Chapter 9

Information Theory and Coding

Introduction

- Information theory applies laws of probability and math to study manipulation and collection of information.
- In communication, it deals with mathematical modeling and analysis of a communication source and communication channel.
- In particular, it provides answers to two fundamental questions:
  - The minimum number of bits per symbol required to fully represent the source.
  - What is the ultimate transmission rate for reliable communication over a noisy channel?

Information Sources

- Information sources is an object that produces an event, the outcome of which is random and in accordance with some probability distribution.
- Sources of Information are:
  - **Analog**: source that has continuous set of amplitudes
  - **Discrete**: source that has only finite set of symbols as possible outputs. The set of possible symbols is called *source alphabet*, and elements of set are called *symbols (Q-levels)*.
  - **Source with Memory**: is one for which a current symbol depends on the previous symbols
  - **Memoryless source**: is one for which a current symbol is independent of the previous symbols (statistically independent)

Amount of Information

- The amount of information contained in an event is closely related to its uncertainty.
  - Events with high probability of occurrence convey relatively little information.
  - Event with probability of *1* conveys zero information
  - Information should be proportional to the uncertainty of an outcome
- Consider a Discrete Memoryless Source (DMS) output symbols randomly from finite alphabet $\{s_0, s_1, s_2, \ldots, s_{K-1}\}$ with probabilities P_k where $k = 1, 2, \ldots K-1$.

  Of course $P_0 + P_1 + P_2 + \ldots + P_{K-1} = 1$

Amount of Information

- How much information is produced by this source?
  - Amount of information is closely related to uncertainty or surprise
- Amount of information with event s_k is defined as

$$I(s_k) = \log_2\left(1/P_k\right) = -\log_2\left(P_k\right)$$

- Unit of information is called *bit*. One *bit* is amount of information, we gain when one of two possible and equally likely events occurs.
- Above definition leads to the following properties:
  - $I(s_k) = 0$ *for* $P_k = 1$
  - $I(s_k) \geq 0$ *for* $0 \leq P_k \leq 1$
  - $I(s_k) > I(s_i)$ *for* $P_k < P_i$
  - $I(s_k s_i) = I(s_k) + I(s_i)$ if s_k and s_i are statistically independent.

Amount of Information

Example: A source emits one of four possible symbols during each signaling interval. These symbols occur with the probabilities: P_o=0.4, P_1=0.3, P_2=0.2, and P_3=0.1. Find the amount of information gained by observing the source emitting each of these symbols.

Solution: Hence, $I(s_k) = log_2 (1/p_k)$ bits, then

$I(s_0) = \log_2 (1/0.4) = 1.322$ bits
$I(s_1) = \log_2 (1/0.3) = 1.737$ bits
$I(s_2) = \log_2 (1/0.2) = 2.322$ bits
$I(s_3) = \log_2 (1/0.1) = 3.322$ bits

Entropy

- Assume that we have a source which emits one of k possible symbols in each signaling interval and these symbols are statistically independent.
- $I(s_k)$ represents amount of information produced by the source when it emits symbol s_k during arbitrary signaling interval.
- $I(s_k)$ is a random variable takes on the values $I(s_0)$, $I(s_1)$, ... $I(s_{k-1})$ with probabilities P_0, P_1, ... P_{k-1}.
- The mean value of $I(s_k)$ is called *Entropy* of *DMS*.

Entropy

- The Mean value of $I(s_k)$ over source alphabet $\{s_0, s_1, s_2, \ldots s_k\}$ is given by

$$H = E[I(s_k)]$$
$$= \sum P_k I(s_k)$$
$$= \sum P_k \log_2 \left(1/P_k\right)$$

- Entropy is a measure of average information content per source symbol.

Entropy

Example: Consider a discrete memoryless source with source alphabet $\{s_0, s_1, s_2\}$ with probabilities $P_0=1/4$, $P_1=1/4$ and $P_2=1/2$. Find the entropy of the source.

