Digital Communications
SSY125, Lecture 8

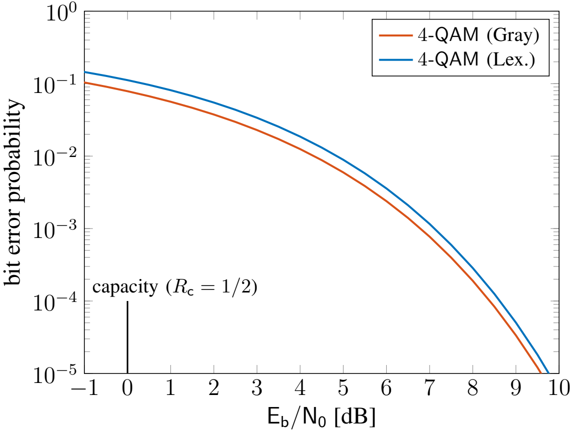
Basics of Error Correcting Coding
(Chapter 7)

Alexandre Graell i Amat
alexandre.graell@chalmers.se
https://sites.google.com/site/agraellamat

November 20, 2019

CHALMERS

Error Correcting Coding



Error Correcting Coding

- Shannon's lesson: To achieve capacity, need of error correcting coding!
- Principle: Introduce redundancy in a controlled manner such that it can be exploited by the receiver to correct errors introduced by the channel.
- Shannon proved the existence of capacity-achieving codes based on random coding arguments (no insight on how to construct practical codes).
- Applications: distributed computing, distributed storage and caching, uncoordinated multiple-access, DNA storage, quantum key distribution, post-quantum cryptography, ...

Error Correcting Coding

Definition (Error correcting code)

A binary block code of code length n and dimension k, $\mathcal{C}(n,k)$, is a collection of 2^k binary tuples of length n bits,

$$\mathcal{C}(n,k) = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_k : \boldsymbol{c}_i \in \{0,1\}^n\},$$

called codewords.

Definition (Encoder)

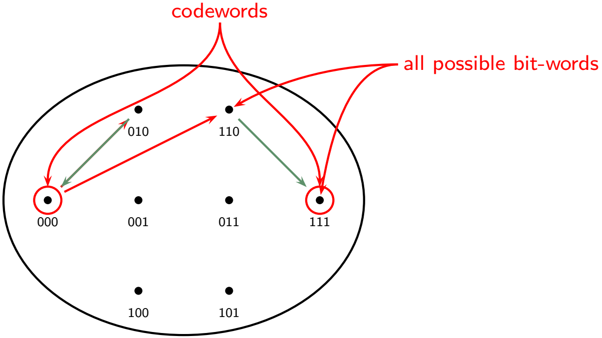
An encoder $\mathcal{E}$ is a set of 2^k pairs $(\boldsymbol{u}, \boldsymbol{c})$, where $\boldsymbol{u}$ is the information word of length k bits and $\boldsymbol{c}$ is the codeword of length n bits. It consists of

(i) 2^k codewords belonging to a set $\mathcal{C} \subset \{0,1\}^n$,

(ii) A mapping function from $\{0,1\}^k$ to $\mathcal{C}$ that maps k information bits $\boldsymbol{u} = (u_1, \ldots, u_k) \in \{0,1\}^k$ into a codeword of n coded bits $\boldsymbol{c} = (c_1, \ldots, c_n) \in \mathcal{C}$.

Error Correcting Coding

Graphical Interpretation

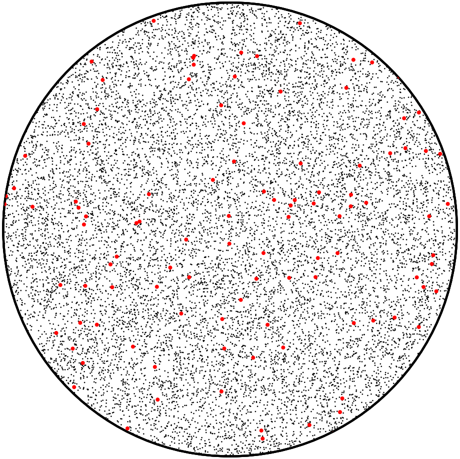


- $\mathcal{C}_{\text{rep}}(n=3, k=1) = \{(0,0,0),(1,1,1)\}$ and encoding

$$u = 0 \quad \rightarrow \quad \boldsymbol{c}_1 = (0,0,0)$$
$$u = 1 \quad \rightarrow \quad \boldsymbol{c}_2 = (1,1,1).$$

Error Correcting Coding

$(14, 7)$ code



Code Rate

- Code rate:

$$R_c \triangleq \frac{k}{n} < 1.$$

  A measure of the redundancy of the code.

