Section 7
Information Theory

Very brief introduction to information theory

Entropy

Definition 7.1

The entropy $H(X)$ of a discrete random variable X is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = \mathrm{E}_X \left[-\log p(x) \right].$$

It is the uncertainty of a random variable.

Example of Entropy

Example 1: Let $X = \begin{cases} 1 & \text{with prob. } \frac{1}{2} \\ 0 & \text{with prob. } \frac{1}{2} \end{cases}$ and $Y = \begin{cases} 100 & \text{with prob. } \frac{1}{2} \\ -100 & \text{with prob. } \frac{1}{2} \end{cases}$.

Find $H(X)$ and $H(Y)$.

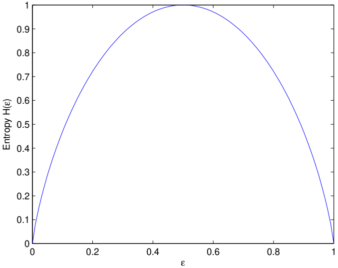
Answer: $H(X) = 1$ and $H(Y) = 1$.

Example 2: A binary r.v. X with $p_X(x = 1) = \epsilon$ and $p_X(x = 0) = 1 - \epsilon$.

Find $H(X)$.

Answer:

$$H(X) = -\epsilon \log_2 \epsilon - (1 - \epsilon) \log_2 (1 - \epsilon).$$



Mutual Information

Definition 7.2

The mutual information between two r.v. X and Y is defined as

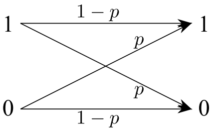
$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)}$$
$$= H(X) - H(X|Y) = H(Y) - H(Y|X),$$

where the conditional entropy
$H(X|Y) := \mathrm{E}_Y [H(X|y)] = \mathrm{E}_Y \left[\mathrm{E}_{X|Y} [-\log p(x|y)]\right]$.

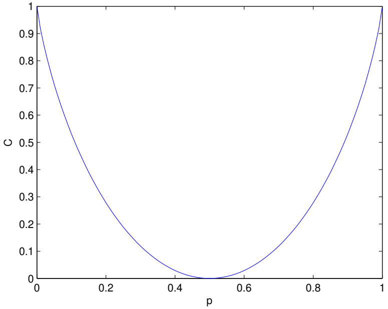
It is the entropy reduction due to the knowledge of the other r.v.

- If X and Y are independent: $I(X;Y) = 0$.
- If there is an invertible mapping between X and Y,
  $I(X;Y) = H(X) = H(Y)$.

Example: BSC



$$I(X;Y) = H(Y) - H(Y|X) = 1 - H(p).$$

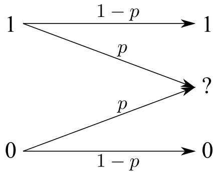


Computation of Mutual Information: BSC

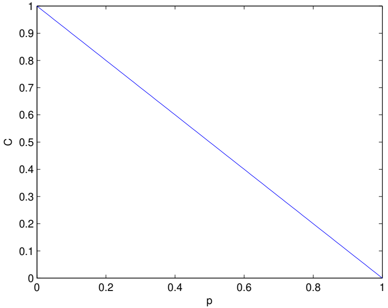
$$I(X;Y) = H(X) - H(X|Y)$$

- Computation of $p_{X|Y}(x|y)$:
  - $p_{X|Y}(0|0) = \frac{\frac{1}{2}(1-p)}{\frac{1}{2}(1-p)+\frac{1}{2}p} = 1 - p$.
  - $p_{X|Y}(1|0) = \frac{\frac{1}{2}p}{\frac{1}{2}(1-p)+\frac{1}{2}p} = p$.
- $H(X|y=0) = H(p)$.
- $H(X|Y) = \frac{1}{2}H(X|y=0) + \frac{1}{2}H(X|y=1) = H(p)$.
- $I(X;Y) = 1 - H(p)$.

Example: BEC



$$I(X;Y) = 1 - p.$$



Computation of Mutual Information: BEC

$$I(X;Y) = H(X) - H(X|Y)$$

- Computation of $p_{X|Y}(x|y)$:
  - $p_{X|Y}(0|0) = 1$, $p_{X|Y}(1|0) = 0$.
  - $p_{X|Y}(0|1) = 0$, $p_{X|Y}(1|1) = 1$.
  - $p_{X|Y}(0|?) = \frac{1}{2}$, $p_{X|Y}(1|?) = \frac{1}{2}$.
- $H(X|y=0) = 0$, $H(X|y=1) = 0$, $H(X|y=?) = 1$.
- $H(X|Y) = p$.
- $I(X;Y) = 1 - p$.

Channel Capacity

Definition 7.3

For a communication channel with input X and output Y, we define the capacity C by

$$C = \max_{p(x)} I\left(X;Y\right).$$

For BSCs and BECs, the optimal $p\left(x\right)$ is given by $\epsilon = \frac{1}{2}$.

Shannon's Coding Theorem

Theorem 7.4

All rates below capacity C are achievable, that is, for every rate $r < C$, there exists a sequence of (n, rn)-codes with probability of decoding error $\lambda^{(n)} \to 0$.
Conversely, any sequence of (n, rn)-codes with $\lambda^{(n)} \to 0$ must have $r \leq C$.

C.E. Shannon, A mathematical theory of communication, Bell Sys. Tech. Journal, 27:379-423, 623-656, 1948.

Encoding: Random codes with $n \to \infty$.
Decoding: Based on "jointly typical sequences".

In practice,

- Want low encoding and decoding complexity (use structures).
- A shift from worst case analysis to average case analysis.
  - Abandom the minimum distance as a design criterion.

Section 8
LDPC Codes

Two Decoding Approaches: Complexity Matters

Maximum likelihood decoding:

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in \mathcal{C}} p\left(\boldsymbol{y}|\boldsymbol{x}\right).$$

- Optimal in terms of error probability.
- Complexity is high.

Bit-MAP decoding (MAP decoding bit by bit):

$$\hat{x}_i \quad = \arg\max_{a \in \{0,1\}} p\left(x_i = a|\boldsymbol{y}\right).$$

- Sub-optimal but near-optimal when carefully designed.
- Complexity is small.

