ELG3175 Introduction to Communication Systems

Introduction to Error Control Coding

Types of Error Control Codes

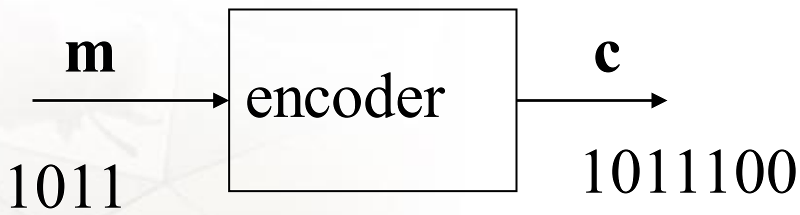
- **Block Codes**
  - **Linear**
    - Hamming, LDPC
  - Non-Linear
  - Cyclic
    - BCH, RS
- Convolutional Codes
- Turbo Codes

Parity Bits

- Suppoose we wish to transmit **m**=[1001001].
- Let us assume that the second bit is received in error, **r** = [1101001].
- The receiver has no way of knowing that the second bit has been incorrectly detected, therefore we must accept the consequences of the detection error.
- Suppose, before transmission, we add an even parity bit to the message to be transmitted, $\mathbf{m}_c$= [1001001*1*].
- Now, let us assume that the second bit is in error, **r** = [1101001*1*]. There are now 5 1's, which is not permitted. Therefore the error is detected and the receiver can request a retransmission.
- The detection of the error was made possible by the addition of the parity bit.

Block Codes

- The data is grouped into segments of k bits.
- Each block of k bits is encoded to produce a block of n bits. where $n>k$. The encoder adds redundancy to the data to be transmitted.
- The code rate is $r = k/n$.



Binary addition and multiplication

- 0+0 = 0, 0+1 = 1, 1+0 = 1 and 1+1=0 (there is no carry).
- 0x = 0 where x = 0 or 1. 1x = x where x = 0 or 1.

- Examples 1010 + 1100 = 0110. 0(10010) = (00000).

Linear Block Codes

- Let C be a code made up of the vectors $\{\mathbf{c}_1, \mathbf{c}_2, \ldots c_K\}$.
- C is a linear code if for any $\mathbf{c}_i$ and $\mathbf{c}_j$ in C, $\mathbf{c}_i+\mathbf{c}_j$ is also in C.
- Example $C = \{\mathbf{c}_1 = 0000, \mathbf{c}_2 = 0110, \mathbf{c}_3 = 1001, \mathbf{c}_4 = 1111\}$.
- $\mathbf{c}_1+\mathbf{c}_x = \mathbf{c}_x$ for $x = 1, 2, 3$ ou 4.
- $\mathbf{c}_x+\mathbf{c}_x = \mathbf{c}_1$.
- $\mathbf{c}_2+\mathbf{c}_3 = \mathbf{c}_4$, $\mathbf{c}_3+\mathbf{c}_4 = \mathbf{c}_2$, $\mathbf{c}_2+\mathbf{c}_4 = \mathbf{c}_3$.
- C is a linear code.
- $C_2 = \{\mathbf{c}_1 = 0001, \mathbf{c}_2 = 0111, \mathbf{c}_3 = 1000, \mathbf{c}_4 = 1110\}$.
- $\mathbf{c}_x+\mathbf{c}_x = 0000$ which is not in C_2.
- C_2 is not linear.

Hamming Weight

- For codeword $\mathbf{c}_x$ of code C, its *Hamming Weight* is the number of symbols in $\mathbf{c}_x$ that are not 0.
- C = {0000 0110 1001 1111}
- H.W{0000} = 0
- H.W{0110} = 2
- H.W{1001} = 2
- H.W{1111} = 4

Hamming Distance

- The *Hamming Distance* between codewords $\mathbf{c}_i$ and $\mathbf{c}_j$ of C is the number of positions in which they differ.

	0000	0110	1001	1111
0000	0	2	2	4
0110	2	0	4	2
1001	2	4	0	2
1111	4	2	2	0

- $\mathbf{c}_i+\mathbf{c}_j = 0$ in the positions in which they are the same and $\mathbf{c}_i+\mathbf{c}_j = 1$ in the positions in which they differ. Therefore $HD\{\mathbf{c}_i, \mathbf{c}_j\} = HW\{\mathbf{c}_i+\mathbf{c}_j\}$.

