16.682: Communication Systems Engineering

Lecture 14: Channel Coding

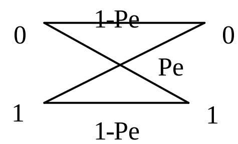
April 5, 2001

Eytan Modiano

Channel Coding

- When transmitting over a noisy channel, some of the bits are received with errors

Example: Binary Symmetric Channel (BSC)



Pe = Probability of error

- Q: How can these errors be removed?

- A: Coding: the addition of redundant bits that help us determine what was sent with greater accuracy

Example (Repetition code)

Repeat each bit n times (n-odd)

Input	Code
0	000……..0
1	11..……..1

<u>Decoder:</u>

- **If received sequence contains n/2 or more 1's decode as a 1 and 0 otherwise**
  - **Max likelihood decoding**

P (error | 1 sent) = P (error | 0 sent)
= P[more than n / 2 bit errors occur]

$$= \sum_{i=\lceil n/2 \rceil}^{n} \binom{n}{i} P_e^{\,i} (1-P_e)^{n-i}$$

Repetition code, cont.

- For P_e < 1/2, P(error) is decreasing in n
  - Þ for any e, $ n large enough so that P (error) < e

Code Rate: ratio of data bits to transmitted bits

- For the repetition code R = 1/n
- To send one data bit, must transmit n channel bits “bandwidth expansion”

- In general, an (n,k) code uses n channel bits to transmit k data bits
  - Code rate R = k / n

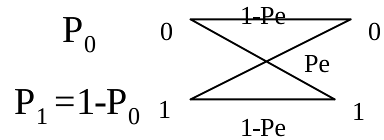
- Goal: for a desired error probability, e, find the highest rate code that can achieve p(error) < e

Channel Capacity

- **The capacity of a discrete memoryless channel is given by,**

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

<u>Example</u>: Binary Symmetric Channel (BSC)



$I(X;Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$

$H(X|Y) = H(X|Y=0)*P(Y=0) + H(X|Y=1)*P(Y=1)$

$H(X|Y=0) = H(X|Y=1) = P_e \log(1/P_e) + (1-P_e)\log(1/ 1-P_e) = H_b(P_e)$

$H(X|Y) = H_b(P_e) \Rightarrow I(X;Y) = H(X) - H_b(P_e)$

$H(X) = P_0 \log(1/P_0) + (1-P_0)\log(1/ 1-P_0) = H_b(p_0)$

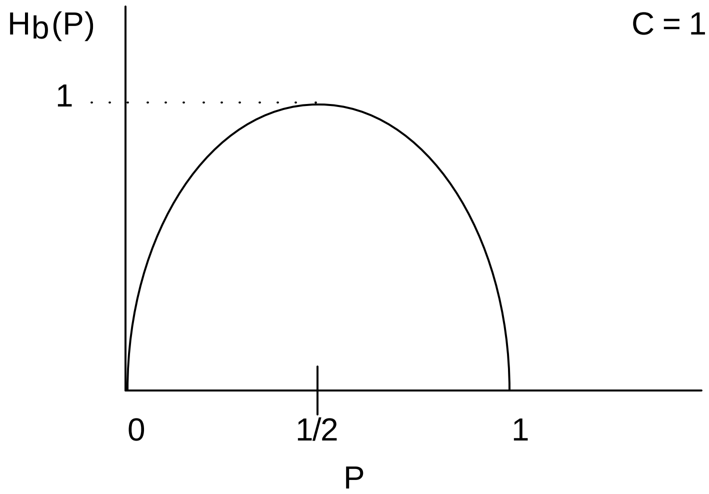
$\Rightarrow I(X;Y) = H_b(P_0) - H_b(P_e)$