Example: for binary codes,

$$\hat{x}_i = \begin{cases} 0 & \text{if } p\left(x_i = 0|\boldsymbol{y}\right) > p\left(x_i = 1|\boldsymbol{y}\right), \\ 1 & \text{otherwise.} \end{cases}$$

The Distributive Law

$$a_1 b_1 + a_1 b_2 + a_2 b_1 + a_2 b_2 = (a_1 + a_2)(b_1 + b_2)$$

4 multiplications, 3 additions → 1 multiplications, 2 additions.

$$\sum_{i,j,k,\ell=0}^{1} a_i b_j c_k d_\ell = \left(\sum_{i=0}^{1} a_i\right) \cdots \left(\sum_{\ell=0}^{1} d_\ell\right)$$

$2^4 \cdot 3$ multiplications, $2^4 - 1$ additions → 3 multiplication, 4 additions.

From the sum of products to the product of sums,
computational complexity is highly reduced.

More on Bit-MAP Decoding

$$
\begin{aligned}
p\left(x_i = a|\boldsymbol{y}\right) &\overset{(a)}{=} \sum_{\boldsymbol{x},\, x_i=a} p\left(\boldsymbol{x}|\boldsymbol{y}\right) \\
&\overset{(b)}{\propto} \sum_{\boldsymbol{x},\, x_i=a} p\left(\boldsymbol{y}|\boldsymbol{x}\right) p\left(\boldsymbol{x}\right) \\
&\overset{(c)}{\propto} \sum_{\boldsymbol{x}\in\mathcal{C},\, x_i=a} p\left(\boldsymbol{y}|\boldsymbol{x}\right) \\
&\overset{(d)}{=} \sum_{\boldsymbol{x}\in\mathcal{C},\, x_i=a} \prod_j p\left(y_j|x_j\right),
\end{aligned}
$$

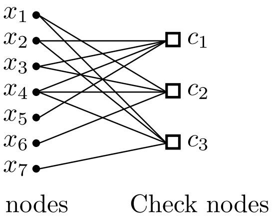
where (a) follows the definition of the marginal probability, (b) is derived by the Bayes rule, (c) comes from the assumption that $p\left(\boldsymbol{x} \notin \mathcal{C}\right) = 0$ and every codeword in the codebook has the equal probability, and (d) follows from memoryless channel model.

sum of products $\overset{?}{\to}$ product of sums.

Tanner graph

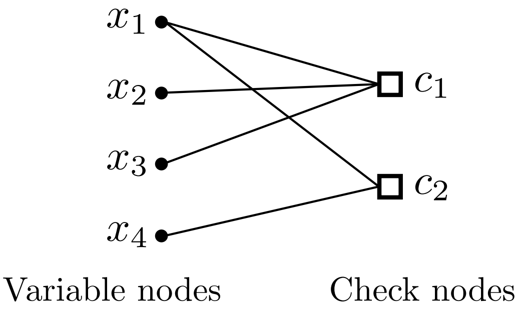
Tanner graph is a (bipartite) graph of which $\boldsymbol{H}$ is the adjacent matrix.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \Rightarrow$$



Two vertices connected by an edge are called adjacent.
Degree of a vertex = # of edges connected to it.

An Example Based on Tanner Graph



An fact in probability theory:

$$\text{If } A \Leftrightarrow B, \text{ then } p(A) = p(A, B) = p(B).$$

Remark: If $A \Leftrightarrow B$, then A and B are the same set. $A = A \bigcap B = B$.

Bit-MAP Decoding

$$\begin{aligned}
&p\left(x_1 = 1 | y_1 \cdots y_4\right) \\
&= p\left(x_1 = 1, x_2 + x_3 = 1, x_4 = 1 | y_1 \cdots y_4\right) \\
&\propto p\left(y_1 \cdots y_4 | x_1 = 1, x_2 + x_3 = 1, x_4 = 1\right) \\
&= p\left(y_1 | x_1 = 1\right) p\left(y_2, y_3 | x_2 + x_3 = 1\right) p\left(y_4 | x_4 = 1\right),
\end{aligned}$$

where

$$\begin{aligned}
&p\left(y_2 y_3 | x_2 + x_3 = 1\right) \\
&= p\left(y_2 | x_2 = 0\right) p\left(y_3 | x_3 = 1\right) + p\left(y_2 | x_2 = 1\right) p\left(y_3 | x_3 = 0\right).
\end{aligned}$$

Only 1 addition and 4 multiplications.

The Direct Computation

Example: Consider a code $\mathcal{C} \subset \mathbb{F}_2^4$. Want to find $p\left(x_1 = 1 | \boldsymbol{y}\right)$.

$$
\begin{aligned}
& p\left(x_1 = 1 | \boldsymbol{y}\right) \\
& = \sum_{x_2=0}^{1} \sum_{x_3=0}^{1} \sum_{x_4=0}^{1} \; p\left(\boldsymbol{x} = [1, x_2, x_3, x_4] \,| \boldsymbol{y}\right) \delta_{[1,x_2,x_3,x_4] \in \mathcal{C}} \\
& \propto \sum_{x_2=0}^{1} \sum_{x_3=0}^{1} \sum_{x_4=0}^{1} \; p\left(y_1 | x_1 = 1\right) \prod_{i=2}^{4} p\left(y_i | x_i\right) \delta_{[1,x_2,x_3,x_4] \in \mathcal{C}}.
\end{aligned}
$$

Totally 8 additions and $8 \times 3 = 24$ multiplications.

Parallel Processing

Parallel Processing: Message Passing

- Compute the probabilities in parallel.
- Only involves local operations.
- Good for FPGA implementation.

Message Passing:

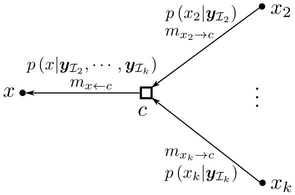
Messages $\equiv$ Probabilities

Initialization

At each variable node x, compute

$$m_x = p\left(x|y\right) \propto p\left(y|x\right).$$

MP at Check Nodes (1)

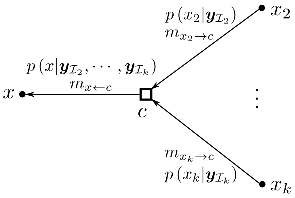


$$p(x=0|\boldsymbol{y}) \propto p(y|x=0)\, p\left(\boldsymbol{y}_{\mathcal{I}_2},\cdots,\boldsymbol{y}_{\mathcal{I}_k} \Big| \sum_{x_i,2\le i\le k} x_i = 0\right)$$

$$= p(y|x=0) \sum_{\sum x_i=0} \prod_{i=2}^{k} p(\boldsymbol{y}_{\mathcal{I}_i}|x_i)$$

$$\propto p(x=0|y) \sum_{\sum x_i=0} \prod_{i=2}^{k} p(x_i|\boldsymbol{y}_{\mathcal{I}_i}).$$

MP at Check Nodes (2)

