Lecture 16: Channel Coding

Prof. Deniz Gunduz

Department of Electrical & Electronic Engineering
Imperial College London

Outline

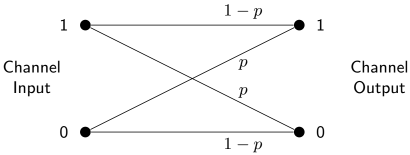
- Channel coding
- Linear block codes and generator matrix
- Hamming weight and distance
- Error detection and correction
- Reference
  - ✓ [Haykin] Chapter 10

Channel Coding: Noise and Errors

- Noise can corrupt the information during transmission
- Corruption of a signal should be avoided if possible
- Different systems will generally require different levels of protection against errors
- Consequently, a number of different channel coding techniques have been developed to detect and correct different types and number of errors

Channel Model

- Binary Symmetric Channel (BSC):



- Error probabilities are symmetric, errors are stationary and statistically independent
- p is presumed to be less than $1/2$, or $p < 1 - p$

Boolean Algebra I

- Two numbers: 0, 1
- Addition $+$: $0+0=1+1=0$; $0+1=1+0=1$
- Multiplication $\times$: $0\times 0=0\times 1=1\times 0=0$; $1\times 1=1$
- Calculation order: same as regular number calculation (multiplication first, from left to right), for example,

$$1\times 1+1\times 0+0=(1\times 1)+(1\times 0)+0=1+0+0=1$$

- Algebraic in Modulo 2:

$$\text{vector } \boldsymbol{a} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \text{matrix } \boldsymbol{B} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{Ba} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \times 1 + 1 \times 0 + 0 \times 1 \\ 1 \times 1 + 0 \times 0 + 1 \times 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Simple Error Checks: Repetition

If the error probability is small and information is fairly fault-tolerant, it is possible to use simple methods to detect errors (e.g., repetition, parity bits)

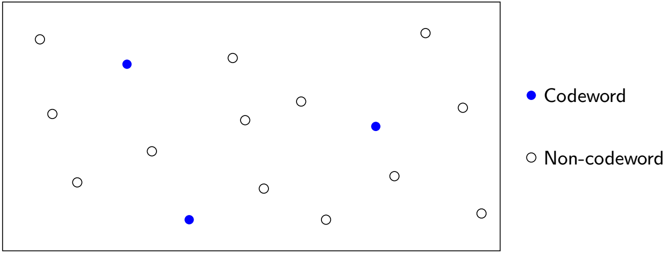
- Repetition – Repeating each bit in the message
  - ✓ If two symbols in an adjacent pair are different, it is likely that an error has occurred
  - ✓ However, this is not very efficient (bit rate is halved)
  - ✓ One repetition provides a means for error detection, but not for error correction
  - ✓ More repetitions are needed for error correction

Simple Error Checks: Adding a "Parity Bit"

- Parity bit – Use of a "parity bit" at the end of the message
  - ✓ A parity bit is a single bit that corresponds to the sum of the other message bits (modulo 2)
  - ✓ $\boldsymbol{u} = \begin{bmatrix} u_1 & \dots & u_k \end{bmatrix} \Rightarrow \boldsymbol{c} = \begin{bmatrix} c_1 & \dots & c_k & p \end{bmatrix}$, $p = c_1 + c_2 + \ldots + c_k$
  - ✓ For example, $011 \rightarrow 0110$; $010 \rightarrow 0101$
  - ✓ This allows any odd number of errors to be detected, but not even numbers
  - ✓ A single parity bit only allows error detection, not error correction
  - ✓ More efficient than simple repetition

An important class of codes that can detect and correct some errors are block codes

- Encode a series of symbols from the source, a "block", into a longer string: codeword or code block
- Error detection: if the received coded block is not a valid codeword
- Error correction: "decode" and associate a corrupted block to a valid coded block by its proximity (as measured by the "Hamming distance")



Linear Block Codes

An (n, k) binary linear block code takes a block of k bits of source data and encodes them using n bits.

$$\text{code rate} = \frac{\text{information bits}}{\text{codeword bits}} = \frac{k}{n}$$

- Linearity: the *Boolean sum* of any two codewords *must* be another codeword, e.g., if $\boldsymbol{a} = [1 \quad 0 \quad 0]$ and $\boldsymbol{b} = [1 \quad 0 \quad 1]$ are codewords, then $\boldsymbol{c} = \boldsymbol{a} + \boldsymbol{b} = [1+1 \quad 0+0 \quad 0+1] = [0 \quad 0 \quad 1]$ is, too