Minimum Distance

- A code's minimum distance is the minimum Hamming distance between two different codewords in the code.
- In our example, d_{min} = 2.
- We saw previously HD{$\mathbf{c}_i$,c_j} = HW{$\mathbf{c}_i+\mathbf{c}_j$} = HW{$\mathbf{c}_x$} where, in the case of linear block codes, $\mathbf{c}_x$ is another codeword in C excluding the all-zero codeword.
  – Therefore for linear block codes, d_{min} = minimum Hamming weight of all codewords in C excluding the all-zero codeword.
- In our example, if we exclude codeword 0000, the remaining codewords are 0110, 1001 and 1111. The minimum Hamming weight is 2. Therefore d_{min} = 2.

Basis of a linear block code

- C is a linear block code.
- Let us choose k linearly independent codewords, $\mathbf{c}_1$, $\mathbf{c}_2$, …, $\mathbf{c}_k$. None of these k codewords can be expressed as a linear combination of the others.
- All codewords in C can then be expressed as a linear combination of these k codewords.
  - The k codewords selected form the basis of code C.
- $c_x = a_1\mathbf{c}_1+a_2\mathbf{c}_2+a_3\mathbf{c}_3+\ldots+a_k\mathbf{c}_k$ where a_i = 0 ou 1 (binary blcok codes).
- In our example, we can select 0110 and 1111, or 0110 and 1001 or 1001 and 1111.
- Example, let us select $\mathbf{c}_1$ = 0110 and $\mathbf{c}_2$ = 1111 as the basis of the code.
  - $0000 = 0\mathbf{c}_1+0\mathbf{c}_2$, $0110 = 1\mathbf{c}_1+0\mathbf{c}_2$, $1001 = 1\mathbf{c}_1+1\mathbf{c}_2$ et $1111 = 0\mathbf{c}_1+1\mathbf{c}_2$.

Generator Matrix

$$\mathbf{G} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_k \end{bmatrix}$$

$$\mathbf{c}_x = \mathbf{m}_x \mathbf{G}$$

Example

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

$$[0 \quad 0]\mathbf{G} = [0 \quad 0 \quad 0 \quad 0]$$

$$[0 \quad 1]\mathbf{G} = [1 \quad 1 \quad 1 \quad 1]$$

$$[1 \quad 0]\mathbf{G} = [0 \quad 1 \quad 1 \quad 0]$$

$$[1 \quad 1]\mathbf{G} = [1 \quad 0 \quad 0 \quad 1]$$

The dimensions of $\mathbf{G}$ are $k \times n$.

Equivalent codes

- The codes generated by $\mathbf{G}_1$ and $\mathbf{G}_2$ are equivalent if they generate the same codewords but with a different mapping to message words.
- Example

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \mathbf{G}_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

m	00	0000	0000
	01	1111	1111
	10	0110	1001
	11	1001	0110

Systematic codes

- A code is systematic if the message bits can be found at the beginning of the codeword.
- $\mathbf{c} = [\mathbf{m}|\mathbf{p}]$.
- $\mathbf{G}_{syst} = [\mathbf{I}_k|\mathbf{P}]$.
- Any generator matrix can be transformed into $\mathbf{G}_{syst}$ using linear transformation.

m	
00	0000
01	0110
10	1001
11	1111

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

Parity Check Matrix

- A parity check matrix H is a matrix with property $\mathbf{c}\mathbf{H}^T = \mathbf{0}$.
- $\mathbf{c}\mathbf{H}^T = 0$ can be written as $\mathbf{m}\mathbf{G}\mathbf{H}^T = \mathbf{0}$
- Therefore $\mathbf{G}\mathbf{H}^T = \mathbf{0}$.
- We can find **H** from $\mathbf{G}_{syst}$.
- $\mathbf{H} = [\mathbf{P}^T | \mathbf{I}_{n-k}]$.

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

- **H** has dimensions $(n-k)\times n$.

uOttawa

Example Hamming (7,4) code

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

Find all of the codewords, find d_{min}, find $\mathbf{H}$.