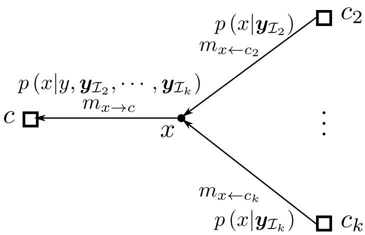


$$m_{x\leftarrow c} = \sum_{\sum x' = -x} \prod_{x'\in\mathcal{X}_c\setminus\{x\}} m_{x'\to c}.$$

Example:

$$p(x=0|\boldsymbol{y}_{\mathcal{I}_2},\boldsymbol{y}_{\mathcal{I}_3}) = p(x_2+x_3=0|\boldsymbol{y}_{\mathcal{I}_2},\boldsymbol{y}_{\mathcal{I}_3})$$
$$= p(x_2=0|\boldsymbol{y}_{\mathcal{I}_2})\,p(x_3=0|\boldsymbol{y}_{\mathcal{I}_3}) + p(x_2=1|\boldsymbol{y}_{\mathcal{I}_2})\,p(x_3=1|\boldsymbol{y}_{\mathcal{I}_3})$$

MP at Variable Nodes

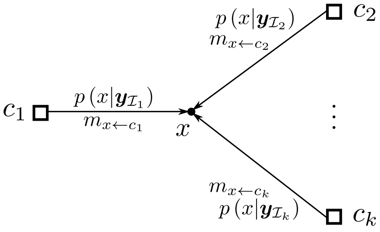


$$\begin{aligned} p\left(x|y,\boldsymbol{y}_{\mathcal{I}_2},\cdots,\boldsymbol{y}_{\mathcal{I}_k}\right) &\propto p\left(y|x\right)p\left(\boldsymbol{y}_{\mathcal{I}_2}|x\right)\cdots p\left(\boldsymbol{y}_{\mathcal{I}_k}|x\right) \\ &\propto p\left(x|y\right)p\left(x|\boldsymbol{y}_{\mathcal{I}_2}\right)\cdots p\left(x|\boldsymbol{y}_{\mathcal{I}_k}\right). \end{aligned}$$

General rule:

$$m_{x\to c} = m_x \prod_{c'\in\mathcal{C}_x\backslash c} m_{x\leftarrow c'}.$$

Decoding at Variable Nodes



$$p\left(x|y,\boldsymbol{y}_{\mathcal{I}_1},\cdots,\boldsymbol{y}_{\mathcal{I}_k}\right)\propto p\left(x|y\right)p\left(x|\boldsymbol{y}_{\mathcal{I}_1}\right)\cdots p\left(x|\boldsymbol{y}_{\mathcal{I}_k}\right).$$

General rule:

$$J_x = m_x \prod_{c\in\mathcal{C}_x} m_{x\leftarrow c}.$$

Message Passing: An Overview

Received $\boldsymbol{y}$ from a binary input memoryless channel.

Initialization:

For all variable nodes, compute $m_x = \Pr(x|y_x)$.

Iterations: $t = 1, 2, \cdots$

$x \to c$: $$m_{x\to c}^{(t)} = \begin{cases} m_x & \text{if } t = 1 \\ m_x \cdot \prod_{c' \in \mathcal{C}_x \setminus \{c\}} m_{x \leftarrow c'}^{(t-1)} & \text{if } t \geq 1 \end{cases}$$

$x \leftarrow c$: $$m_{x\leftarrow c}^{(t)} = \sum_{\sum x' = -x} \prod_{x' \in \mathcal{X}_c \setminus \{x\}} m_{x' \to c}^{(t)}.$$

Dec: $$J_x = m_x \cdot \prod_{c \in \mathcal{C}_x} m_{x\leftarrow c}^{(t)}$$
$$\hat{x} = \arg \max_a J_{x=a}$$

Terminate:

Up to T iterations or found a codeword.

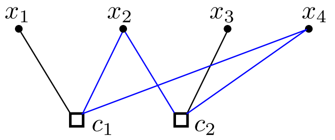
Complexity

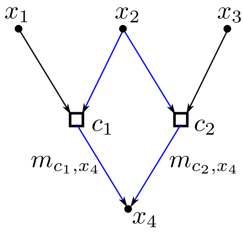
- Computation at variable nodes.
  - Multiplication only: complexity $\approx |\mathcal{C}_x| = d_x$.
- Computation at check nodes.
  - Sum of many terms: let $d_c = |\mathcal{X}_c|$.
    $\binom{d_c-1}{0} + \binom{d_c-1}{2} + \binom{d_c-1}{4} + \cdots$
  - Want d_c be small ($d_c \le 6$).
- Overall complexity.
  $n\left(d_{x,\max} + 2^{d_{c,\max}}\right) = O\left(n\right)$ when $d_{c,\max}$ is small

LDPC (Low-Density Parity-Check) codes:

- d_c and d_x are small.
- regular if $d_{c_1} = \cdots = d_{c_{n-k}} = d_c$ and $d_{x_1} = \cdots = d_{x_n} = d_x$.
- Example: a $(d_x = 3, d_c = 6)$ regular LDPC code.

Performance - When the Computation is not Precise





True Bit-MAP decoding:
Want to compute $\Pr(x_4|y_1y_2y_3y_4)$.

Using message passing:

$$\begin{aligned} J_{x_4} &= m_{x_4} \cdot m_{c_1,x_4} \cdot m_{c_2,x_4} \\ &= \Pr(x_4|y_4)\Pr(x_4|y_1y_2)\Pr(x_4|y_2y_3). \end{aligned}$$

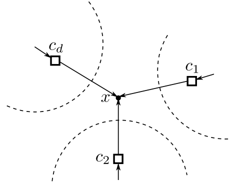
In some sense, we are computing $\Pr\left(x_4|y_1y_2^2y_3y_4\right)$:

$\Pr(x_2|y_2)$ has been used more than it should.

The Performance of Message Passing

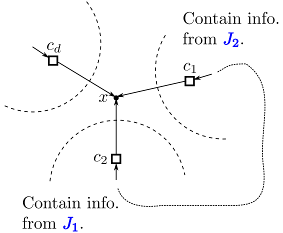
Computation is exact when the graph forms a tree (no cycles).

It becomes problematic if $\exists$ cycles.

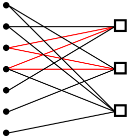


$L(x_i|y) = L(x_i|y_i) \prod_t L(x_i|\boldsymbol{y}_{J_t})$

J_t's are disjoint.