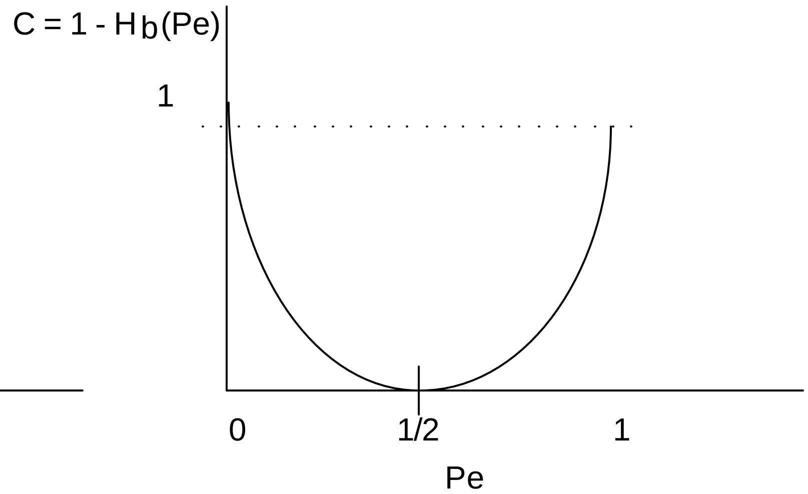
Capacity of BSC

$$I(X;Y) = H_b(P_0) - H_b(P_e)$$

- $H_b(P) = P \log(1/P) + (1-P) \log(1/\ 1-P)$
  - $H_b(P) <= 1$ with equality if $P=1/2$

$$C = \max_{P0} \{I(X;Y) = H_b(P_0) - H_b(P_e)\} = 1 - H_b(P_e)$$





$C = 0$ when $P_e = 1/2$ and $C = 1$ when $P_e = 0$ or $P_e=1$

Channel Coding Theorem (Claude Shannon)

Theorem: For all R < C and e > o; there exists a code of rate R whose error probability < e

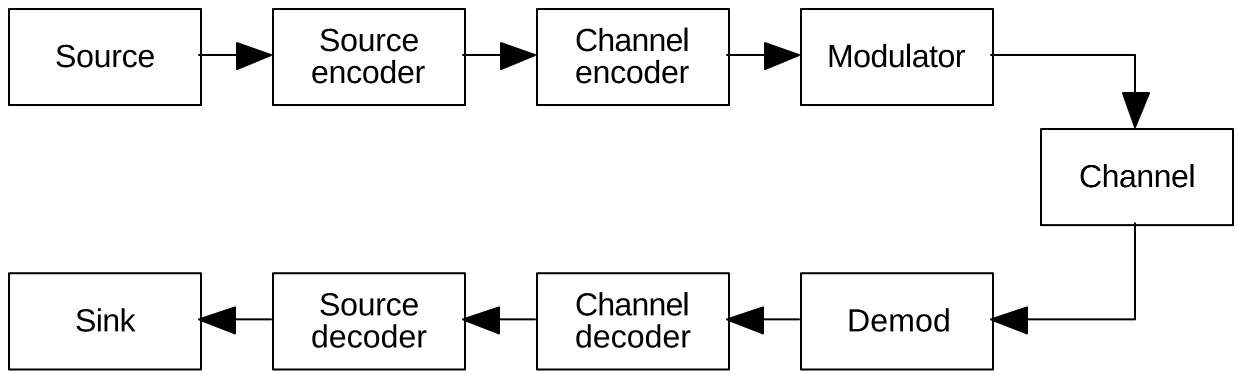
- e can be arbitrarily small
- Proof uses large block size n
  - as n ® ¥ capacity is achieved

- In practice codes that achieve capacity are difficult to find
  - The goal is to find a code that comes as close as possible to achieving capacity

- Converse of Coding Theorem:
  - For all codes of rate R > C, $ e_0 > 0, such that the probability of error is always greater than e_0
    - For code rates greater than capacity, the probability of error is bounded away from 0

Channel Coding

- **Block diagram**



Approaches to coding

- **Block Codes**
  - Data is broken up into blocks of equal length
  - Each block is “mapped” onto a larger block

