Introduction to Information Theory

Lecture 12 Part B

Lecturer: Haim Permuter *Scribe: Eyal Yakir*

I. BackGround

A. Section 1: Introduction to Error-Correcting Codes

In our previous lectures, we learned about codes that help compress data. Now, we're going to discuss another type of code called error-correcting codes. These codes are really important because they help ensure reliable communication, especially when we're dealing with channels that have a lot of noise or interference. When we send data through a communication system, it can get mixed up and corrupted due to noise and interference, which can cause errors in what we receive. Error-correcting codes are like detectives that can detect and fix these errors, making sure our data gets transmitted accurately and securely.

B. Section 2: Introduction to Error-Correcting Codes

To comprehend error-correcting codes, it is essential to understand the notion of the channel rate. The channel rate signifies the efficiency of transmitting information over the channel and can be calculated as the ratio of the number of information bits transmitted, denoted as k, to the total number of bits transmitted, denoted as n. This includes both the information bits and any redundant bits introduced by the code. In the case of a specific code represented as (n, k), where n denotes the message size (in bits) and k denotes the data in the message (in bits), the rate R can be computed as:

$$R = \frac{k}{n} \tag{1}$$

The rate of a channel provides insights into how efficiently we can transmit the desired information, taking into account the presence of redundant bits for error correction.

Example 1 Consider the n-repetition code, where a message abc is encoded as $\underbrace{aa\ldots a}_{\text{n times}}\underbrace{bb\ldots b}_{\text{n times}}\underbrace{cc\ldots c}_{\text{n times}}$. We will analyze the Binary Eraser Channel, characterized by a probability of $1-p$ for correct transmission and p for deletion. The channel's conditional probabilities are denoted as $p(y=i|x=i)=1-p$ and $p(y=?|x=i)=p$.

The Binary Eraser Channel transmits each symbol with a probability of $1-p$ for correct transmission and p for deletion. Symbol i is received correctly with probability $1-p$, while an erasure symbol is received with probability p.

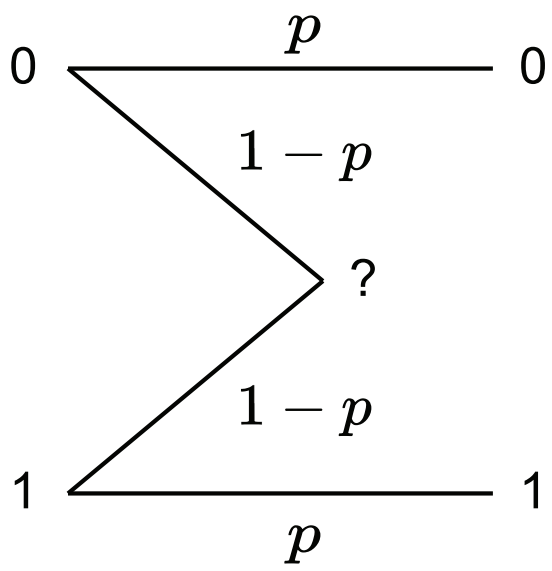

Fig. 1: channel probabilities diagram

Next, we will delve into the examination of the error probability of the code and the rate for varying values of n. For a given n, the probability of encountering an error is the probability of deletion of n bits. $p(\text{Fail})=p^n$. As we transmit n bits for every bit of data, the code corresponds to a $(n,1)$ code, resulting in a rate of $R=\frac{1}{n}$.

C. Capacity of a Channel

The capacity of a channel represents the maximum achievable rate for transmitting information without errors. It serves as a fundamental limit, determined by factors such as noise, interference, and bandwidth restrictions. Typically measured in bits per channel

use, a high-capacity channel is considered "good," while a low-capacity channel is deemed "bad". Mathematically, the capacity C of a channel is obtained by maximizing the mutual information $I(Y;X)$ between the channel's input X and output Y, averaged over all possible input distributions $p(X)$:

Theorem 1 (Shannon's Theorem for Channel Capacity) The capacity of a channel, denoted as C, is given by:

$$C = \max_{p(X)} I(Y;X) \tag{2}$$

In this equation, $I(Y;X)$ represents the mutual information between the channel's input and output. It quantifies the amount of information that can be reliably transmitted through the channel.