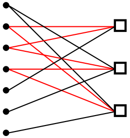


Examples of cycles:



Girth = 4

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



Girth = 6

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

$\boldsymbol{H}$ is sparser $\Rightarrow$ less probability to have short cycles.

Simplification for BSC

- Define conditional likelihood $L(x|y) := p(x=0|y)/p(x=1|y)$.
  - $p(x=0|y) = \frac{L(x|y)}{1+L(x|y)}$.
  - $p(x=1|y) = \frac{1}{1+L(x|y)}$.
  - Conditional independence:
    - $L(x_1,\cdots,x_d|y_1,\cdots,y_d) = \prod_i L(x_i|y_i)$
    - Equiprobable assumption: $\Pr(x=1) = \Pr(x=0) = 0.5$.
- Conditional log-likelihood ratio of x given y.
  - $\ln L(x|y) = \ln \frac{p(x=0|y)}{p(x=1|y)}$.
  - $\hat{x} = 0$ if $\ln L(x|\boldsymbol{y}) > 0$ and $\hat{x} = 1$ if $\ln L(x|\boldsymbol{y}) < 0$.

Messages at Variable Nodes

Initialization:

$$m_x = \ln \frac{p(x=0|y)}{p(x=1|y)} = \ln L\left(x|y\right).$$

Iterations:

$$\begin{aligned} m_{x\to c} &= \ln \frac{p\left(x=0|y, y_{\mathcal{I}_2}, \cdots, y_{\mathcal{I}_k}\right)}{p\left(x=1|y, y_{\mathcal{I}_2}, \cdots, y_{\mathcal{I}_k}\right)} \\ &= \ln \frac{p\left(x=0|y\right) p\left(x=0|y_{\mathcal{I}_2}\right) p\left(x=0|y_{\mathcal{I}_k}\right)}{p\left(x=1|y\right) p\left(x=1|y_{\mathcal{I}_2}\right) p\left(x=1|y_{\mathcal{I}_k}\right)} \\ &= m_x + \sum_{c'\in\mathcal{C}_x\backslash\{c\}} m_{x\leftarrow c'}. \end{aligned}$$

Decoding:

$$J_x = m_x + \sum_{c\in\mathcal{C}_x} m_{x\leftarrow c}.$$

Messages at Check Nodes

$$m_{x\leftarrow c} = \ln \frac{p\left(x=0|y_{\mathcal{I}_2},\cdots y_{\mathcal{I}_2}\right)}{p\left(x=1|y_{\mathcal{I}_2},\cdots y_{\mathcal{I}_2}\right)}$$

$$= \ln \frac{p\left(\sum_{x'\in\mathcal{X}_c\backslash\{x\}} x'=0|y_{\mathcal{I}_2},\cdots y_{\mathcal{I}_2}\right)}{p\left(\sum_{x'\in\mathcal{X}_c\backslash\{x\}} x'=1|y_{\mathcal{I}_2},\cdots y_{\mathcal{I}_2}\right)}.$$

Theorem 8.1

Let $L_i = L\left(x_i|y_i\right)$, $1\le i\le d$, *and* $m_i = \ln L_i$. *Then*

$$\ln L\left(x_1+\cdots+x_d|y_1\cdots y_d\right) = \ln \frac{p\left(\sum x_i=0|y_1\cdots y_d\right)}{p\left(\sum x_i=1|y_1\cdots y_d\right)}$$

$$= \ln \frac{1+\prod_{i=1}^d \tanh\left(m_i/2\right)}{1-\prod_{i=1}^d \tanh\left(m_i/2\right)}.$$

A Useful Lemma

Lemma 8.2

$2 \cdot p\left(\sum_{i=1}^{d} x_i = 0 | \boldsymbol{y}_{1:d}\right) - 1 = \prod_{i=1}^{d} \left(2 \cdot p\left(x_i = 0 | y_i\right) - 1\right).$

Proof: Let $d = 2$.
Let $p = 2 \cdot p\left(x_1 = 0 | y_1\right) - 1$ and $q = 2 \cdot p\left(x_2 = 0 | y_2\right) - 1$.
Then $p\left(x_1 = 1 | y_1\right) = 1 - p\left(x_1 = 0 | y_1\right) = 1 - \frac{1+p}{2} = \frac{1-p}{2}$.
Similarly, $p\left(x_2 = 1 | y_2\right) = \frac{1-q}{2}$.
Then

$$\begin{aligned} & p\left(x_1 + x_2 = 0 | y_1 y_2\right) \\ & = p\left(x_1 = 0 | y_1\right) p\left(x_2 = 0 | y_2\right) + p\left(x_1 = 1 | y_1\right) p\left(x_2 = 1 | y_2\right) \\ & = \frac{1+p}{2} \frac{1+q}{2} + \frac{1-p}{2} \frac{1-q}{2} = \frac{1+pq}{2}. \end{aligned}$$

Hence,

$$2p\left(x_1 + x_2 = 0 | y_1 y_2\right) - 1 = pq,$$

which proves this lemma.

Proof of Theorem 8.1

$$\ln L\left(\sum_{i=1}^{d} x_i | \boldsymbol{y}_{1:d}\right) = \ln \frac{1+\prod_{i=1}^{d}\tanh\left(m_i/2\right)}{1-\prod_{i=1}^{d}\tanh\left(m_i/2\right)}.$$

Proof:

1. $p\left(x_i = 0|y_i\right) = \frac{L_i}{1+L_i} \Rightarrow 2p\left(x_i = 0|y_i\right) - 1 = \frac{L_i - 1}{L_i + 1}$.
2. $2p\left(x_i = 0|y_i\right) - 1 = \frac{L_i - 1}{L_i + 1} = \frac{e^{\ln L_i} - 1}{e^{\ln L_i} + 1} = \tanh\left(m_i/2\right)$.
   2.1 $\tanh\left(x\right) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}$.
3. By Lemma 8.2, $2p\left(\sum_i x_i = 0|\boldsymbol{y}\right) - 1 = \prod_{i=1}^{d}\tanh\left(m_i/2\right)$.
4. Hence
   $p\left(\sum_i x_i = 0|\boldsymbol{y}\right) = \frac{1}{2}\left(1 + \prod\tanh\left(m_i/2\right)\right)$.
   $p\left(\sum_i x_i = 1|\boldsymbol{y}\right) = 1 - p\left(\sum_i x_i = 0|\boldsymbol{y}\right) = \frac{1}{2}\left(1 - \prod\tanh\left(m_i/2\right)\right)$
5. Hence
   $\ln L\left(\sum_{i=1}^{d} x_i|\boldsymbol{y}_{1:d}\right) = \ln \frac{1+\prod_{i=1}^{d}\tanh(m_i/2)}{1-\prod_{i=1}^{d}\tanh(m_i/2)}$.