- The set of codewords forms a vector space, within which mathematical operations can be defined and performed

- To construct a linear block code we define a matrix, the generator matrix $\boldsymbol{G}$, which converts blocks of source symbols into longer blocks corresponding to codewords

- $\boldsymbol{G}$ is a $k \times n$ matrix (k rows, n columns) that takes a source block $\boldsymbol{u}$ (a binary vector of length k), to a codeword $\boldsymbol{x}$ (a binary vector of length n)

$$\boldsymbol{x} = \boldsymbol{u} \cdot \boldsymbol{G}$$

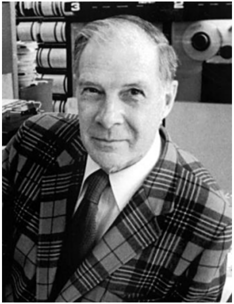
$$\begin{array}{c}\text{Message}\\ \text{vector } \boldsymbol{u}\end{array} \longrightarrow \boxed{\begin{array}{c}\text{Generator}\\ \text{matrix } \boldsymbol{G}\end{array}} \longrightarrow \begin{array}{c}\text{Code}\\ \text{vector } \boldsymbol{x}\end{array}$$

$$\boldsymbol{u} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{x} = \boldsymbol{u}\boldsymbol{G} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$$

- Linearity: Summation of two codewords is another codeword.
  - ✓ If $\boldsymbol{x}_1 = \boldsymbol{u}_1\boldsymbol{G}$, $\boldsymbol{x}_2 = \boldsymbol{u}_2\boldsymbol{G}$ are two codewords, then $\boldsymbol{x}_1 + \boldsymbol{x}_2 = \boldsymbol{u}_1\boldsymbol{G} + \boldsymbol{u}_2\boldsymbol{G} = (\boldsymbol{u}_1 + \boldsymbol{u}_2)\boldsymbol{G}$ is another codeword!

Hamming Weight

Richard Hamming (1915 – 1998) established code theory and method when he worked at AT&T Bell Labs, New Jersey, USA.



- Hamming weight of a binary vector $\boldsymbol{a}$ (written as $w_{\mathrm{H}}(\boldsymbol{a})$), is the number of non-zero elements it contains. For example:
  - ✓ 001110011 has a Hamming weight of 5
  - ✓ 000000000 has a Hamming weight of 0

- Hamming Distance between two binary vectors, $\boldsymbol{a}$ and $\boldsymbol{b}$, is written as $d_{\mathsf{H}}(\boldsymbol{a},\boldsymbol{b})$, and is equal to the Hamming weight of their (Boolean) sum

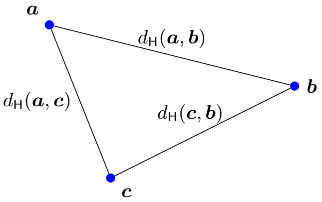
$$d_{\mathsf{H}}(\boldsymbol{a},\boldsymbol{b}) = w_{\mathsf{H}}(\boldsymbol{a}+\boldsymbol{b})$$

For example, 01110011 and 10001011 have a Hamming distance of

$$d_{\mathsf{H}}(01110011, 10001011) = w_{\mathsf{H}}(01110011 + 10001011) = w_{\mathsf{H}}(11111000) = 5$$

- Triangle inequality

$$d_{\mathsf{H}}(\boldsymbol{a},\boldsymbol{b}) \le d_{\mathsf{H}}(\boldsymbol{a},\boldsymbol{c}) + d_{\mathsf{H}}(\boldsymbol{c},\boldsymbol{b})$$



(7,4) Hamming Code

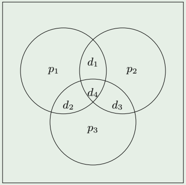
- Data bits: $\boldsymbol{u} = \begin{bmatrix} d_1 & d_2 & d_3 & d_4 \end{bmatrix}$
- Generator matrix:

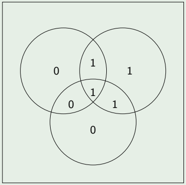
$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{P} \end{bmatrix}$$

- Codeword:

$$\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G} = \begin{bmatrix} d_1 & d_2 & d_3 & d_4 & d_2{+}d_3{+}d_4 & d_1{+}d_3{+}d_4 & d_1{+}d_2{+}d_4 \end{bmatrix}$$

- Parity bits: $p_1 = d_1{+}d_2{+}d_4, \quad p_2 = d_1{+}d_3{+}d_4, \quad p_3 = d_2{+}d_3{+}d_4$