Solution: The entropy of the given source is

$$H = P_0 \log_2 (1/P_0) + P_1 \log_2 (1/P_1) + P_2 \log_2 (1/P_2)$$
$$= \tfrac{1}{4} \log_2 (4) + \tfrac{1}{4} \log_2 (4) + \tfrac{1}{2} \log_2 (2)$$
$$= 2/4 + 2/4 + \tfrac{1}{2}$$
$$= 3/2 \text{ bits per symbol}$$

Properties of Entropy

- For a discrete memoryless source with a fixed alphabet:
  - $\mathcal{H} = 0$, if and only if the probability $P_k = 1$ for some k, and the remaining probabilities in the set are all zero. This lower bound on the entropy corresponds to 'no uncertainty'.
  - $\mathcal{H} = \log_2(K)$, if and only if $P_k = 1/K$ for all k. This upper bound on the entropy corresponds to 'maximum uncertainty'.
- We therefore have $0 \leq H \leq \log_2(K)$

Entropy

- Now, let us examine $\mathcal{H}$ under different cases for a binary source:
  - Case I: $P_1 = 0.01, P_2 = 0.99 \Rightarrow \mathcal{H} = 0.08$ bits/symbol
  - Case II: $P_1 = 0.40, P_2 = 0.6 \Rightarrow \mathcal{H} = 0.97$ bits/symbol
  - Case III: $P_1 = 0.5, P_2 = 0.5 \Rightarrow \mathcal{H} = 1.00$ bits/symbol

Information Rate

- If information source generates messages at the rate of *r* symbols per second, the *information rate* (*R*) is the average number of bits of information per seconds.
- $\mathcal{R} = r\,\mathcal{H}$ (bits/sec), where $\mathcal{H}$ is the entropy of the source.
- If symbol rate for two sources of the same Entropy are r_1 and r_2,. Then,

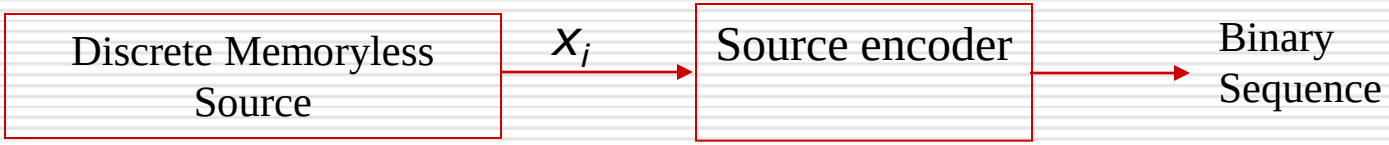
$$R_1 = r_1 H$$

$$R_2 = r_2 H$$

$$\text{If} \quad r_1 > r_2 \quad \text{means} \, R_1 > R_2$$

Source Coding

- Conversion of output of a DMS into a sequence of binary symbols (binary code word) is called source coding.
- Device that performs this conversion is called source encoder.



$X = \{x_1, x_2, \ldots x_m\}$

- Objective of source coding is to minimize the required average bit rate for representation of source.

Source Coding

Code Length and Code Efficiency

- Let X be DMS with entropy $H(X)$ and alphabet $\{x_1, x_2, ..., x_m\}$ with probabilities of occurrence $P(x_i)(i = 1,..., m)$.
- Let the binary code word assigned to symbol x_i by encoder have length n_i bits.
- The length of code word is the number of binary digits in the code word.
- The average code word length L per source symbol is given by

$$L = \sum_{i=1}^{m} p(x_i) n_i$$

Source Coding

Code Length and Code Efficiency

- The code efficiency η is defined as

$$\eta = \frac{L_{\min}}{L}$$

where L_{min} is the minimum possible value of L given as $L_{min} = \mathcal{H}(X)$.