Belief Propagation Algorithm

Received $\boldsymbol{y}$ from a binary input memoryless channel.

Initialization:

For all variable nodes, compute $m_x = \ln L(x|y)$.

Iterations: $t = 1, 2, \cdots$

$x \to c$: $$m_{x\to c}^{(t)} = \begin{cases} m_x & \text{if } t = 1 \\ m_x + \sum_{c' \in \mathcal{C}_x \setminus \{c\}} m_{x\leftarrow c'}^{(t-1)} & \text{if } t \geq 1 \end{cases}$$

$x \leftarrow c$: $$m_{x\leftarrow c}^{(t)} = \ln \frac{1+\prod_{x' \in \mathcal{X}_c \setminus \{x\}} \tanh\left(m_{x'\to c}^{(t)}/2\right)}{1-\prod_{x' \in \mathcal{X}_c \setminus \{x\}} \tanh\left(m_{x'\to c}^{(t)}/2\right)}$$

Dec: $$J_x = m_x + \sum_{c \in \mathcal{C}_x} m_{x\leftarrow c}^{(t)}$$
$$\hat{x} = \begin{cases} 0 & \text{if } J_x \geq 0 \\ 1 & \text{if } J_x < 0 \end{cases}$$

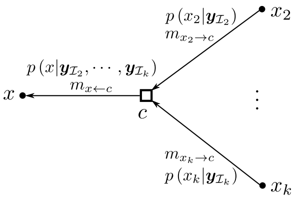
Terminate:

Up to T iterations or found a codeword.

Simplification for BEC

- Three possible cases for $\ln L\left(x_i|y_i\right)$
  - If $y_i = 0$: $\ln L\left(x_i|y_i\right) = \ln \frac{p(x_i=0|y_i=0)}{p(x_i=1|y_i=0)} = +\infty$.
  - If $y_i = 1$: $\ln L\left(x_i|y_i\right) = \ln \frac{p(x_i=0|y_i=1)}{p(x_i=1|y_i=1)} = -\infty$.
  - If $y_i =?$: $\ln L\left(x_i|y_i\right) = \ln \frac{p(x_i=0|y_i=1)}{p(x_i=1|y_i=1)} = \ln 1 = 0$.
- Use three symbols to represent these three events: 0, 1, $?$.
  - That is, $m_{x\to c}, m_{x\leftarrow c} \in \{0, 1, ?\}$.

At the Check Nodes



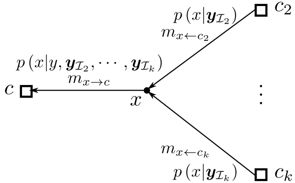
If none of the $m_{x'\to c} = ?$, then

$$m_{x\leftarrow c} = -\sum_{x'\in\mathcal{X}_c\setminus\{x\}} m_{x'\to c}.$$

If some $m_{x'\to c} = ?$, then

$$m_{x\leftarrow c} = ?.$$

At the Variable Nodes



If either $m_x = 1$ or some $m_{x\leftarrow c'} = 1$, then

$$m_{x\rightarrow c} = 1.$$

If either $m_x = 0$ or some $m_{x\leftarrow c'} = 0$, then

$$m_{x\rightarrow c} = 0.$$

If $m_x =?$ or all $m_{x\leftarrow c'} =?$, then

$$m_{x\rightarrow c} =?.$$

If the set $\{m_x, m_{x\leftarrow c'}\}$ contains both 1 or 0,

it is impossible. Return error.

Message Passing for BEC

Iterations: $t = 1, 2, \cdots$

$x \to c$: $m_{x\to c}^{(t)} = m_x$, if $t = 1$. If $t > 1$,

$$m_{x\to c}^{(t)} = \begin{cases} ? & \text{if } \left\{m_x, m_{x\leftarrow c'}^{(t-1)}\right\} \text{ only contains } ? \\ 1 & \text{if } \left\{m_x, m_{x\leftarrow c'}\right\} \text{ contains } 1 \\ 0 & \text{if } \left\{m_x, m_{x\leftarrow c'}^{(t-1)}\right\} \text{ contains } 0 \\ \text{Error} & \text{if } \left\{m_x, m_{x\leftarrow c'}^{(t-1)}\right\} \text{ contains both} 1 \text{ and } 0 \end{cases}$$

$x \leftarrow c$: $$m_{x\leftarrow c}^{(t)} = \begin{cases} ? & \text{if } ? \in \{m_{x'\to c}\} \\ -\sum_{x'\in\mathcal{X}_c\setminus\{x\}} m_{x'\to c} & \text{otherwise} \end{cases}.$$

Analysis for BEC

Assumption:

- Consider a raondom $(d_x = 3,\ d_c = 6)$ regular LDPC codes.
- Let code length $n \to \infty$.
- Assume that the graph is "nearly" cycle-free.

LDPC codes are almost capacity achieving.

- Consider $\mathrm{BEC}\,(\epsilon)$.

Detailed Probability Computation

- At the initialziation stage, for each variable node it holds that
$$p\left(m_x = ?\right) = \epsilon.$$
- At a check node, the value of $m_{x \leftarrow c}$ depends on five input values $m_{x' \rightarrow c}$. Any $m_{x' \rightarrow c} = ?$ implies that $m_{x \leftarrow c} = ?$.
$$p\left(m_{x \leftarrow c} = ?\right) = 1 - \left(1 - \epsilon\right)^5.$$
- At a variable node, $m_{x \rightarrow c} = ?$ if all input messages $m_{x \leftarrow c'}$ and m_x are $?$.
$$p\left(m_{x \rightarrow c} = ?\right) = \epsilon p^2\left(m_{x \leftarrow c} = ?\right) = \epsilon\left[1 - \left(1 - \epsilon\right)^5\right]^2.$$
- $\vdots$

Track the Probability

$p^{(0)} = \epsilon$, $p^{(1)} = \epsilon \left[1 - \left(1 - p^{(0)}\right)^5\right]^2, \cdots$

$$p^{(t)} = \epsilon \left[1 - \left(1 - p^{(t-1)}\right)^5\right]^2$$

Example:

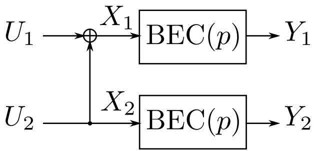
- $\epsilon = 0.4$: $p^{(1)} = 0.3402$, $p^{(2)} = 0.3062$, $\cdots$, $p^{(20)} = 2.76 \times 10^{-21}$.
- The critical value of $\epsilon^* = 0.43$.
  - The $(3,6)$ LDPC codes can recover the codeword with high probability for $\mathrm{BEC}\left(\epsilon\right)$ with $\epsilon < 0.43$.
- Note that the rate of the code is $3/6 = 0.5 \gtrapprox 0.43$. It is almost capacity achieving.