Decoding

- The received word, **r** = **c**+**e**, where e = error pattern.
- For example if **c** = (1 1 0 0 1 1 0 1) and **r** = (1 0 0 0 1 1 0 1), then **e** = (0 1 0 0 0 0 0 0).
- Assuming that errors occur independently with probability $p < 0.5$
  - Therefore, code bits are correctly detected with probability $(1-p)$
- Lower weight error patterns are more probable than higher weight ones.

Example

- C = {(00000) (01011) (10110) (11101)}
- **r** = (11111)
- If **c** = (00000), then **e** = (11111) which occurs with probability p^5.
- If **c** = (01011), then **e** = (10100) which occurs with probability $p^2(1-p)^3$.
- If **c** = (10110), then **e** = (01001) which occurs with probability $p^2(1-p)^3$.
- If **c** = (11101), then **e** = (00010) which occurs with probability $p(1-p)^4 > p^2(1-p)^3 > p^5$.
- Therefore receiver selects **c** = (11101) as most likely transmitted codeword and outputs message that corresponds to this codeword.

Standard Array Decoding

- Lookup table that maps received words to most likely transmitted codewords.
- Each received word points to a memory address which holds the value of the most likely transmitted word.

00000	01011	10110	11101
00001	01010	10111	11100
00010	01001	10100	11111
00100	01111	10010	11001
01000	00011	11110	10101
10000	11011	00110	01101
10001	11010	00111	01100
11000	10011	01110	00101

uOttawa

How to Build Standard Array

- Write out all possible received words.
- Remove all codewords and place at top of columns with all-zero codeword at left side (left most column corresponds to error pattern)
- Take lowest weight vector from remaining words and place in left column. Add this vector to all codewords and place result below that codeword.
  - Remove all of these results from list of all possible received words.
- Repeat until list of possible received words is exhausted

Syndrome decoding

- $\mathbf{S} = \mathbf{r}\mathbf{H}^T$.
- $\mathbf{r}=\mathbf{c}+\mathbf{e}$, therefore $\mathbf{S} = (\mathbf{c}+\mathbf{e})\mathbf{H}^T = \mathbf{c}\mathbf{H}^T + \mathbf{e}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T$.
- All vectors in the same row of the standard array produce the same syndrome.
- Syndrome points to a memory address whcih contains the most likely error pattern, then decoder computes $\mathbf{c} = \mathbf{r}+\mathbf{e}$.

Example

- For our code:

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

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

Example continued

- Suppose **r** = (01001), then

$$(0 \quad 1 \quad 0 \quad 0 \quad 1)\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

- This indicates that the 4th bit is in error : **e** = (00010) and **c** = (01011).

Error correcting and Error Detecting Capabilities of a code

- t = number of error that decoder can always correct.
- J = number of errors that decoder can always detect.
- $t = (d_{min}-1)/2$ (d_{min} is odd) or $(d_{min}-2)/2$ (d_{min} is even).
- $J = d_{min} - 1$
- We can have codes that both correct and detect errors, then $t+j = d_{min} - 1$ where $j > t$.

uOttawa

Performance: Decoder Failure

- Probability of decoder failure = probability that decoder selects the incorrect codeword = probability that error pattern is not one of the error patterns that it can correct
  - In our example, the decoder can correct all 5 error patterns of weight 1 and 2 error patterns of weight two. The probability that the error pattern IS one of these is $(1\text{-}p)^5+5p(1\text{-}p)^4 + 2p^2(1\text{-}p)^3$. Therefore $P(E) = 1- (1\text{-}p)^5\text{-}5p(1\text{-}p)^4 - 2p^2(1\text{-}p)^3$
  - In many cases, the code has too many codewords to construct a standard array.
  - But we usually know d_{min}, therefore we know t.

Performance: Decoder Failure

$$P(E) = 1 - \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i}$$

Performance: Bit Error Rate

- $(1/k)P(E) < P_b < P(E)$

Performance: Probability Undetected Error

- P(U) = probability that an error is undetected = probability that syndrome = 0 even if error pattern is not 0 = probability that error pattern is same as a codeword.
- In our example $P(U) = 2p^3(1-p)^2 + p^4(1-p)$.
- If we don't know the codewords because code is too large, then P(U) < probability error pattern has weight greater than j = 1 – probability that error pattern has weight j or less

$$P(U) < 1 - \sum_{i=0}^{j} \binom{n}{i} p^i (1-p)^{n-i}$$

uOttawa