Error-control coding

A good communication system requires transmitting information from one end of the system at a rate and a level of reliability and quality that are acceptable to a user at the other end.

- Two key systems parameters are transmitted signal power and channel bandwidth. In general,
  - Signal power ↑ error rate ↓
  - Bandwidth ↑ error rate ↓
- In practice, we often arrive at a modulation scheme and find that it is not possible to provide acceptable data quality.
- ∴ **Error-control coding**

Feed-Forward Error Correction (FEC)

The channel encoder in the transmitter accepts message bits and adds **redundancy** according to a prescribed rule, thereby producing encoded data at a higher bit rate.

The channel decoder in the receiver exploits the redundancy to decide which message bits were actually transmitted.

Automatic-Repeat Request (ARQ)

FEC is used for both the detection and correction of errors.

- Only a one-way link between the transmitter and receiver is required.

ARC uses redundancy merely for the purpose of error detection. Upon the detection of an error in a transmitted code word, the receiver requests a repeat transmission of the corrupted code word, which need a return path.

- Used only on half-duplex or full-duplex links
- widely used in computer-communication systems

Classification of channel coding

Block Codes

- has no memory
- adding redundant bits that are algebraically related to the message bits.

Convolutional Codes

- has memory
- discrete time convolution of the input sequence with the impulse response of the encoder.

Channel Coding Theorem

Channel Capacity $C = B\log_2(1+S/N)$ bit/sec

If a discrete memoryless channel has capacity C bit/s and a source generates information at a rate less than C bit/s, then there exists a coding technique such that the output of the source may be transmitted over the channel with an arbitrarily low probability of symbol error.

- This theorem gives a fundamental limit on the rate at which the transmission of reliable messages can take place over a discrete memoryless channel.

Binary operations

Channel encoding and decoding involves the binary arithmetic operations of modulo-2 addition and multiplication for binary codes {0,1}:

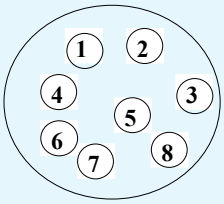
$0 + 0 = 0$	$0 \cdot 0 = 0$
$0 + 1 = 1$	$0 \cdot 1 = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$
$1 + 1 = 0$	$1 \cdot 1 = 1$

Modulo-2 addition is the Exclusive-OR (XOR) operation

Modulo-2 multiplication is the AND operation

Linear Block Code

Linear – the modulo-2 sum of any codewords is another valid codeword.



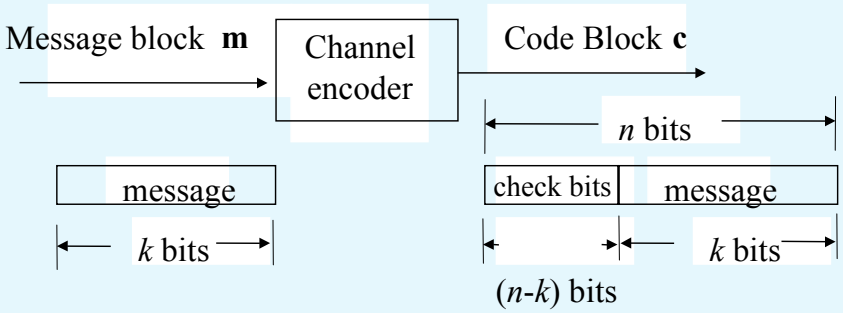
A code of 8 possible codewords

- 'Linear' means code 1 + code 2 = either code 1, 2, 3, 4, 5, 6, 7 or 8.
- Binary code is linear (e.g. 101+111 =010)

Block

Block

- the input message is divided into blocks of k bits and coded into n-bit codewords.



Message block **m**: $[m_0 \quad m_1 \quad \cdots \quad m_{k-1}]$

Code block **c**: $[c_0 \quad c_1 \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad c_{n-1}]$

$$= [b_0 \quad b_1 \quad \cdots \quad b_{n-k-1} \quad m_0 \quad m_1 \quad \cdots \quad m_{k-1}]$$

$$= [\mathbf{b} : \mathbf{m}]$$

This is known as a systematic (n, k) linear block code.

- A k-bit message is coded into a n-bit codeword consisted of (n - k) parity-check bits and k message bits.
- Code rate (or Rate efficiency) = k / n

Encoding: Coefficient matrix P and Generator matrix G

Encoding

Now, the n-bit code word is

$$c_0 = b_0$$
$$c_1 = b_1$$
$$\vdots$$
$$c_{\text{n-k-1}} = b_{n-k-1}$$
$$c_{\text{n-k}} = m_0$$
$$\vdots$$
$$c_{n-1} = m_{k-1}$$

The $(n - k)$ parity-check bits are linear sum of the k message bits, i.e., $b_i = p_{0i}m_0 + p_{1i}m_1 + \ldots + p_{k-1,i}m_{k-1}$

where $p_{ij} = \begin{cases} 1 & \text{if } b_i \text{ depends on } m_j \\ 0 & \text{otherwise} \end{cases}$

The coefficient p_{ij} are chosen in such a way that the rows of the generator matrix **G** are linearly independent.