Section 9
Polar Codes

Polar Codes

- Polar codes are provable capacity-achieving for many binary channels, including BEC.
- The construction is deterministic. There is no "choose from an ensemble and verify" step.
- Encoding complexity is $O(n \log n)$.
- Decoding complexity is $O(n \log n)$.
- Block error probability decays as $2^{-\sqrt{n}}$ (provable). The property can be used for very low error probability applications with finite code length.

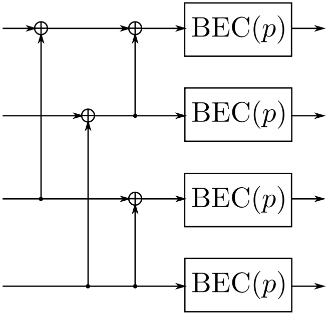
Encoding: Basic Building Block



Mapping matrix from $\boldsymbol{u}$ to $\boldsymbol{x}$:

$$[x_1, x_2] = [u_1, u_2] \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \end{array} \right] = \boldsymbol{u}\boldsymbol{G}_1.$$

Level Two

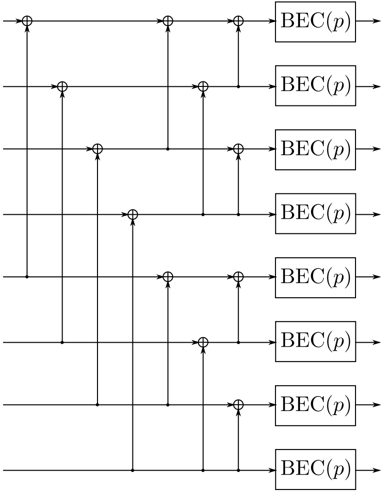


Mapping matrix from $\boldsymbol{u}$ to $\boldsymbol{x}$:

$$\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}_2 = \boldsymbol{u}\begin{bmatrix} \boldsymbol{G}_1 & 0 \\ \boldsymbol{G}_1 & \boldsymbol{G}_1 \end{bmatrix}.$$

Level Three

$$\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G}_3 = \boldsymbol{u}\begin{bmatrix} \boldsymbol{G}_2 & 0 \\ \boldsymbol{G}_2 & \boldsymbol{G}_2 \end{bmatrix}.$$



Recursive Construction of $\boldsymbol{G}_k$

- $\boldsymbol{G}_k$ is a $2^k \times 2^k$ matrix.
- We will delete some of the rows to form the generator matrix for a polar code.
- Carefully choose the rows to be deleted.

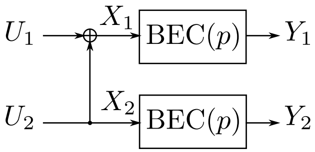
Towards this end, consider successive decoding of the basic building block

- Compute $I(U_1; Y_1 Y_2)$: Decode U_1 based on $y_1 y_2$.
- Compute $I(U_2; U_1 Y_1 Y_2)$: Decode U_2 based on given $u_1 y_1 y_2$.
- Channel Polarization:$I(U_2; U_1 Y_1 Y_2) > I(U_1; Y_1 Y_2)$
    - The channel of U_2 is more reliable than the channel of U_1.

For the level k, find the "unreliable" channels and "delete" the corresponding rows.

Channel Polarization

Consider successive decoding of the basic building block:



- Decode U_1 based on $y_1 y_2$.
  - Channel capacity: $I(U_1; Y_1 Y_2)$:
- Decode U_2 based on given $u_1 y_1 y_2$.
  - Channel capacity $I(U_2; U_1 Y_1 Y_2)$:
- Channel Polarization:$I(U_2; U_1 Y_1 Y_2) > I(U_1; Y_1 Y_2)$
  - The channel of U_2 is more reliable than the channel of U_1.

For the level k, find the "unreliable" channels and "delete" the corresponding rows.

Channel Polarization

- $$I\left(U_1U_2;Y_1Y_2\right)=2\left(1-p\right). \tag{4}$$
- $$I\left(U_1U_2;Y_1Y_2\right)=I\left(U_1;Y_1Y_2\right)+I\left(U_2;U_1Y_1Y_2\right). \tag{5}$$
- $$I\left(U_1;Y_1Y_2\right)=H\left(U_1\right)-H\left(U_1|Y_1Y_2\right)=1-2p+p^2. \tag{6}$$
- $$I\left(U_2;U_1Y_1Y_2\right)=H\left(U_2\right)-H\left(U_2|U_1Y_1Y_2\right)=1-p^2. \tag{7}$$

Mutual Information Computations (1)

Proof of (4): Recall for BEC(p), $I(X;Y) = 1 - p$.
$I(X_1X_2;Y_1Y_2)$ means we use the same channel twice.
It can be shown that
$I(X_1X_2;Y_1Y_2) = I(X_1;Y_1) + I(X_2;Y_2) = 2(1-p)$.
At the same time, the mapping from U_1U_2 to X_1X_2 is invertible.
There is no information gain or loss.
Hence, $I(U_1U_2;Y_1Y_2) = I(X_1X_2;Y_1Y_2) = 2(1-p)$.

Mutual Information Computations (2)

Proof of (5): It is proved by two steps.

- Chain rule: $I\left(U_1U_2;Y_1Y_2\right)=I\left(U_1;Y_1Y_2\right)+I\left(U_2;Y_1Y_2|U_1\right)$.

$$\begin{aligned} I\left(U_1U_2;Y_1Y_2\right) &= \mathrm{E}\left[\log\frac{p(u_1u_2\boldsymbol{y})}{p(u_1u_2)p(\boldsymbol{y})}\right] \\ &= \mathrm{E}\left[\log\frac{p(u_2\boldsymbol{y}|u_1)p(u_1){\color{blue}p(\boldsymbol{y}|u_1)}}{p(u_2|u_1)p(u_1)p(\boldsymbol{y}){\color{blue}p(\boldsymbol{y}|u_1)}}\right] \\ &= \mathrm{E}\left[\log\frac{p(u_2\boldsymbol{y}|u_1)}{p(u_2|u_1)p(\boldsymbol{y}|u_1)}+\log\frac{p(u_1\boldsymbol{y})}{p(u_1)p(\boldsymbol{y})}\right] \\ &= I\left(U_2;Y_1Y_2|U_1\right)+I\left(U_1;Y_1Y_2\right). \end{aligned}$$