Example 2 we will find the capacity of the Binary Eraser Channel described in example 1 (without repetition code).

$$\begin{aligned} I(Y;X) \ &= H(X) - H(X|Y) \\ \overset{(a)}{=} H(X) - H(X|Y=0)\cdot P(Y=0) - H(X|Y=1)\cdot P(Y=1) - H(X|Y=?)\cdot P(Y=?) \\ \overset{(b)}{=} H(X) - pH(X) = (1-p)H(X) \end{aligned}$$

where a is from Law of total probability and (b) is correct becuase

$$p(x|y=?) = \frac{p(x, y=?)}{p(y=?)} = \frac{p(y=?|x)\cdot p(x)}{p(y=?)} = \frac{p\cdot p(x)}{p} = p(x)$$

Furthermore, as both $H(X|Y=0)$ and $H(X|Y=1)$ exhibit deterministic behavior (when we receive 1, we can ascertain that 1 was sent; when we send 0, we can confidently confirm that 0 was sent), there is no uncertainty involved, resulting in $H(X|Y=0) = H(X|Y=1) = 0$.

Using Shannon's theorem, we can determine the maximum capacity (C) of a communication channel by maximizing the mutual information (I) between the input variable (X) and the output variable (Y), denoted as $I(Y;X)$. Mathematically, this can be expressed as:

$$C = \max_{p(X)} I(Y;X) = \max_{p(X)} (1-p)H(X)$$

where $p(X)$ represents the probability distribution of the input variable. In the case of a Bernoulli distribution with a probability of success of 0.5, the maximum capacity can be simplified to:

$$C = 1 - p$$

II. Polar Codes

A. Concept of Polar Codes

The concept behind polar codes is based on the idea of polarization, where a channel is transformed into subchannels with significantly different error probabilities. Polar codes leverage this polarization phenomenon to efficiently and reliably transmit information over a noisy channel. The key concept in polar codes is to identify the reliable subchannels and exploit them for transmission while effectively ignoring the unreliable subchannels. By selectively utilizing the reliable subchannels, polar codes achieve near-optimal performance in terms of error correction capability and channel capacity.

B. Encoder

1) Motivation for Encoding: In the field of digital communication systems, one of the key objectives is to ensure the reliable transmission of data over error-prone channels. To address this challenge, error-correcting codes are often employed. These codes introduce additional redundant bits to the original digital signal, enabling the detection and correction of errors during the transmission process.

The main motivation for incorporating encoding techniques into digital communication systems is twofold. Firstly, the encoding process aims to add a minimal number of spare bits to a digital signal of length K, in order to enhance its error-correcting capabilities. This results in the ability to effectively detect and correct errors that may occur during transmission. Secondly, the encoding and subsequent decoding process should be efficient and accurate, ensuring that the original information can be correctly retrieved from the received encoded signal.

By achieving these goals, the integration of error-correcting codes not only minimizes the occurrence of failed transmissions but also optimizes channel usage. This, in turn, leads to a reduction in overall channel costs, making encoding an essential aspect of digital communication systems.

2) Process and Theory of Encoding: In the context of information theory, encoders are generally perceived as transformations that map k bits into N bits. In the scenario of polar codes, the objective is to segment the channel – that is, the transformation – into multiple sub-channels, each possessing distinct properties. Some sub-channels may be noise-ridden (or "bad"), while others might be noise-free (or "good"). In this section, we first present a proof demonstrating that the main channel can indeed be subdivided, followed by an outline of the encoding process.

Consider an encoder-decoder system where an extension of k bits results in n input bits $(v_1, v_2, \ldots, v_n)$ and n output bits $(y_1, y_2, \ldots, y_n)$. In this case, the input bits are statistically independent, denoted as $v_i \perp v_j, \forall i \neq j$.

Theorem 2 (Segmentation of Channel Self-Entropy into Sub-channels)

$$\begin{aligned} I(v_1, v_2, \ldots, v_n; y_1, y_2, \ldots, y_n) =& I(v_1; y_1, y_2, \ldots, y_n) \\ &+ I(v_2; y_1, y_2, \ldots, y_n, v_1) \\ &+ \ldots + I(v_n; y_1, y_2, \ldots, y_n, v_1, v_2, \ldots, v_{n-1}) \end{aligned}$$

Proof: The theorem employs the definition of mutual information in conjunction with the chain rule.