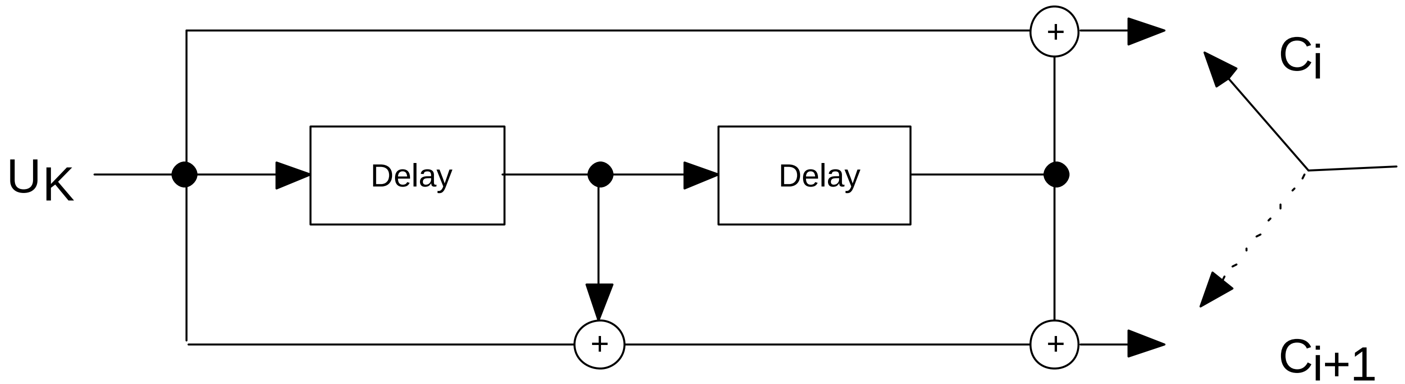
___Example: (6,3) code, n = 6, k = 3, R = 1/2

000 ® 000000	100 ® 100101
001 ® 001011	101 ® 101110
010 ® 010111	110 ® 110010
011 ® 011100	111 ® 111001

- **An (n,k) binary block code is a collection of 2^k binary n-tuples (n>k)**
  - n = block length
  - k = number of data bits
  - n-k = number of checked bits
  - R = k / n = code rate

Approaches to coding

- **Convolutional Codes**
  - **The output is provided by looking at a sliding window of input**



$$C_{(2K)} = U_{(2K)} \oplus U_{(2K-2)}, \quad C_{(2K+1)} = U_{(2K+1)} \oplus U_{(2K)} \oplus U_{(2K-1)}$$

$\oplus$ mod(2) addition (1+1=0)

Block Codes

- A block code is systematic if every codeword can be broken into a data part and a redundant part
  - Previous (6,3) code was systematic

Definitions:

- **Given X Î $\{0,1\}^n$, the Hamming Weight of X is the number of 1's in X**

- **Given X, Y in $\{0,1\}^n$, the Hamming Distance between X & Y is the number of places in which they differ,**

$$d_H(X,Y) = \sum_{i=1}^{n} X_i \oplus Y_i = Weight(X+Y)$$

$$X+Y = [x_1 \oplus y_1, x_2 \oplus y_2, \cdots, x_n \oplus y_n]$$

- **The minimum distance of a code is the Hamming Distance between the two closest codewords:**

$$d_{min} = \min \{d_H(C_1,C_2)\}$$
$$C_1,C_2 \text{ Î } C$$

Decoding

- r may not equal to u due to transmission errors
- Given r how do we know which codeword was sent?

__Maximum likelihood Decoding__:

Map the received n-tuple r into the codeword C that maximizes,

P { r | C was transmitted }

__Minimum Distance Decoding__ (nearest neighbor)

Map r to the codeword C such that the hamming distance between r and C is minimized (I.e., min d_H (r,C))

Þ **For most channels Min Distance Decoding is the same as Max likelihood decoding**

Linear Block Codes

- **A (n,k) linear block code (LBC) is defined by 2^k codewords of length n**

$$C = \{ C_1 \ldots . C_m \}$$

- **A (n,k) LBC is a K-dimensional subspace of $\{0,1\}^n$**
  - **(0…0) is always a codeword**
  - **If $C_1, C_2 \in C$, $C_1 + C_2 \in C$**

- **Theorem: For a LBC the minimum distance is equal to the min weight (W_{min}) of the code**

$$W_{min} = \min_{(\text{over all } Ci)} \text{Weight}(C_i)$$

<u>Proof</u>: Suppose $d_{min} = d_H(C_i, C_j)$, where $C_1, C_2 \in C$

$d_H(C_i, C_j) = \text{Weight}(C_i + C_j)$,
but since C is a LBC then $C_i + C_j$ is also a codeword