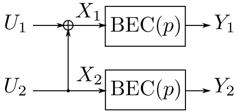
- $I\left(U_2;Y_1Y_2|U_1\right)=I\left(U_2;U_1Y_1Y_2\right)$.

$$\begin{aligned} I\left(U_2;Y_1Y_2|U_1\right) &= H\left(U_2|U_1\right)-H\left(U_2|U_1Y_1Y_2\right) \\ &= H\left(U_2\right)-H\left(U_2|U_1Y_1Y_2\right)=I\left(U_2;U_1Y_1Y_2\right). \end{aligned}$$

Hence

$$2\left(1-p\right)=I\left(U_1U_2;Y_1Y_2\right)=I\left(U_1;Y_1Y_2\right)+I\left(U_2;U_1Y_1Y_2\right).$$

Mutual Information Computation (3)



Computation of $H(U_1|y_1y_2)$ where $y_1y_2 \in \{0, 1, ?\}^2$.

- If $y_1 \neq ?$ and $y_2 \neq ?$, then U_1 can be uniquely identified.
- As long as one of y_1 and y_2 is $?$, then U_1 has equal probability to be 0 or 1.

$$H(U_1|y_1y_2) = \begin{cases} 0 & \text{if } y_1 \neq ? \text{ and } y_2 \neq ?, \\ 1 & \text{otherwise.} \end{cases}$$

Mutual Information Computation (4)

Proof of (6): $H\left(U_1 | Y_1 Y_2\right) = \mathrm{E}_{Y_1 Y_2}\left[H\left(U_1 | y_1 y_2\right)\right]$.

- $p_{Y_1 Y_2}(00) = p_{Y_1 Y_2}(01) = p_{Y_1 Y_2}(10) = p_{Y_1 Y_2}(11) = \frac{1}{4}(1-p)^2$.
- $p_{Y_1 Y_2}(?0) = p_{Y_1 Y_2}(?1) = p\left(\frac{1}{2}(1-p)\right) = \frac{p(1-p)}{2}$.
  $p_{Y_1 Y_2}(0?) = p_{Y_1 Y_2}(1?) = \frac{p(1-p)}{2}$.
- $p_{Y_1 Y_2}(??) = p^2$.

Hence,

$$H\left(U_1 | Y_1 Y_2\right) = 4\left(\frac{p(1-p)}{2} \cdot 1\right) + p^2 \cdot 1 = 2p - p^2.$$

Hence,

$$I\left(U_1 ; Y_1 Y_2\right) = 1 - 2p + p^2.$$

Mutual Information Computation (5)

Proof of (7): $H\left(U_2|Y_1Y_2\right) = \mathrm{E}_{Y_1Y_2}\left[H\left(U_2|u_1y_1y_2\right)\right]$.

$$H\left(U_2|u_1y_1y_2\right) = \begin{cases} 1 & \text{if } y_1 = y_2 = ?, \\ 0 & \text{otherwise.} \end{cases}$$

- If $y_1 = ?$ but $y_2 \neq ?$, then $u_2 = y_2$.
- If $y_1 \neq ?$ but $y_2 = ?$, then $u_2 = u_1 + y_1$.

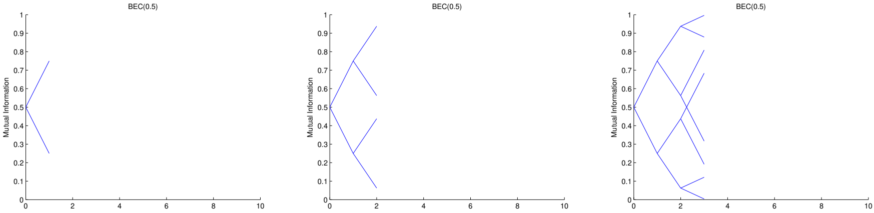
Based on the computations of $p_{Y_1Y_2}\left(y_1y_2\right)$, one has

$$H\left(U_2|U_1Y_1Y_2\right) = p^2 \cdot 1 = p^2.$$

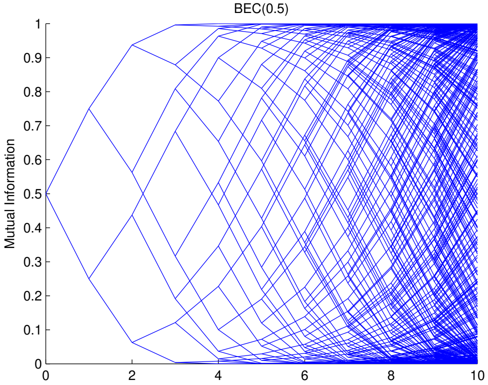
Hence,

$$I\left(U_2;U_1Y_1Y_2\right) = 1 - p^2.$$

Mutual Information: $\mathrm{BEC}\,(0.5)$



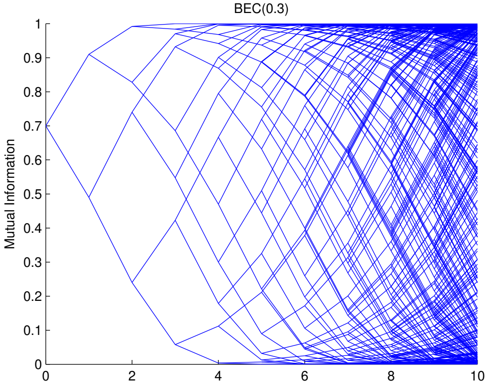
Mutual Information: $\text{BEC}\,(0.5)$



Number of channels with $I > 0.95$: $421\ (41\%)\ (\overset{k\to\infty}{\to} 50\%)$

Number of channels with $I < 0.05$: $421\ (41\%)\ (\overset{k\to\infty}{\to} 50\%)$

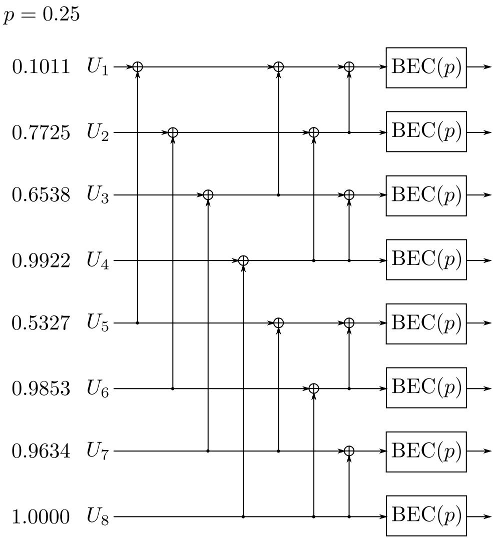
Mutual Information: BEC (0.3)



Number of channels with $I > 0.95$: 633 (62%) $(\overset{k\to\infty}{\to} 70\%)$

Number of channels with $I < 0.05$: 230 (22%) $(\overset{k\to\infty}{\to} 30\%)$

Encoding Example for $BEC\,(0.25)$ (1)



Encoding Example for $BEC\,(0.25)$ (2)

Design a code of which the rate is less than $1 - p = 0.75$.
For example, an $[8, 5]$ code.
From the mutual information calculation, the channels of U_1, U_5 and U_3 are the most unreliable.