Commencing with the left-hand side of the equation, we find:

$$\begin{aligned} I(v_1, v_2, \ldots, v_n; y_1, y_2, \ldots, y_n) &= \sum_{i=1}^{n} I(v_i; y_1, y_2, \ldots, y_n, v_1, v_2, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n) \\ &= I(v_1; y_1, y_2, \ldots, y_n) + I(v_2; y_1, y_2, \ldots, y_n, v_1) + \ldots \\ &\quad + I(v_n; y_1, y_2, \ldots, y_n, v_1, v_2, \ldots, v_{n-1}) \end{aligned}$$

The derivation uses the definition of mutual information, where each term represents the mutual information between the corresponding input and output variables. This effectively validates the theorem. ■

In light of our objective, let us recall that we aim to transform a k-bit input into an n-bit output where $k < n$. Given our proven assertion that a channel with n independent inputs and n outputs is equivalent to n channels each with one input and n outputs sharing information, we intend to capitalize on this characteristic. If we designate the i-th channel as c_i, we can determine the following:

$$\begin{aligned}
\text{Self-Entropy}(c_1) &= I(u_1; y_1, y_2, \ldots, y_n) \\
\text{Self-Entropy}(c_2) &= I(u_2; y_1, y_2, \ldots, y_n) \\
\text{Self-Entropy}(c_3) &= I(u_3; y_1, y_2, \ldots, y_n, u_1, u_2) \\
&\vdots \\
\text{Self-Entropy}(c_n) &= I(u_n; y_1, y_2, \ldots, y_n, u_1, u_2, \ldots, u_{n-1})
\end{aligned}$$

Subsequently, we classify the channels $c_1, c_2, \ldots, c_n$ in order from the most noisy to the clearest. The exact methodology for this classification is beyond the scope of this text. When transmitting k bits of information using n bits of transmitted data (where $k < n$), our approach is to dispatch the data on the top k channels and assign a value of 0 to the remaining $n - k$ less favourable channels. The latter category is commonly referred to as "frozen bits". The Polar Code encoding algorithm involves several steps:

1) **Initialization:** Start by constructing a $2^n \times 2^n$ polarizing matrix, G_n, using the Kronecker power of the basic polarizing matrix, $G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. For example, if $n = 3$, the polarizing matrix G_3 will be an 8×8 matrix.
2) **Channel Selection:** Once the polarizing matrix is formed, we proceed to select the best k channels. The channels are ranked based on their reliability. The best k channels are considered 'good' and the rest are 'bad'.

3) **Frozen Bits:** The bits transmitted through 'bad' channels are called 'frozen bits', and these bits are typically set to zero.
4) **Data Encoding:** We take the k data bits and the $n-k$ frozen bits and encode them by multiplying with the G_n matrix.

Example 3 Let us illustrate the polar code encoding process using an example where $n = 8$ and $k = 4$. In this case, our system employs four information bits and four frozen bits.

For a Polar code with $n = 8$, we need to construct the generator matrix G_8. This is achieved by performing the Kronecker product on the base matrix $G = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ three times, because $n = 2^3$.

$$G_8 = G_2^{\otimes 3} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3}$$

Given the ranking of the channels, the indices of the frozen channels are $1, 2, 3, 5$. Suppose the input bit sequence is $u = [f, f, f, d_1, f, d_2, d_3, d_4]$, where f represents a frozen bit (which is set to 0), and d denotes an information bit.

The encoding process comprises the multiplication of the information bit sequence u with the generator matrix G_8 to yield the codeword $x = uG_8$. Note that the calculation is carried out under a binary field, implying that the addition operation is synonymous with the XOR operation and multiplication corresponds to the AND operation.

$$x = uG_8 = \begin{bmatrix} f & f & f & d_1 & f & d_2 & d_3 & d_4 \end{bmatrix} \times G_8$$

$$= \begin{bmatrix} d_1 + d_2 + d_3 + d_4 & d_2 + d_4 & d_1 + d_3 + d_4 & & \\ & & d_4 & d_2 + d_3 + d_4 & d_2 + d_4 & d_3 + d_4 & d_4 \end{bmatrix}$$

After obtaining the codeword x, it is transmitted over the communication channel.

Decoding in Polar Codes

3) Motivation for Decoding: Decoding plays a crucial role in extracting the original transmitted message from the received codeword in polar codes. The decoding process aims to recover the message accurately, even in the presence of noise and channel errors. There are various decoding algorithms used in polar codes, with the successive cancellation (SC) algorithm being the most commonly employed.

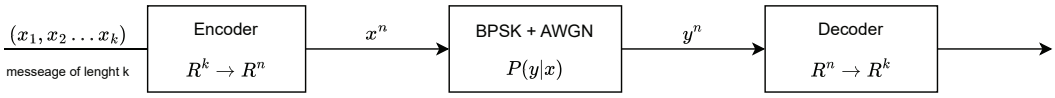

Fig. 2: overview of channel

4) Channel Model Overview: The channel model overview provides a comprehensive understanding of the intricacies involved in a polar code system. In this system, a data input with a magnitude denoted as K undergoes a meticulous encoding process, resulting in an N-dimensional codeword, as explained in the preceding section. Subsequently, the codeword is transmitted through a channel from the transmitter to the receiver. However, during transmission, the codeword encounters inevitable noise that affects its integrity. Our primary objective is to accurately recover the original data by effectively mitigating the impact of transmission noise through the decoding process.

