Chapter 8: Channel coding

Chapter 8

8.1. Introduction
8.2. Shannon second theorem
8.3. Decoding rules
8.4. Majority logic decoding
8.5. Hamming distance
8.6. Bound of length of the codeword
8.7. Detection/correction code construction
8.8. Parity code
8.9. Hamming code
8.10. Cyclic code

Remind

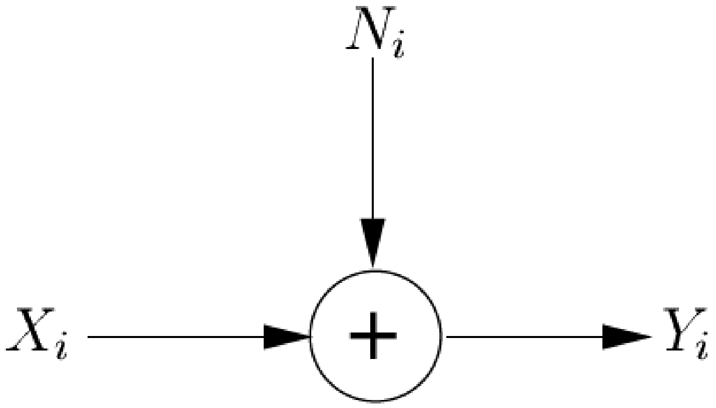
- Previous lesson:
  - What is the purpose of source coding?

  → Find methods to represent message with the minimum number of code symbols
  - Source coding is normally used for noiseless channel (information rate < channel capacity)
- If (information rate > channel capacity) → a part of information cannot be conveyed through the channel

  → need to have another coding for noisy channel
    - Channel coding

8.1. Introduction

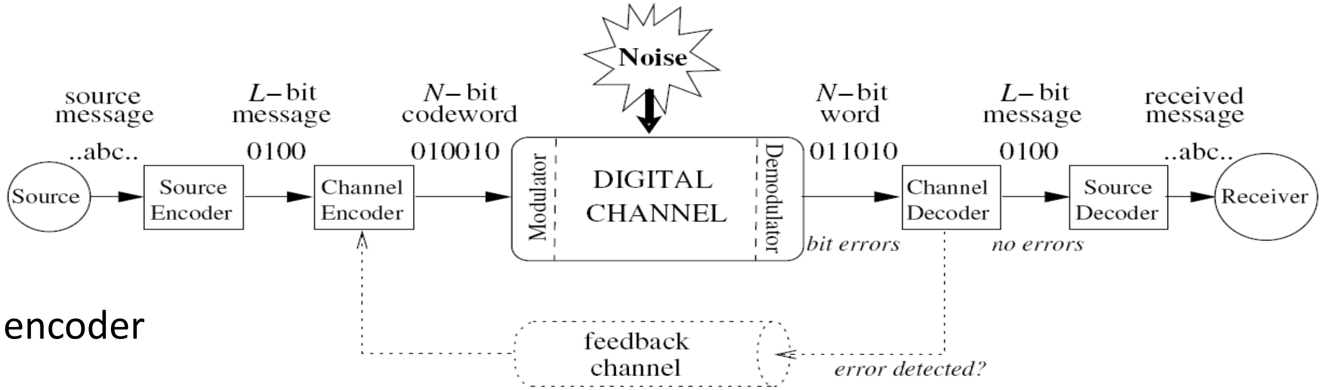
- Channel converts input signal into output signal with the affect of noise
  - Output = Input + Noise
  - Noise is considered as Gaussian random variable



$Y_i = X_i + N_i$

8.1. Introduction (Cont.)

- Noisy communication system:



- Channel encoder:
    - Its input is output of source encoder
        - Two ways to do:
            - Put directly the source encoder outputs on channel encoder
                - Called Continuous code
            - Divide the sequence of the source encoder outputs into blocks called L-symbol message of Channel encoder (L is length of block or number of code symbols of block)
                - In case of binary code → L-bit message
                - Called Block code
    - Output of channel encoder:
        - In case of continuous code: continuous code symbols come out
        - In case of block code: N-symbol codeword of channel encoder (codeword) (N>L)
            - In case of binary code → N-bit codeword

8.1. Introduction (cont.)

- Channel coder:
  - Traditionally, block code is used to present the theory of channel coding
  - Tasks of channel coding is to ensure reliable communication in noisy channel (prevent errors occur in the channel)
    - Error detection code
    - Error correction code
  - When information rate is greater than channel capacity, to prevent the information loss → add “dummy symbol” to the output of source coder