- Directly related to the spectral efficiency. For BPSK,

$$R = \frac{k}{n} = R_c \quad \text{[bits per symbol]}$$

  Furthermore,

$$\mathsf{E}_s = \mathsf{E}_b R_c$$
$$\frac{\mathsf{E}_b}{\mathsf{N}_0} = \frac{\mathsf{E}_s}{\mathsf{N}_0} \frac{1}{R_c}.$$

Hamming Weight and Hamming Distance

Definition (Hamming weight)

For a binary vector $\boldsymbol{c} = (c_1, \ldots, c_n)$ of length n, the Hamming weight, denoted by $w_{\mathsf{H}}(\boldsymbol{c})$, is the number of entries in which $c_i = 1$, i.e.,

$$w_{\mathsf{H}}(\boldsymbol{c}) = |\{c_i = 1\}|.$$

Definition (Hamming distance)

For any two binary vectors $\boldsymbol{c}$ and $\tilde{\boldsymbol{c}}$ of length n, the Hamming distance, denoted by $d_{\mathsf{H}}(\boldsymbol{c}, \tilde{\boldsymbol{c}})$ is the number of entries in which $\boldsymbol{c}$ and $\tilde{\boldsymbol{c}}$ differ.

- It follows

$$d_{\mathsf{H}}(\boldsymbol{c}, \tilde{\boldsymbol{c}}) = w_{\mathsf{H}}(\boldsymbol{c} + \tilde{\boldsymbol{c}}).$$

Minimum Hamming Distance

Definition (Minimum Hamming distance)

The minimum Hamming distance of a code $\mathcal{C}$, denoted by $d_{\mathsf{min}}(\mathcal{C})$, is defined as

$$d_{\mathsf{min}}(\mathcal{C}) \triangleq \min_{\substack{\boldsymbol{c},\tilde{\boldsymbol{c}}\in\mathcal{C} \\ \boldsymbol{c}\neq\tilde{\boldsymbol{c}}}} d_{\mathsf{H}}(\boldsymbol{c},\tilde{\boldsymbol{c}}).$$

- Relevant parameter related to the error correction (and detection) capabilities of the code.
- $\mathcal{C}(n,k)$ code with minimum Hamming distance $d_{\mathsf{min}} \longrightarrow \mathcal{C}(n,k,d_{\mathsf{min}})$ code.

Linear Block Codes

Definition (Linear block code)

A binary block code $\mathcal{C}(n,k)$ is linear if and only if its 2^k codewords $\boldsymbol{c}_1,\ldots,\boldsymbol{c}_k$ form a k-dimensional subspace of the n-dimensional vector space $\{0,1\}^n$.

(i) For $\boldsymbol{c}\in\mathcal{C}$ and $\tilde{\boldsymbol{c}}\in\mathcal{C}$, then $\boldsymbol{c}+\tilde{\boldsymbol{c}}\in\mathcal{C}$.

(ii) $(0,\ldots,0)\in\mathcal{C}$.

(iii)

$$d_{\mathsf{min}}(\mathcal{C})=\min_{\substack{\boldsymbol{c}\in\mathcal{C}\\ \boldsymbol{c}\neq\boldsymbol{0}}} w_{\mathsf{H}}(\boldsymbol{c})$$

Theorem (Singleton bound)

The minimum Hamming distance of a linear code $\mathcal{C}(n,k)$ satisfies

$$d_{\mathsf{min}}\leq n-k+1.$$

Optimum Decoding of Linear Block Codes

- BPSK modulation with $\mathsf{E_s} = 1$, i.e., $\mathsf{X}_1 = -1$ and $\mathsf{X}_2 = +1$.
- c_i modulated onto $x_i = (-1)^{c_i}$, i.e., mapping $0 \rightarrow +1$ and $1 \rightarrow -1$.
- Transmission over a memoryless AWGN channel,

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n},$$

  with $N_i \sim \mathcal{N}(0, \sigma^2)$.