For our analysis, we will focus on the BPSK+AWGN (Binary Phase Shift Keying + Additive White Gaussian Noise) transmission method, although the principles discussed are applicable to other transmission methods as well. In the BPSK mapping scheme, a binary digit '1' is mapped to an analog voltage of $+1$, while a binary digit '0' is mapped to an analog voltage of -1. The noise in the system is mathematically represented as $n \sim \mathcal{N}(0, \sigma^2)$, where n signifies the noise that follows a Gaussian distribution with a mean of 0 and a variance denoted as σ^2.

5) MAP/LLR Decoding: When analyzing the transmission across a channel affected by noise, it becomes necessary to ascertain the original transmitted bit (0/1) from the received analog signal. The decoding procedure involves transforming the received signal into its digital representation through the utilization of the Log-Likelihood Ratio (LLR) concept. The LLR is determined by taking the logarithm of the ratio between the probabilities of receiving a 1 or a 0, given the received bit (r). Mathematically, it can be represented as follows:

$$LLR = \log\left(\frac{P(m=1|r)}{P(m=0|r)}\right) \tag{4}$$

Here, r represents the received bit, and m denotes the corresponding sent bit. Upon examining the LLR, if its value exceeds 0, it indicates that the sent bit was likely 1. Conversely, if the LLR is less than 0, it suggests that the sent bit was likely 0. Therefore, the decoding process involves making a determination based on the sign of the LLR. for a Gaussian distribution noise $n \sim \mathcal{N}(0, \sigma^2)$ we get

$$\frac{P(m=1|r)}{P(m=0|r)} \overset{(base)}{=} \frac{\frac{P(r|m=1)\cdot P(m=1)}{P(r)}}{\frac{P(r|m=0)\cdot P(m=0)}{P(r)}} = \frac{P(r|m=1)}{P(r|m=0)} \overset{(Gaussian)}{=} \frac{\frac{1}{\sqrt{2\pi\sigma^2}}\cdot e^{-\frac{(r-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}}\cdot e^{-\frac{(r+1)^2}{2\sigma^2}}} = e^{\frac{2r}{\sigma^2}} \tag{5}$$

$$LLR = \log\left(\frac{P(m=1|r)}{P(m=0|r)}\right) = \log e^{\frac{2r}{\sigma^2}} = r\cdot\frac{2}{\sigma^2} \tag{6}$$

By extension, for n inputs, a similar outcome will be obtained.

$$\frac{P(m=1|r_1, r_2...r_n)}{P(m=0|r_1, r_2...r_n)} = e^{(r_1+r_2...r_n)\cdot\frac{2}{\sigma^2}} \tag{7}$$

$$LLR = \log\frac{P(m=1|r_1, r_2...r_n)}{P(m=0|r_1, r_2...r_n)} = (r_1 + r_2...r_n)\cdot\frac{2}{\sigma^2} \tag{8}$$

6) Decoding Algorithm: 2 presented the system as a single channel that converts K bits of data into N-bit codewords. However, an alternative perspective is to view it as N channels that transforms 1 bit into 1 bit, as shown in 3. In this case, N-K channels remain fixed/frozen and do not carry any data. By drawing upon a well-established proof,

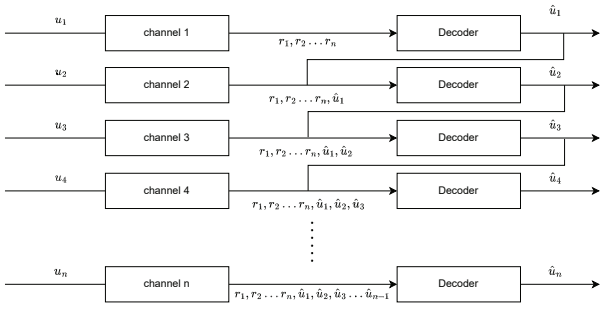

Fig. 3: Diagram of sub-channels

it becomes possible to perceive the channel as a collection of multiple sub-channels, each possessing its own unique self-entropy.

If we let u_i represent the i-th bit of the transmitted message and y_i denote the i-th bit received from the channel.