8.1. Introduction (cont.)

- Set of L-symbols message is denoted by M= $\{m_1, m_2, \ldots, m_{r^L})$ while
  - r: base of code
  - r^L is number of combinations of L-symbols message
  - Each m_i is L-symbols message to be input of channel encoder
    - m_i is also called as combination for carrying information
    - In source coding chapter, each symbol of source code has maximum amount of information → the symbols of m_i have identical probability→ all m_i have identical probability
  - In this chapter, L-symbols message is shortly called as message
- Channel coding converts L-symbols message into N-symbols codeword (denoted code (N,L))
  - Number of codeword = number of message = r^L
  - All codewords have similar probability
- Added symbols are called “check symbol” or “redundancy symbol”
  - Number of “check symbol” denote by R_N = N-L
- Code rate (R): the ratio between the number of code symbols that need to be transmitted and the number of code symbols that must be transmitted through encoding
  - In channel coding, R = $\frac{L}{N}$
  - Are also ratio between number of symbols of input message and the number of symbols of codeword

8.1. Introduction (cont.)

- Number of available combination of channel coding = r^N
  - Channel coding always has “don’t care combination”
    - Number of “don’t care combinations” = $r^N - r^L$ (number of available combination minus number of codeword)
    - Number of “don’t care combinations” > 0
    - Set of all “don’t care combinations” is denoted by B_N
- Set of N-symbols codeword is denoted by A = $\{a_1, a_2, \ldots, a_{r^L})$
  - Each a_i is a N-symbols codeword at the output of channel encoder
    - Denoted by a_i = $\{a_{i1}\ a_{i2}\ \ldots a_{iN}\ \}$, each a_{ij} is a code symbol
    - Each a_i is a codeword that encodes a m_i
    - Each codeword a_i becomes input of the channel
  - In this chapter, N-symbols codeword is shortly called as codeword
- A combination received at the channel output is denoted by b_j
  - b_j = $\{b_{j1}\ b_{j2}\ \ldots b_{jN}\ \}$, each b_{ij} is a code symbol
  - b_j = $a_i + e$
  - e = $\{e_1\ e_2\ \ldots e_N\}$: error combination that represents noise
    - e_i=0, no error at i^{th} position
    - e_i=1/…/r-1, error at i^{th} position
- When b_j is a codeword a_i, set of all b_j is denoted by B_M
- When b_j is not a codeword, set of all b_j is denoted by B_M^C

8.2. Shannon second theorem

- Let a discrete channel have the capacity C and a discrete source the information rate R.
  - If $R \leq C$ there exists a coding system such that the output of the source can be transmitted over the channel with an arbitrarily small frequency of errors
- Role of Shannon second theorem in channel coding?

→Permission to use coding for reliable communication in noisy channel

- How?

8.3. Decoding rules

- Rule to determine what is the transmitted codeword when a received combination (word) appears in output of the channel
- Let a_i be i^{th} N-symbol codeword that is transmitted through the channel
- Let b be the corresponding N-symbol word produced at the output of the channel
  - N-symbol word is a N-symbol combination that maybe a codeword or not
- The channel decoder has to apply a decoding rule on the received word b to decide which codeword a_i was transmitted
- The decision rule is denoted by D(.)
  - Correct decision a_i = D(b)
- Let $P_N(b|\, a_i)$ be the probability of receiving b given a_i was transmitted
- For a discrete memoryless channel this probability can be expressed in terms of the channel probabilities as follows:

$$P_N(\mathbf{b}|\mathbf{a}_i) = \prod_{t=1}^{N} P(b_t|a_{it})$$

8.3. Decoding rules (Cont.)

- According to Bayes rule:

$$P_N(\mathbf{a}_i|\mathbf{b}) = \frac{P_N(\mathbf{b}|\mathbf{a}_i)P_N(\mathbf{a}_i)}{P_N(\mathbf{b})}$$

- If the decoder decodes b into the codeword a_i (correct decoding) then
  - Correct probability is $P_N(a_i|b)$
  - Wrong probability is 1- $P_N(a_i|b)$
- To minimize the error, the codeword a_i should be chosen so as to maximize $P_N(a_i|b)$.