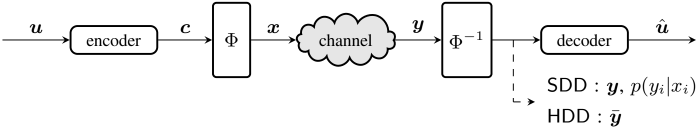
- Optimum decoding rule (equiprobable codewords):

$$\hat{\boldsymbol{x}}_{\mathsf{ML}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{y}|\boldsymbol{x}),$$

- Equivalently,

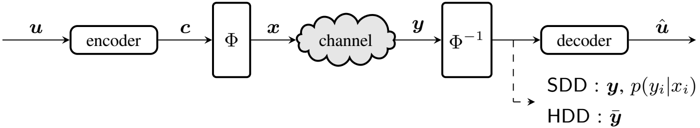
$$\hat{\boldsymbol{c}}_{\mathsf{ML}} = \arg\max_{\boldsymbol{c} \in \mathcal{C}} p(\boldsymbol{y}|\boldsymbol{c}).$$

Optimum Decoding of Linear Block Codes



- Soft-decision decoding: the decoder estimates $\boldsymbol{c}$ based on the full observation $\boldsymbol{y}$ (equivalently, the decoder is fed with the transition probabilities $p(y_i|x_i)$).
- Hard-decision decoding: the demodulator takes hard decisions at the channel output and the sequence of hard-detected symbols, denoted by $\bar{\boldsymbol{y}}$, is fed to the decoder.

AWGN Channel with Hard Decisions



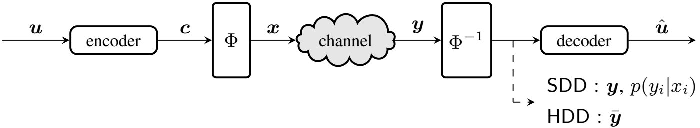
- Each received value y_i is quantized to two levels,

$$\bar{y}_i = \begin{cases} 1 & y_i < 0 \\ 0 & y_i \geq 0. \end{cases}$$

- The equivalent channel between the encoder and the decoder is a discrete memoryless channel (input $\boldsymbol{c}$ and output $\bar{\boldsymbol{y}}$).
- Transition probabilities:

$$\Pr\left(\bar{y}_i = 0 \middle| c_i = 1\right), \qquad \Pr\left(\bar{y}_i = 1 \middle| c_i = 1\right),$$
$$\Pr\left(\bar{y}_i = 1 \middle| c_i = 0\right), \qquad \Pr\left(\bar{y}_i = 0 \middle| c_i = 0\right).$$

AWGN Channel with Hard Decisions

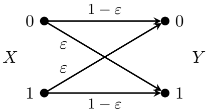


- Transition probabilities:

$$\Pr\left(\bar{y}_i = 0 \middle| c_i = 1\right) = \mathsf{Q}\left(\sqrt{\frac{2R_{\mathsf{c}}\mathsf{E_b}}{\mathsf{N_0}}}\right) \triangleq \varepsilon$$

$$\Pr\left(\bar{y}_i = 1 \middle| c_i = 1\right) = 1 - \mathsf{Q}\left(\sqrt{\frac{2R_{\mathsf{c}}\mathsf{E_b}}{\mathsf{N_0}}}\right).$$

- Due to symmetry, $\Pr\left(\bar{y}_1 = 1 \middle| c_i = 0\right) = \Pr\left(\bar{y}_i = 0 \middle| c_i = 1\right)$ and $\Pr\left(\bar{y}_1 = 0 \middle| c_i = 0\right) = \Pr\left(\bar{y}_i = 1 \middle| c_i = 1\right)$.
- The equivalent channel between $\boldsymbol{c}$ and $\bar{\boldsymbol{y}}$ is the binary symmetric channel!



Hard-Decision Decoding

- Decoder estimates the transmitted codeword based on the binary sequence of quantized values $\bar{\boldsymbol{y}} = (\bar{y}_1, \ldots, \bar{y}_n)$.
- Assuming a memoryless channel,

$$\begin{aligned}\hat{\boldsymbol{c}} &= \arg\max_{\boldsymbol{c}\in\mathcal{C}} p(\bar{\boldsymbol{y}}|\boldsymbol{c}) \\ &= \arg\max_{\boldsymbol{c}\in\mathcal{C}} \prod_{i=1}^{n} p(\bar{y}_i|c_i).\end{aligned}$$

- As we have seen,

$$p(\bar{y}_i|c_i) = \begin{cases} \varepsilon & \bar{y}_i \neq c_i \\ 1-\varepsilon & \bar{y}_i = c_i \end{cases}.$$

