x_2=0\}} \right) \left(\sum_{x_3} f_3(x_3) \mathbb{1}_{\{x_2+x_3=0\}} \right)$$

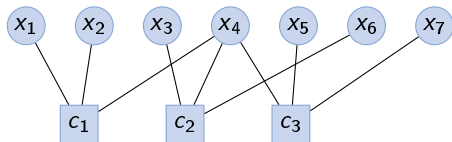
$$\hat{x}_3 = \arg \max_{x_3 \in \{0,1\}} f_3(x_3) \left[\sum_{x_2} f_2(x_2) \left(\sum_{x_1} f_1(x_1) \mathbb{1}_{\{x_1+x_2=0\}} \right) \mathbb{1}_{\{x_2+x_3=0\}} \right]$$



Example 2: Other Code Over BSC(0.1)

$y = (0, 1, 1, 0, 0, 0, 0)$, $P(y_l = x | x_l = x) = 0.9$, $P(y_l = 1 - x | x_l = x) = 0.1$

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



$$f_i(x) = P(y_i | x), \quad f_2(\cdot) = f_3(\cdot) = (0.1, 0.9)$$

$$f_1(\cdot) = f_4(\cdot) = f_5(\cdot) = f_6(\cdot) = f_7(\cdot) = (0.9, 0.1)$$

$$\mathbb{1}_{\{x \in C\}} = \mathbb{1}_{\{x_1 + x_2 + x_4 = 0\}} \mathbb{1}_{\{x_3 + x_4 + x_6 = 0\}} \mathbb{1}_{\{x_4 + x_5 + x_7 = 0\}}$$

$$\hat{x}_1 = \arg \max_{x_1 \in \{0,1\}} \sum_{x_2, \dots, x_7} \prod_{i=1}^7 f_i(x_i) \mathbb{1}_{\{x \in C\}}$$

$$\hat{x}_4 = \arg \max_{x_4 \in \{0,1\}} \sum_{x_1, x_2, x_3, x_5, x_6, x_7} \prod_{i=1}^7 f_i(x_i) \mathbb{1}_{\{x \in C\}}$$

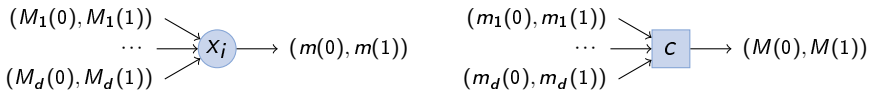
Belief Propagation Algorithm

Initialization:

$(1,1) \longrightarrow X$ set all check-to-variable messages to $(1,1)$

Node processing update rules:

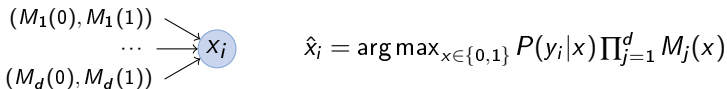
Update all variable-to-check and then all check-to-variable messages.



$$m(x) = P(y_i|x) \prod_{j=1}^d M_j(x) \qquad M(x) = \sum_{x_1, \dots, x_d} \mathbb{1}_{\{x_1 + \dots + x_d = x\}} \prod_{i=1}^d m_i(x_i)$$

Repeat these node processing update rules as many times as needed.

Bitwise ML estimate of bit x_i :



Simplification of the Messages

Message update rules can be simplified to only track one number, log-likelihood ratio

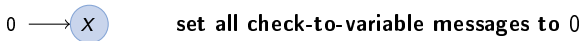
$$l = \ln r = \ln \frac{m(0)}{m(1)}$$

Derivation ...

Belief Propagation Algorithm

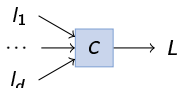
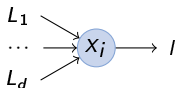
Log-likelihood Ratio Domain

Initialization:



Node processing update rules:

Update all variable-to-check and then all check-to-variable messages.

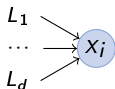


$$I = \ln \frac{P(y_i|0)}{P(y_i|1)} + \sum_{j=1}^d L_j$$

$$L = 2 \tanh^{-1} \left(\prod_{i=1}^d \tanh(l_i/2) \right)$$

Repeat these node processing update rules as many times as needed.

Bitwise ML estimate of bit x_j :



$$\hat{x}_i = \begin{cases} 0 & \text{if } \ln \frac{P(y_i|0)}{P(y_i|1)} + \sum_{j=1}^d L_j > 0 \\ 1 & \text{otherwise} \end{cases}$$