Systematic codes

Theorem: Any (n,k) LBC can be represented in Systematic form
where: data = $x_1..x_k$, codeword = $x_1..x_k\, c_{k+1}..x_n$

– Hence we will restrict our discussion to systematic codes only

- The codewords corresponding to the information sequences:
  $e_1 = (1,0,..0)$, $e_2=(0,1,0..0)$, $e_k = (0,0,..,1)$ for a basis for the code
  – Clearly, they are linearly independent
  – K linearly independent n-tuples completely define the K dimensional subspace that forms the code

Information sequence	Codeword
$e_1 = (1,0,..0)$	$g_1 = (1,0,..,0, g_{(1,k+1)} \ldots g_{(1,n)})$
$e_2=(0,1,0..0)$	$g_2 = (0,1,..,0, g_{(2,k+1)} \ldots g_{(2,n)})$
$e_k = (0,0,..,1)$	$g_k = (0,0,..,k, g_{(k,k+1)} \ldots g_{(k,n)})$

- $g_1, g_2, \ldots, g_k$ form a basis for the code

The Generator Matrix

$$G = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_k \end{bmatrix} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & & & g_{2n} \\ \vdots & & & \\ g_{k1} & & & g_{kn} \end{bmatrix}$$

- **For input sequence x = (x_1,…,x_k): C_x = xG**
  - **Every codeword is a linear combination of the rows of G**
  - **The codeword corresponding to every input sequence can be derived from G**
  - **Since any input can be represented as a linear combination of the basis (e_1,e_2,…, e_k), every corresponding codeword can be represented as a linear combination of the corresponding rows of G**
- **Note: $x_1 \longleftrightarrow C_1$, $x_2 \longleftrightarrow C_2$ => $x_1+x_2 \longleftrightarrow C_1+C_2$**

Example

- **Consider the (6,3) code from earlier:**

100 ® 100101; 010 ® 010111; 001 ® 001011

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

Codeword for (1,0,1) = (1,0,1)G = (1,0,1,1,1,0)

$$G = \left[\begin{array}{c:c} I_K & P_{Kx(n-K)} \end{array} \right]$$

$$\mathrm{I_K} = \text{KxK identity matrix}$$

The parity check matrix

$$H = \left[\begin{array}{c:c} P^T & I_{(n-K)} \end{array}\right]$$

$$\mathrm{I}_{(\text{n-K})} = (n - \mathrm{K})\mathrm{x}(\mathrm{n} - \mathrm{K})\ \text{identity matrix}$$

Example:

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

Now, if c_i is a codework of C then, $c_i H^T = \vec{0}$

- "C is in the null space of H"
- Any codeword in C is orthogonal to the rows of H

Decoding

- v = transmitted codeword = $v_1 \ldots v_n$
- r = received codeword = $r_1 \ldots r_n$
- e = error pattern = $e_1 \ldots e_n$

- r = v + e

- S = rH^T = Syndrome of r

  = $(v+e)H^T = vH^T + eH^T = eH^T$

- S is equal to ‘0’ if and only if e Î C
  – I.e., error pattern is a codeword

- S ¹ 0 => error detected
- S = 0 => no errors detected (they may have occurred and not detected)

- Suppose S ¹ 0, how can we know what was the actual transmitted codeword?

Syndrome decoding

- **Many error patterns may have created the same syndrome**

  For error pattern $e_0 \Rightarrow S_0 = e_0H^T$

  Consider error pattern $e_0 + c_i$ (c_i Î C)

  $$S'_0 = (e_0 + c_{i)})H^T = e_0\, H^T + c_i\, H^T = e_0\, H^T = S_0$$

- **So, for a given error pattern, e_0, all other error patterns that can be expressed as $e_0 + c_i$ for some c_i Î C are also error patterns with the same syndrome**

- **For a given syndrome, we can not tell which error pattern actually occurred, but the most likely is the one with minimum weight**
  - **Minimum distance decoding**

- **For a given syndrome, find the error pattern of minimum weight (e_{min}) that gives this syndrome and decode: $r' = r + e_{min}$**