Hard-Decision Decoding

$$
\begin{aligned}
\hat{\boldsymbol{c}} &= \arg\max_{\boldsymbol{c}\in\mathcal{C}} p(\bar{\boldsymbol{y}}|\boldsymbol{c}) = \arg\max_{\boldsymbol{c}\in\mathcal{C}} \prod_{i=1}^{n} p(\bar{y}_i|c_i)\\
&= \arg\max_{\boldsymbol{c}\in\mathcal{C}} \varepsilon^{d_{\mathsf{H}}(\boldsymbol{c},\bar{\boldsymbol{y}})}(1-\varepsilon)^{n-d_{\mathsf{H}}(\boldsymbol{c},\bar{\boldsymbol{y}})}\\
&= \arg\max_{\boldsymbol{c}\in\mathcal{C}} \log\left(\varepsilon^{d_{\mathsf{H}}(\boldsymbol{c},\bar{\boldsymbol{y}})}(1-\varepsilon)^{n-d_{\mathsf{H}}(\boldsymbol{c},\bar{\boldsymbol{y}})}\right)\\
&= \arg\max_{\boldsymbol{c}\in\mathcal{C}} d_{\mathsf{H}}(\boldsymbol{c},\bar{\boldsymbol{y}})\log\varepsilon + (n-d_{\mathsf{H}}(\boldsymbol{c},\bar{\boldsymbol{y}}))\log(1-\varepsilon)\\
&= \arg\max_{\boldsymbol{c}\in\mathcal{C}} d_{\mathsf{H}}(\boldsymbol{c},\bar{\boldsymbol{y}})\log\left(\frac{\varepsilon}{1-\varepsilon}\right) + n\log(1-\varepsilon)\\
&= \arg\max_{\boldsymbol{c}\in\mathcal{C}} d_{\mathsf{H}}(\boldsymbol{c},\bar{\boldsymbol{y}})\log\left(\frac{\varepsilon}{1-\varepsilon}\right)\\
&= \arg\min_{\boldsymbol{c}\in\mathcal{C}} d_{\mathsf{H}}(\boldsymbol{c},\bar{\boldsymbol{y}}),
\end{aligned}
$$

where in the last equality we assumed that $\varepsilon < 0.5$.

ML Decoding Rule

Choose among all possible transmitted codewords the codeword $\boldsymbol{c}$ that minimizes the Hamming distance between $\boldsymbol{c}$ and $\bar{\boldsymbol{y}}$.

Soft-Decision Decoding

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}\in\mathcal{C}} p(\boldsymbol{y}|\boldsymbol{c}) = \arg\max_{\boldsymbol{c}\in\mathcal{C}} \prod_{i=1}^{n} p(y_i|c_i)$$

$$= \arg\max_{\boldsymbol{c}\in\mathcal{C}} \ln \prod_{i=1}^{n} p(y_i|c_i) = \arg\max_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{n} \ln p(y_i|x_i)$$

- Using $p_{Y|X}(y|x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\|y-x\|^2}{2\sigma^2}}$,

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{N} \ln p(y_i|x_i) = \arg\max_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{n} \frac{-(y_i - x_i)^2}{2\sigma^2}$$

$$= \arg\min_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{n} (y_i - x_i)^2 = \arg\min_{\boldsymbol{c}\in\mathcal{C}} ||\boldsymbol{y} - \boldsymbol{x}||^2$$

$$= \arg\min_{\boldsymbol{c}\in\mathcal{C}} d_{\mathsf{E}}^2(\boldsymbol{x}, \boldsymbol{y}) = \arg\min_{\boldsymbol{c}\in\mathcal{C}} d_{\mathsf{E}}(\boldsymbol{x}, \boldsymbol{y}).$$

ML Decoding Rule

Choose among all possible transmitted codewords the codeword $\boldsymbol{c}$ that minimizes the Euclidean distance between the modulated sequence $\boldsymbol{x}$ and $\boldsymbol{y}$.

Soft-Decision Decoding

- Alternatively, using $x_i = (-1)^{c_i}$,

$$\begin{aligned}
\hat{\boldsymbol{c}} &= \arg\min_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{n} (y_i - x_i)^2 = \arg\min_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{n} (y_i - (-1)^{c_i})^2 \\
&= \arg\min_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{n} (y_i^2 + 1 - 2y_i(-1)^{c_i}) \\
&= \arg\min_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{n} (-2y_i(-1)^{c_i}) \\
&= \arg\max_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{n} y_i(-1)^{c_i} = \arg\max_{\boldsymbol{c}\in\mathcal{C}} \sum_{i=1}^{n} y_i x_i.
\end{aligned}$$

ML Decoding Rule

Choose among all possible transmitted codewords the codeword $\boldsymbol{c}$ that maximizes the correlation metric between $\boldsymbol{x}$ and $\boldsymbol{y}$.