Example

(7,4) Hamming Code

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

- Data bits: $\boldsymbol{u}_1 = [1\ 0\ 1\ 0] \Rightarrow$ codeword $\boldsymbol{x}_1 = \boldsymbol{u}_1\boldsymbol{G} = [1\ 0\ 1\ 0\ 1\ 0\ 1]$
- Data bits: $\boldsymbol{u}_2 = [1\ 1\ 0\ 1] \Rightarrow$ codeword $\boldsymbol{x}_2 = \boldsymbol{u}_2\boldsymbol{G} = [1\ 1\ 0\ 1\ 0\ 0\ 1]$
- Data bits: $\boldsymbol{u}_3 = [0\ 0\ 1\ 0] \Rightarrow$ codeword $\boldsymbol{x}_3 = \boldsymbol{u}_3\boldsymbol{G} = [0\ 0\ 1\ 0\ 1\ 1\ 0]$
- Calculate the Hamming distances between each pair of codewords $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3$ and compare them to the Hamming distances between each pair of data bits $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3$

$$d_{\mathsf{H}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = 4 > d_{\mathsf{H}}(\boldsymbol{u}_1, \boldsymbol{u}_2) = 3$$
$$d_{\mathsf{H}}(\boldsymbol{x}_1, \boldsymbol{x}_3) = 3 > d_{\mathsf{H}}(\boldsymbol{u}_1, \boldsymbol{u}_3) = 1$$
$$d_{\mathsf{H}}(\boldsymbol{x}_2, \boldsymbol{x}_3) = 7 > d_{\mathsf{H}}(\boldsymbol{u}_2, \boldsymbol{u}_3) = 4$$
$$\boldsymbol{x} = \boldsymbol{u}\boldsymbol{G} = \boldsymbol{u}\begin{bmatrix}\boldsymbol{I} & \boldsymbol{P}\end{bmatrix} = \begin{bmatrix}\boldsymbol{u} & \boldsymbol{u}\boldsymbol{P}\end{bmatrix}$$
$$d_{\mathsf{H}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = d_{\mathsf{H}}(\boldsymbol{u}_i, \boldsymbol{u}_j) + d_{\mathsf{H}}(\boldsymbol{u}_i\boldsymbol{P}, \boldsymbol{u}_j\boldsymbol{P})$$

- To determine the number of errors a particular code can *detect* and *correct*, we look at the minimum Hamming distance between any two codewords.
- From linearity, the zero vector must be a codeword. The minimum Hamming distance of a code is the same as minimum weight of non-zero codewords.
- We define the minimum distance between any two codewords to be

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

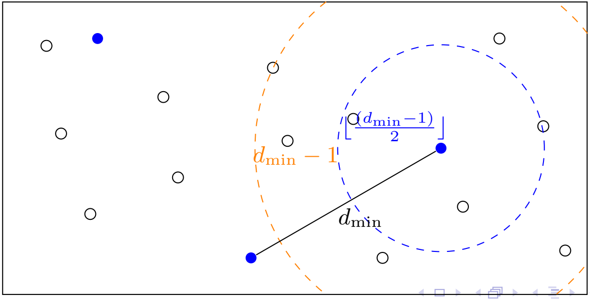
where $\mathcal{C}$ is the set of codewords.

Error Detection and Correction

- The number of errors that can be detected is then $d_{\min} - 1$ since $d_{\min}$ errors may turn an input codeword into a different valid codeword. Less than $d_{\min}$ errors will turn an input codeword into a vector that is not a valid codeword.

- Number t of errors that can be corrected is $t = \lfloor (d_{\min} - 1)/2 \rfloor$, simply the number of errors that can be detected divided by two and rounded down to the nearest integer since any output vector with less than this number of errors will be "nearer" to the input codeword.

- (7,4) Hamming code has $d_{\min} = 3$. It can detect one or two bit errors, and correct any single bit error.



Applications of Coding

- The first success was the application of convolutional codes in deep space probes 1960's-70's.
  - ✓ Mariner Mars, Viking, Pioneer missions by NASA
- Voyager, Galileo missions were further enhanced by concatenated codes (RS + convolutional).
- The next chapter was trellis coded modulation (TCM) for voice-band modems in 1980's.
- 1990's saw turbo codes approached capacity limit (now used in 3G).
- Followed by another breakthrough – space-time codes in 2000's (used in WiMax, 4G)
- The current frontier: LDPC, fountain codes, network coding, polar codes in 5G

Note