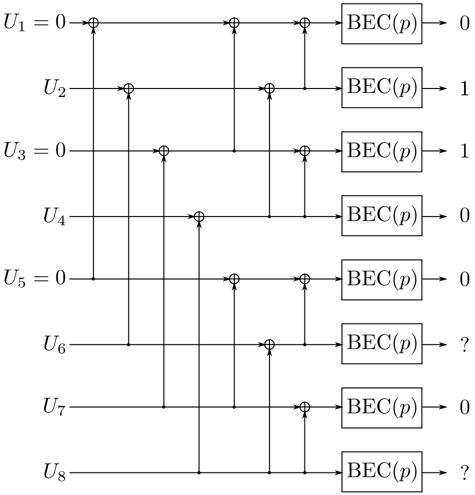
- Set $U_1 = 0$, $U_3 = 0$, and $U_5 = 0$.
- Map the information bits to other bits.
- The generator matrix $\boldsymbol{G}$ of the code can be generated by deleting the 1st, 3rd, and 5th rows of $\boldsymbol{G}_3$.

Encoding Example for $BEC\,(0.3)$ (3)

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

If $\boldsymbol{w} = 11011$, then $\boldsymbol{x} = \boldsymbol{w}\boldsymbol{G} = 01100101$.

Decoding Example (1)



Decoding Example (2)

$$
\begin{array}{cccc}
0 & 0 & 1 & 0 \\
? & 1 & 1 & 1 \\
0 & 1 & 1 & 1 \\
? & 0 & 0 & 0 \\
0 & 0 & ? & 0 \\
? & ? & ? & ? \\
1 & 1 & ? & 0 \\
? & ? & ? & ? \\
 & \multicolumn{2}{c}{\leftarrow} &
\end{array}
\qquad\qquad
\begin{array}{cccc}
0 & 0 & 1 & 0 \\
? & 1 & 1 & 1 \\
0 & 1 & 1 & 1 \\
? & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
? & ? & 1 & 1 \\
1 & 1 & 1 & 0 \\
? & ? & 1 & 1 \\
 & \multicolumn{2}{c}{\rightarrow} &
\end{array}
\qquad\qquad
\begin{array}{cccc}
0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
 & \multicolumn{2}{c}{\leftarrow} &
\end{array}
$$

Hence $\boldsymbol{x} = 01100101$, $\boldsymbol{w} = 11011$. (complexity $O(n \log n)$)

Decoding Details

$$
\begin{matrix} 0 & ? & ? & 0 \\ ? & ? & ? & 1 \\ 0 & ? & ? & 1 \\ ? & ? & ? & 0 \\ 0 & ? & ? & 0 \\ ? & ? & ? & ? \\ ? & ? & ? & 0 \\ ? & ? & ? & ? \end{matrix}
\Rightarrow
\begin{matrix} 0 & ? & 1 & 0 \\ ? & ? & 1 & 1 \\ 0 & ? & 1 & 1 \\ ? & ? & 0 & 0 \\ 0 & ? & ? & 0 \\ ? & ? & ? & ? \\ ? & ? & ? & 0 \\ ? & ? & ? & ? \end{matrix}
\Rightarrow
\begin{matrix} 0 & 0 & 1 & 0 \\ ? & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ ? & 0 & 0 & 0 \\ 0 & ? & ? & 0 \\ ? & ? & ? & ? \\ ? & ? & ? & 0 \\ ? & ? & ? & ? \end{matrix}
\Rightarrow
\begin{matrix} 0 & 0 & 1 & 0 \\ ? & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ ? & 0 & 0 & 0 \\ 0 & 0 & ? & 0 \\ ? & ? & ? & ? \\ 1 & 1 & ? & 0 \\ ? & ? & ? & ? \end{matrix}
$$

Decoding Details

$$\begin{array}{cccc} 0 & 0 & 1 & 0 \\ ? & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ ? & 0 & 0 & 0 \\ 0 & 0 & ? & 0 \\ ? & ? & ? & ? \\ 1 & 1 & ? & 0 \\ ? & ? & ? & ? \end{array} \Rightarrow \begin{array}{cccc} 0 & 0 & 1 & 0 \\ ? & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ ? & 0 & 0 & 0 \\ 0 & {\color{red}0} & {\color{red}1} & 0 \\ ? & ? & ? & ? \\ 1 & {\color{red}1} & {\color{red}1} & 0 \\ ? & ? & ? & ? \end{array} \Rightarrow \begin{array}{cccc} 0 & 0 & 1 & 0 \\ ? & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ ? & 0 & 0 & 0 \\ 0 & 0 & {\color{red}1} & {\color{red}0} \\ ? & ? & {\color{red}1} & {\color{red}1} \\ 1 & 1 & {\color{blue}1} & {\color{blue}0} \\ ? & ? & {\color{blue}1} & {\color{blue}1} \end{array}$$

Decoding Details

$$
\begin{matrix} 0 & 0 & 1 & 0 \\ ? & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ ? & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ ? & ? & 1 & 1 \\ 1 & 1 & 1 & 0 \\ ? & ? & 1 & 1 \end{matrix}
\Rightarrow
\begin{matrix} 0 & 0 & 1 & 0 \\ ? & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ ? & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ ? & {\color{red}0} & {\color{red}1} & 1 \\ 1 & 1 & 1 & 0 \\ ? & {\color{red}1} & {\color{red}1} & 1 \end{matrix}
\Rightarrow
\begin{matrix} 0 & 0 & 1 & 0 \\ {\color{red}1} & {\color{red}1} & 1 & 1 \\ 0 & 1 & 1 & 1 \\ {\color{blue}1} & {\color{blue}0} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ {\color{red}0} & {\color{red}0} & 1 & 1 \\ 1 & 1 & 1 & 0 \\ {\color{blue}1} & {\color{blue}1} & 1 & 1 \end{matrix}
$$