- When η approaches unity, the code is said to be efficient.
- Code redundancy is defined as

$$\gamma = 1 - \eta$$

Source Coding

Source coding theorem

- **Source coding theorem**: for a $\mathcal{DMS}$ X with entropy $\mathcal{H}(x)$, the average code word length L per symbol is bounded as

$$L \geq H(X)$$

- Thus the code efficiency can be rewritten as

$$\eta = \frac{H(X)}{L}$$

Source Coding

Classification of Codes

- Classification of code is best illustrated by an example

Table 1: Binary Codes

x_i	Code 1	Code 2	Code 3	Code 4	Code 5	Code 6
x_1	00	00	0	0	0	1
x_2	01	01	1	10	01	01
x_3	00	10	00	110	011	001
x_4	11	11	11	111	0111	0001

Source Coding

Classification of Codes

1. Fixed-Length Codes: is one whose code word length is fixed. *Codes 1 and 2 in Table 1*
2. Variable-Length Codes: Is one whose word length is not fixed. *All codes in Table 1 except codes 1 and 2.*
3. Distinct Codes: is one whose each code word is distinguishable from other. *All codes in Table 1 except code 1.*
4. Prefix-Free Code: is one in which no code word is the prefix of any other cod word. *Codes 2, 4, 6 of Table 1 are Prefix-free codes*

Source Coding

Classification of Codes

- To illustrate the meaning of prefix code consider the three source codes in Table-2

Table 2: Illustrating Prefix Code

Symbol	Code 1	Code 2	Code 3
s_0	0	0	0
s_1	1	10	01
s_2	00	110	011
s_3	11	111	0111

Source Coding

Classification of Codes

- Uniquely Decodable Codes: is one in which the original source sequence can be reconstructed perfectly for encoded binary sequence.
  - Code 3 of Table 1 is not a uniquely decodable code. Because, binary sequence 1001 may correspond to the source sequences: $x_2\ x_3\ x_2$ *or* $x_2\ x_1\ x_1\ x_2$.
- A sufficient condition for uniquely decodable code is that no code word is a prefix of another.

Shanon-Fano Encoding

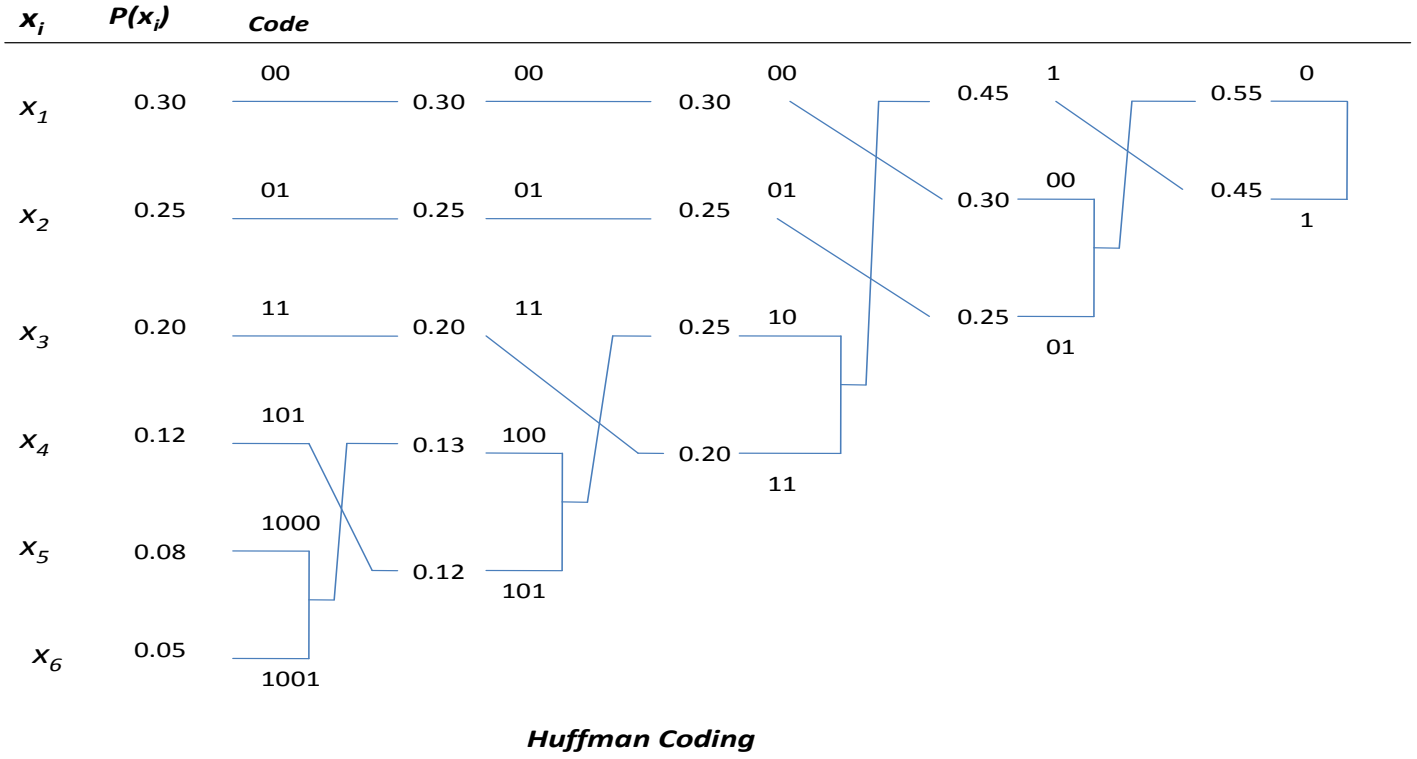
x_i	$P(x_i)$	Step 1	Step 2	Step 3	Step 4	Code 5
x_1	0.30	0	0			00
x_2	0.25	0	1			01
x_3	0.20	1	0			10
x_4	0.12	1	1	0		110
x_5	0.08	1	1	1	0	1110
x_6	0.05	1	1	1	1	1110