8.3. Decoding rules (Cont.)

- Minimum-error decoding rule:
  - To minimum wrong probability 1- $P_N(a_i|\, b)$ → maximize $P_N(a_i|\, b)$
  - Apply the Bayes rule:

$$P_N(a_i|b) = \frac{P_N(b|a_i)\ P_N(a_i)}{P_N(b)} \qquad P_N(a_j|b) = \frac{P_N(b|a_j)\ P_N(a_j)}{P_N(b)}$$

maximize $P_N(a_i|\, b)$ →maximize $P_N(b|a_i)\ P_N(a_i)$

choose $\boldsymbol{a_i}$ if $\boldsymbol{P_N(b|a_i)\ P_N(a_i) \geq P_N(b|a_j)\ P_N(a_j)}$ for all $\boldsymbol{a_j}$

- Maximum-likelihood decoding rule:

choose $\boldsymbol{a_i}$ if $\boldsymbol{P_N(b|a_i) \geq P_N(b|a_j)}$ for all $\boldsymbol{a_j}$

if $P_N(a_i)$ is identical for all *i*

8.3. Decoding rules (Cont.)

- For example:
  - BSC channel has channel matrix P, L=2, N=3. The codewords and its probabilities are shown in the above table. Output of the channel b=111

$$\mathbf{P} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$$

Code word	$P_N(\mathbf{a}_i)$
$\mathbf{a}_1 = (000)$	0.4
$\mathbf{a}_2 = (011)$	0.2
$\mathbf{a}_3 = (101)$	0.1
$\mathbf{a}_4 = (110)$	0.3

  - Minimum-error decoding rule?

8.4. Majority logic decoding

- Repetition code is the code that each message symbol is repeated in the codeword
  - Denoted by (n,m) where n is repetition time of one symbol, m is number of symbols of message
- Majority logic decoding is a method to decode repetition codes
  - Assumption that the largest number of occurrences of a symbol was the transmitted symbol

    → The most appearing symbol in the received combination is the transmitted symbol
- if a (n,1) binary repetition code is used, then each input bit is mapped to the codeword as a string of n-replicated input bits. Generally n=2t+1, an odd number. (t is arbitrary integer)

8.4. Majority logic decoding

- The repetition codes can detect up to t = (n-1)/2 transmission errors.
- Decoding algorithm
  - The codeword is (n,1), where n=2t+1, an odd number.
  - Calculate the d_H Hamming weight of the received repetition code.
    - d_H Hamming weight is the number of positions has non-zero values in the codeword
  - if d_H ≤ t, decoded codeword to be all 0’s
  - If $d_H > t$, decoded codeword to be all 1’s
- Example
  - In a (n,1) code, if received word R=1 0 1 1 0, then it would be decoded as n=5,t=2, d_H =3, so codeword =1 1 1 1 1
  - Hence the transmitted message bit was 1.

8.5. Hamming distance

- Consider the two N-symbol words a=$a_1 a_2 \ldots a_N$ and b= $b_1 b_2 \ldots b_N$
- The Hamming distance between a and b, d(a,b), is defined as the number of symbol positions in which a and b differ.
- The Hamming distance is a metric on the space of all symbol words of length N
- The Hamming distance obeys the following conditions:

  1. d(a,b) ≥ 0 with equality when a = b
  2. d(a,b) = d(b,a)
  3. d(a,b) + d(b,c) ≥ d(a,c) (triangle inequality)

8.5. Hamming distance (cont.)

- Example: Let N = 8

$$\mathbf{a} = 11010001$$
$$\mathbf{b} = 00010010$$
$$\mathbf{c} = 01010011$$

  - d(a,b) = 4, d(b,c) = 2, d(a,c) = 2
  - d(a,b) + d(b,c) = 4 + 2 ≥ d(a,c) = 2

8.5. Hamming distance (cont.)

- Hamming distance decoding rule
  - Let:
    - b is a received N-symbol word upon transmission one N-symbol a_i
    - $\mathrm{d(a,b)} = \mathrm{t}$, channel has t-symbol error
    - a is N-symbol word that decoder decide when decoder receives b
  - **If b = a_i then decide that a_i was sent**
  - **If b ≠ a_i then, for any a_j, decide a was sent if d(a,b) < d(a_j,b) for any j** (according to maximum-likelihood decoding rule)
    - If there is only one candidate a then
      - a was sent where t = d(a,b)
      - t-symbol error is corrected
    - If there is more than one candidate a, then
      - t-symbol error can only be detected (d(a,b)>0: error exist)

8.5. Hamming distance (cont.)

Message ($L = 2$)	**Code word** ($N = 3$)
00	000
01	001
10	011
11	111

- Example:
  - If received N-symbol word b = 000 → correct decision a = 000
  - If b ≠ 000 when t=1:

$\mathbf{b} = b_1 b_2 b_3$	**Closest code word**	**Action**
010	000 (b_2 in error), 011 (b_3 in error)	1-bit error detected
100	000 (b_1 in error)	1-bit error corrected
101	001 (b_1 in error), 111 (b_2 in error)	1-bit error detected
110	111 (b_3 in error)	1-bit error corrected

8.5. Hamming distance (cont.)

- Minimum distance of a code: the minimum Hamming distance between any two codewords of this code
  - d(K_N) = min (d(a,b)) while a and b are N-symbol codewords of N-symbol code. K_N is a code with length of N symbols
  - Example: K_N :

$$\begin{array}{l} 11010001 \\ 00010010 \\ 01010011 \end{array} \rightarrow d(K_N) = 2$$