Soft-Decision Decoding vs. Hard-Decision Decoding

Example: $(3,1)$ Repetition Code

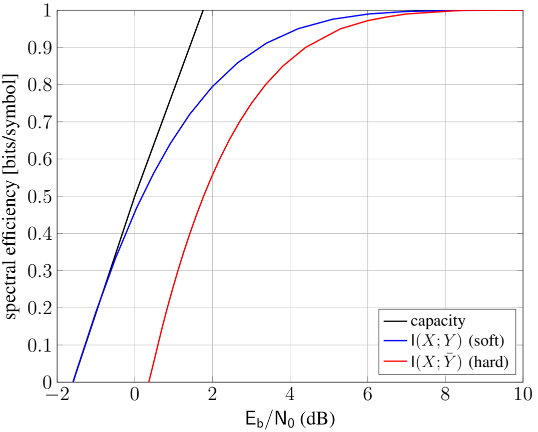
- Transmit $u=0$: $u=0 \longrightarrow \boldsymbol{c}=(0,0,0) \longrightarrow \boldsymbol{x}=(+1,+1,+1)$.
- We receive $\boldsymbol{y}=(-0.2,+1.1,-0.7)$ ($\bar{\boldsymbol{y}}=(1,0,1)$).
- Hard-decision decoding decides for: $\hat{\boldsymbol{c}}=(1,1,1)$ hence $\hat{u}=1$.
- Soft-decision decoding
  - Correlation metric:

$$(0,0,0): \qquad \sum_{i=1}^{3} y_i(-1)^0 = +0.2$$

$$(1,1,1): \qquad \sum_{i=1}^{3} y_i(-1)^1 = -0.2$$

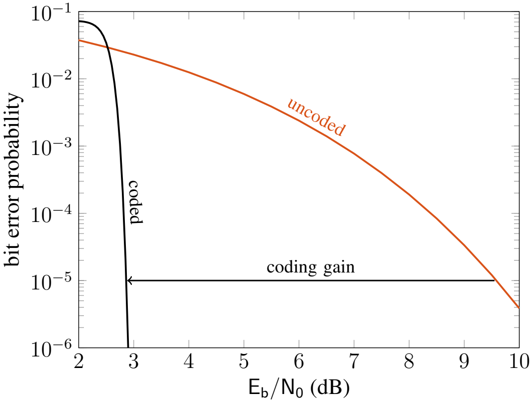
  - Decides for $\hat{\boldsymbol{c}}=(0,0,0)$ and hence $\hat{u}=0$!

Soft-Decision Decoding vs. Hard-Decision Decoding



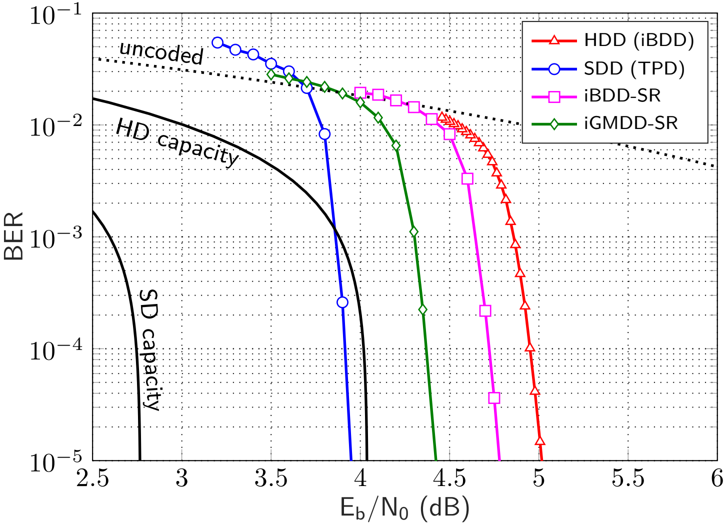
- BPSK transmission, AWGN channel.
- Hard-decision decoding results in a loss of 1–2 dB.

The Advantage of Coding



- Coding gain: the difference (in decibels) in the required E_b/N_0 to achieve a given probability of error.

Soft-Decision Decoding vs. Hard-Decision Decoding



- AWGN channel, $R_c = 0.87$ product code.