$H(X) = 2.36$ bits/symbol

$L = 2.38$ bits/symbol

$\eta = H(X)/L = 0.99$

Huffman Coding



Huffman Coding

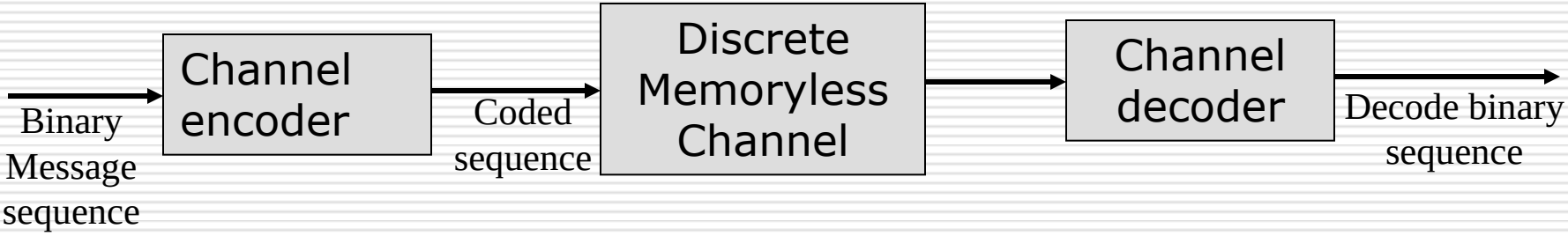
$H(X)$ = 2.36 bits/symbol

L = 2.38 bits/symbol

η = $H(X)/L$ = 0.99

Channel Coding

- A basic diagram for the channel coding is shown below



- The binary message at the input may be output of source encoder.
- Channel encoder introduces redundancy into the data stream by adding bits to message bits in such a way to facilitate detection and correction of bit errors at the receiver.

Channel Coding

- The channel decoder in the receiver exploits the redundancy to determine which bits are actually transmitted.
- The combined objective of the channel encoder and decoder is to minimize the effect of channel noise.