Standard Array

$$M = 2^K$$

$$\begin{array}{ccccc} C_1 & C_2 & \cdots & C_M & Syndrome \\ e_1 & e_1 + C_2 & & e_1 + C_M & S_1 \\ \vdots & e_2 + C_2 & & e_2 + C_M & S_2 \\ e_{2^{(n-K)}-1} & & & & S_{2^{(n-K)}-1} \end{array}$$

- **Row 1 consists of all M codewords**
- **Row 2 e_1 = min weight n-tuple not in the array**
  - **I.e., the minimum weight error pattern**
- **Row i, e_i = min weight n-tuple not in the array**

- **All elements of any row have the same syndrome**
  - **Elements of a row are called “co-sets”**
- **The first element of each row is the minimum weight error pattern with that syndrome**
  - **Called “co-set leader”**

Decoding algorithm

- Receive vector r

1) Find $S = rH^T$ = syndrome of r

2) Find the co-set leader e, corresponding to S

3) Decode: $C = r+e$

- "Minimum distance decoding"
  - Decode into the codeword that is closest to the received sequence

Example (syndrome decoding)

- **Simple (4,2) code**

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

Data	**codeword**
00	**0000**
01	**0101**
10	**1010**
11	**1111**

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad H^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

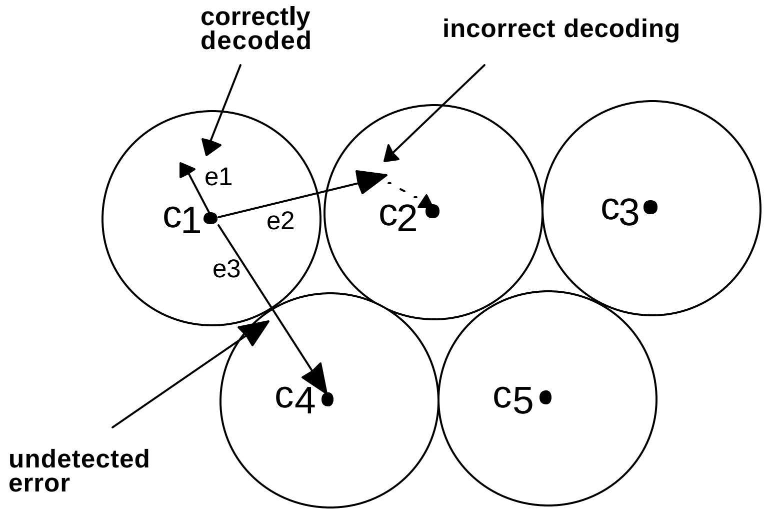
Standard array

0000	**0101**	**1010**	**1111**	**Syndrome**
1000	**1101**	**0010**	**0111**	**10**
0100	**0001**	**1110**	**1011**	**01**
1100	**1001**	**0110**	**0011**	**11**

Suppose 0111 is received, S = 10, co-set leader = 1000

Decode: C = 0111 + 1000 = 1111

Minimum distance decoding



- Minimum distance decoding maps a received sequence onto the nearest codeword
- If an error pattern maps the sent codeword onto another valid codeword, that error will be undetected (e.g., e3)
  - Any error pattern that is equal to a codeword will result in undetected errors
- If an error pattern maps the sent sequence onto the sphere of another codeword, it will be incorrectly decoded (e.g., e2)

Performance of Block Codes

- **Error detection: Compute syndrome, $S \neq 0$ => error detected**
  - **Request retransmission**
  - **Used in packet networks**

- **A linear block code will detect all error patterns that are not codewords**

- **Error correction: Syndrome decoding**
  - **All error patterns of weight < $d_{min}/2$ will be correctly decoded**
  - **This is why it is important to design codes with large minimum distance (d_{min})**
  - **The larger the minimum distance the smaller the probability of incorrect decoding**

Hamming Codes

- **Linear block code capable of correcting single errors**
  - **$n = 2^m - 1$, $k = 2^m - 1 - m$**
    (e.g., (3,1), (7,4), (15,11)…)
  - **$R = 1 - m/(2^m - 1)$ => very high rate**
  - **$d_{min} = 3$ => single error correction**

- **Construction of Hamming codes**
  - **Parity check matrix (H) consists of all non-zero binary m-tuples**

Example: (7,4) hamming code (m=3)

$$H = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \qquad G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$