The algorithm operates as follows:

(Start)
- Upon reaching a new node, provided with the current belief $(r_1, r_2, \dots, r_n)$, the values are redefined as $r_1 = (r_1, r_2, \dots, r_{\frac{n}{2}})$ and $r_2 = (r_{\frac{n}{2}+1}, \dots, r_n)$.
- If the left child exists, traverse to that node while calculating $\hat{u}_{\text{left}} = f(r_1, r_2)$, and proceed to (Start).
- If the left child does not exist but the right child exists, traverse to the right child while calculating $\hat{u}_{\text{right}} = g(r_1, r_2, \hat{u}_{\text{left}})$, and proceed to (Start).
- Upon receiving $\hat{u}_{\text{right}}$ and $\hat{u}_{\text{left}}$ from the child nodes, update the belief to $\hat{r} = (\hat{u}_{\text{right}} \oplus \hat{u}_{\text{left}}, \hat{u}_{\text{right}})$ and propagate upward.

where

$$\begin{cases} \text{sign}(LLR(r_1)) \cdot \text{sign}(LLR(r_2)) \cdot \min(|LLR(r_1)|, |LLR(r_2)|) & \text{if not a leaf} \\ 0 & \text{if leaf and frozen} \\ 0 & \text{if leaf and } LLR(u_i) > 0 \\ 1 & \text{if leaf and } LLR(u_i) < 0 \end{cases} \tag{9}$$

$$g(r_1, r_2, b) = r_2 + (1 - 2b) \cdot r_1 \tag{10}$$

and for 2 vectors $\overline{r_1}$, $\overline{r_2}$ where $\overline{r_1} = (r_1^1, r_2^1 \dots r_n^1)$,$\overline{r_2} = (r_1^2, r_2^2 \dots r_n^2)$

$$f(r_1, r_2) = (f(r_1^1, r_1^2), f(r_2^1, r_2^2) \dots f(r_n^1, r_n^2)) \tag{11}$$

$$g(r_1, r_2, b) = (g(r_1^1, r_1^2, b), g(r_2^1, r_2^2, b) \dots g(r_n^1, r_n^2, b)) \tag{12}$$

Once the root node receives the approximated analog belief $\hat{u} = (\hat{u}_1, \hat{u}_2, \dots, \hat{u}_n)$, it is converted into digital values (0s and 1s) using the LLR/MAP rule as previously discussed. If $\hat{u}_i > 0$, it is changed to 1, and if $\hat{u}_i < 0$, it is changed to 0.

Example 4 In the context of a (4,2) polar code, we are provided with the channel ranks as 1, 2, 3, and 4. Consequently, we can deduce that bits u_1 and u_2 are frozen, while bits u_3 and u_4 contain the transmitted data. Given an analog input signal (r_1, r_2, r_3, r_4), the decoding process follows the steps illustrated in 4. After the algorithm finishes, it

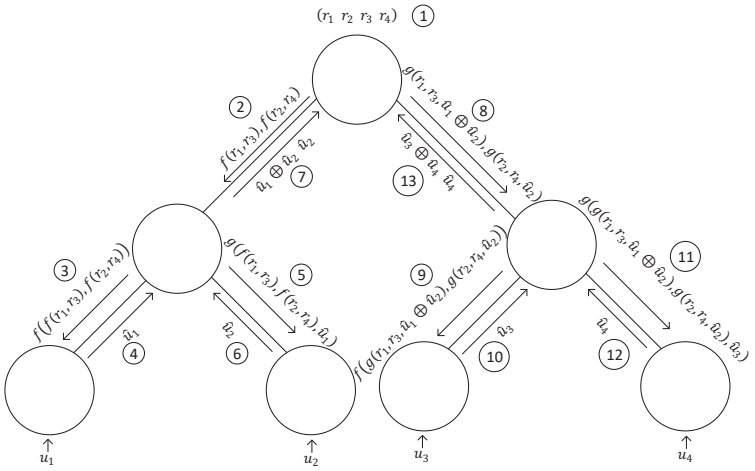

Fig. 4: tree diagram

is necessary to determine the most likely sent digital signal based on the given analog signal. To accomplish this, we employ the Log-Likelihood Ratio (LLR) / Maximum A Posteriori (MAP) rule. Considering the belief at the given tree node $(\hat{u_1}, \hat{u_2}, \hat{u_3}, \hat{u_4})$, we can conclude that the sent digital signal was $(\text{MAP}(\hat{u_1}), \text{MAP}(\hat{u_2}), \text{MAP}(\hat{u_3}), \text{MAP}(\hat{u_4}))$.

The MAP function is determined as follows:

$$\text{MAP}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases}$$

III. References

1) NPTEL-NOC IITM youtube channel.