Channel Coding

Block codes

- Block code is a code having all its words of the same length.
- In block codes, the *binary message* or *data sequence* is divided into sequential blocks each of k bits long and each *k-bit* block is encoded into an *n-bit* block, where $n > k$.
- The resultant *block code* is called an *(n,k)* block code.
- Number of redundant bits added by encoder to each data block is $n - k$.

Channel Coding

Linear Block Codes

- **Linear Codes**: Let **a** = $(a_1, a_2, \ldots, a_n)$, and **b** = $(b_1, b_2, \ldots, b_n)$ be two code words in a code C. The sum of **a** and **b** denoted by

$$\mathbf{a} \oplus \mathbf{b} = (a_1 \oplus b_1, a_2 \oplus b_2, \cdots a_n \oplus b_n)$$

$$\textit{where}\ \ 0 \oplus 0 = 0 \quad 1 \oplus 1 = 0 \quad 0 \oplus 1 = 1 \quad 1 \oplus 0 = 1$$

a code **C** is called linear if the sum of two codes words is always a code word in **C**.

Channel Coding

Linear Block Codes

- **Hamming Weight and Distance**:
  - Let **c** be a code word of length *n*. The *Hamming weight* of **c**, denoted by $w(\mathbf{c})$ is the number of 1's in **c**.
  - Let **a** and **b** be code words of length *n*. The *Hamming distance* between **a** and **b**, denoted by $d(\mathbf{a},\mathbf{b})$, is the number of positions in which **a** and **b** differ.
  - Hamming distance can be written in terms of Hamming weight as

$$d(\mathbf{a},\mathbf{b}) = w(\mathbf{a} \oplus \mathbf{b})$$

Channel Coding

Linear Block Codes

- Consider the following code vectors:

$$\mathbf{c}_1 = [1 \quad 0 \quad 0 \quad 1 \quad 0]$$

$$\mathbf{c}_2 = [0 \quad 1 \quad 1 \quad 0 \quad 1]$$

- Find:
  - $w(\mathbf{c}_1)$ and $w(\mathbf{c}_2)$
  - $d(\mathbf{c}_1, \mathbf{c}_2)$
- Solution: $w(\mathbf{c}_1) = 2$, $w(\mathbf{c}_2) = 3$.

$$d(\mathbf{c}_1, \mathbf{c}_2) = 5$$

$$= w(\mathbf{c}_1 \oplus \mathbf{c}_2) = w[1 \quad 1 \quad 1 \quad 1 \quad 1] = 5$$

Channel Coding

Linear Block Codes

- The minimum distance *dmin* of linear code C is defined as the smallest *Hamming distance* between any pair of code words in C
- The minimum distance d*min* of linear code C is defined as the smallest *Hamming weight* of the non zero code vectors in the code

$$d\min = \min_{c \neq 0} w(c)$$

Channel Coding

Linear Block Codes

- **Error Detection and Correction Capabilities**:
  - *Theorem*: A linear code C of minimum distance d*min* can detect up to q errors if and only if

$$d\min \geq q+1$$

  - *Theorem*: A linear code C of minimum distance d*min* can correct up to q errors if and only if

$$d\min \geq 2q+1$$

Channel Coding

Linear Block Codes

- **Generator Matrix**: In an (n,k) linear block C, we define a code vector $\mathbf{c}$ and a data vector $\mathbf{d}$ as follows:

$$\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \dots & c_n \end{bmatrix}$$

$$\mathbf{d} = \begin{bmatrix} d_1 & d_2 & \dots & d_k \end{bmatrix}$$

- Here we assume that the first k bits of $\mathbf{c}$ are the data bits and the last $(n\text{-}k)$ bits are the *parity check bits* formed by linear combination of data bits.