8.5. Hamming distance (cont.)

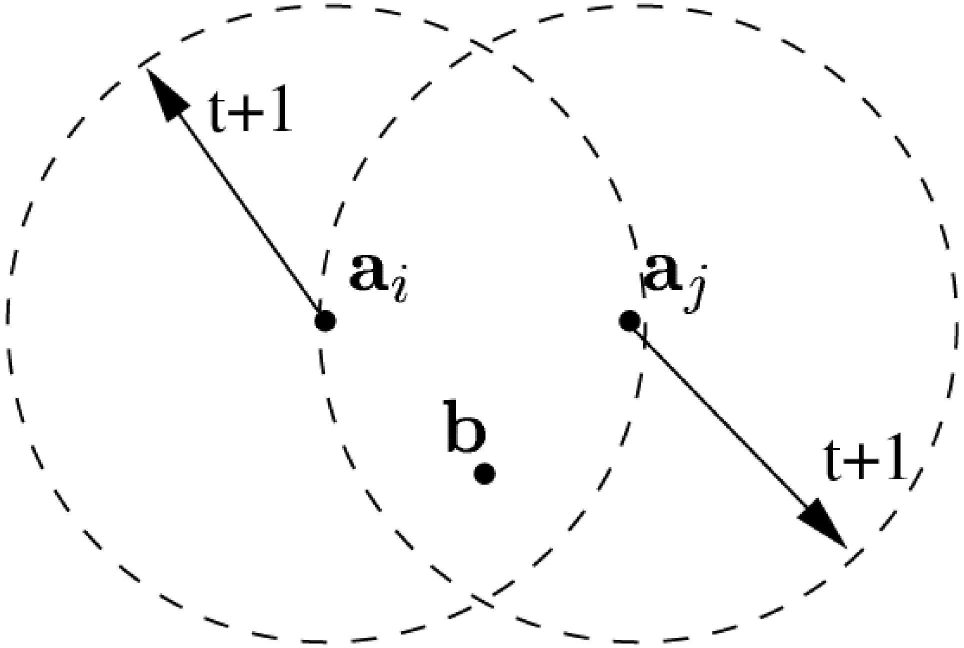
- Error detection and correction using Hamming distance:
  - t-symbol detection:
    - Block code, K_N, detects up to t errors if and only if its minimum distance is greater than t:

      $$d(K_N) > t \quad (8.1)$$

      - When the hamming distance between the two codewords is at least equal to t + 1, the wrong codeword turns into a non-codeword, so the error is detected.
      - (8.1): bound on the minimum distance of code of the detection code

8.5. Hamming distance (cont.)

- Error detection and correction using Hamming distance:
  - Diagram of t-symbol error detection for d(K_N) = t+1:
    - a_i, a_j are two N-symbol codewords. Each circle consists all words that have Hamming distance with codeword ≤ (t+1). Channel has only t-error, the received combination (b) must be in this circle

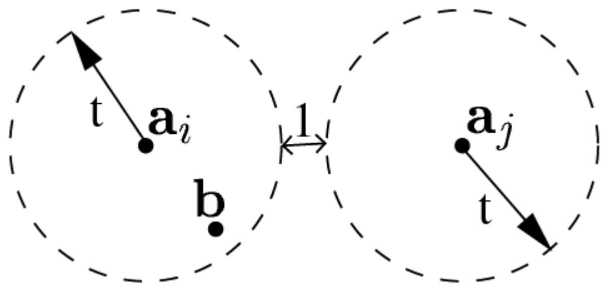


8.5. Hamming distance (cont.)

- Error detection and correction using Hamming distance:
  - t-symbol correction:
    - Block code, K_N, correct up to t errors if and only if its minimum distance is greater than t:

      d(K_N) > 2t (8.2)

      - When the hamming distance between the two codewords is at least equal to 2t + 1, The combinations received when transmitting a code word are separated, so the error is corrected.
      - (8.2): bound on the minimum distance of code of the correction code
  - Diagram of t-symbol error correction for d(K_N) = 2t+1:
    - a_i, a_j are two N-symbol codewords. Each circle consists all words that have Hamming distance with codeword ≤ t.



8.5. Hamming distance (cont.)

- Example

Message	Code word
00	000
01	011
10	101
11	110

d(K_N) = 2

→detect only one error(t=1) because d(K_N) > t, not correct because d(K_N) <= 2t

8.5. Hamming distance (cont.)

- Example

Message	Code word
0	000
1	111

- d(K_N) = 3 → detect two errors, correct one error