Channel Coding

Linear Block Codes

- Thus

$$c_1 = d_1$$
$$c_2 = d_2$$
$$\vdots$$
$$c_{k+1} = p_{11}d_1 \oplus p_{12}d_2 \oplus \cdots \oplus p_{1k}d_k$$
$$c_{k+2} = p_{21}d_1 \oplus p_{22}d_2 \oplus \cdots \oplus p_{2k}d_k$$
$$\vdots$$
$$c_{k+m} = p_{m1}d_1 \oplus p_{m2}d_2 \oplus \cdots \oplus p_{mk}d_k$$

where $m = n\text{-}k$. This equation can be written in matrix form as

Channel Coding

Linear Block Codes

$$\mathbf{c} = \mathbf{dG} = \begin{bmatrix} d_1 & d_2 & \cdots & d_k \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & 0 & p_{11} & p_{21} & \cdots & p_{m1} \\ 0 & 1 & \cdots & 0 & p_{12} & p_{22} & \cdots & p_{m2} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1 & p_{1k} & p_{2k} & \cdots & p_{mk} \end{bmatrix}$$

where $\mathbf{G} = \begin{bmatrix} I_k & P^T \end{bmatrix}$

Here $\mathbf{I}_k$ is kth- order identity matrix and $\mathbf{P}^T$ is the transpose of matrix $\mathbf{P}$ given by

Channel Coding

Linear Block Codes

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \cdots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mk} \end{bmatrix}$$

- The k x n matrix $\mathbf{G}$ is called the *generator matrix.*

Channel Coding

Linear Block Codes

- **Parity-Check Matrix:** let $\mathbf{H}$ denotes an m x n matrix defined by $\mathbf{H} = \begin{bmatrix} \mathbf{P} & \mathbf{I}_m \end{bmatrix}$

  where $m = n - k$ and $\mathbf{I}_m$ is m*th*-order identity matrix. Then

$$\mathbf{H}^T = \begin{bmatrix} \mathbf{P}^T \\ \mathbf{I}_m \end{bmatrix}$$

- The matrix $\mathbf{H}$ is called the *parity-check matrix* of C.
- The minimum distance d*min* of a linear block code C is equal to the minimum number of rows of $\mathbf{H}^T$ that can be added to produce 0.

Channel Coding

Linear Block Codes

- **Syndrome Decoding**: Let **r** denote the received word of length *n* when code word **c** of length *n* was sent over a noisy channel.

  Consider a single error in the *i*th position has occurred. The error position can be identified by comparing the syndrome vector of **r** , defined as

  $$\mathbf{s} = \mathbf{r}\mathbf{H}^T$$

  to the rows of H^T.

- A Single error correcting codes are called *Hamming codes*

Channel Coding

Linear Block Codes

- For a (6,3) systematic linear code, the parity-check bits c_4, c_5, and c_6 are formed from the following equations:

$$c_4 = d_1 \oplus d_3$$
$$c_5 = d_1 \oplus d_2 \oplus d_3$$
$$c_6 = d_1 \oplus d_2$$

1. Write down the generator matrix **G**
2. Construct all possible code words
3. Suppose that the received word is 010111. Decode this received word by finding the location of the error and the transmitted data bits.

Channel Coding

Linear Block Codes

1. From the given equation we have

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

then

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{P}^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

2. Since $k = 3$, we have $2^3 = 8$ data words. Thus, if $d = [101]$, then

Channel Coding

Linear Block Codes

$$\mathbf{c} = \mathbf{dG} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

In a similar manner, the other code words can be constructed. They are listed in Table

d	c	d	c
000	000000	100	100111
001	001110	101	101001
010	010011	110	110100
011	011101	111	111010

Channel Coding

Linear Block Codes

3.

$$\mathbf{H}^T = \begin{bmatrix} \mathbf{P}^T \\ \mathbf{I}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Now **r** = [0 1 0 1 1 1], the syndrome **s** of **r** is

$$s = rH^T = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Channel Coding

Linear Block Codes

Since **s** is equal to the fourth row of $\mathbf{H}^T$, an error is at the fourth bit, the correct code word is 010011 and the data bits are 010.

Thank you !

*****