In a matrix form, $\mathbf{b} = \mathbf{mP}$

where $\mathbf{P} = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0,n-k-1} \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$

and $\begin{aligned} \mathbf{c} &= [\mathbf{b} : \mathbf{m}] \\ &= [\mathbf{mP} : \mathbf{m}] \\ &= \mathbf{m}[\mathbf{P} : \mathbf{I_k}] \\ &= \mathbf{mG} \end{aligned}$

(**P** is called coefficient matrix)

$$\mathbf{c} = \mathbf{mG}$$

(1 x n) (1 x k) (k x n)

where

$$\mathbf{G} = \left[\begin{array}{cccc:cccccc} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ \cdot & \cdot & & \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & \cdot & \cdot & & & & \cdot \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{array}\right] \quad k \text{ rows}, \; n \text{ columns}$$

Example 1

Consider a (6, 3) linear block code

$$\mathbf{G} = [\mathbf{P} \vdots \mathbf{I}_3] = \left[\begin{array}{ccc:ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

$\mathbf{c} = \mathbf{mG}$ $\quad$ **m** - 8 possible 3-bit messages

For $\mathbf{m} = [1\ 1\ 1]$

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

Example 1

Similarly,

m	**c**
0 0 0	0 0 0 0 0 0
0 0 1	1 1 0 0 0 1
0 1 0	1 0 1 0 1 0
0 1 1	0 1 1 0 1 1
1 0 0	0 1 1 1 0 0
1 0 1	1 0 1 1 0 1
1 1 0	1 1 0 1 1 0
1 1 1	0 0 0 1 1 1

Encoding: needs to store **G** or **P**

For a *k*-bit message block, there are 2^k possible messages:

$$\begin{matrix}\mathbf{m}_0 \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_{2^k-1}\end{matrix} \quad \mathbf{c} = \mathbf{mG} \quad \begin{matrix}\mathbf{c}_0 \\ \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_{2^k-1}\end{matrix}$$

n bits and 2^k possible combinations
(there are 2^{n-k} - unused combinations)

k bits and 2^k possible combinations

Sum of codewords

To prove the sum of 2 codewords is another codeword, take codeword $\mathbf{c}_i + \mathbf{c}_j$,

$$\mathbf{c}_i + \mathbf{c}_j = \mathbf{m}_i\mathbf{G} + \mathbf{m}_j\mathbf{G}$$
$$= (\mathbf{m}_i + \mathbf{m}_j)\mathbf{G}$$

Since the mod-2 sum of $\mathbf{m}_i$ and $\mathbf{m}_j$ must result in another message vector within the set of 2^k message, say $\mathbf{m}_k$. So,

$$(\mathbf{m}_i + \mathbf{m}_j)\mathbf{G} = \mathbf{m}_k\mathbf{G} = \mathbf{c}_k$$

$$\therefore \mathbf{c}_i + \mathbf{c}_j = \mathbf{c}_k$$

sum of two codewords give another codeword

Decoding: Parity-check matrix H

Let **H** denote an (*n*-k) x *n* Parity-check matrix, defined as

$\mathbf{H} = [\mathbf{I}_{n-k} \vdots \mathbf{P}^T]$ $\mathbf{P}^T$ is a $(n - k)$ x k matrix

$$= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & \vdots & P_{00} & P_{10} & \cdots & P_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & \vdots & P_{01} & P_{11} & \cdots & P_{k-1,1} \\ 0 & 0 & 1 & \cdots & 0 & \vdots & P_{02} & P_{12} & \cdots & P_{k-1,2} \\ . & . & . & & & \vdots & . & . & & . \\ . & . & . & & & \vdots & . & . & & . \\ . & . & . & & & \vdots & . & . & & . \\ 0 & 0 & 0 & \cdots & 1 & \vdots & P_{0,n-k-1} & P_{1,n-k-1} & \cdots & P_{k-1,n-k-1} \end{bmatrix}$$

Property of H

Consider $\mathbf{HG}^{\mathbf{T}} = [\mathbf{I}_{\mathbf{n-k}} \vdots \mathbf{P}^{\mathbf{T}}] \begin{bmatrix} \mathbf{P}^{\mathbf{T}} \\ \hline \mathbf{I}_{\mathbf{k}} \end{bmatrix}$

$= \mathbf{P}^{\mathbf{T}} + \mathbf{P}^{\mathbf{T}} = [\mathbf{0}]$

Thus, $\mathbf{HG}^{\mathbf{T}} = 0 \quad (\Rightarrow \mathbf{GH}^{\mathbf{T}} = 0)$

Therefore, for any codeword **c**

$$\mathbf{cH}^{\mathbf{T}} = \mathbf{mGH}^{\mathbf{T}} = 0$$

and this equation is called parity-check equation

Example

e.g., $\mathbf{GH}^{\mathbf{T}} = \left[\begin{array}{ccc:ccc} 1 & 0 & 0 & P_{00} & P_{10} & P_{20} \\ 0 & 1 & 0 & P_{01} & P_{11} & P_{21} \\ 0 & 0 & 1 & P_{02} & P_{12} & P_{22} \end{array}\right] \begin{bmatrix} P_{00} & P_{10} & P_{20} \\ P_{01} & P_{11} & P_{21} \\ P_{02} & P_{12} & P_{22} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

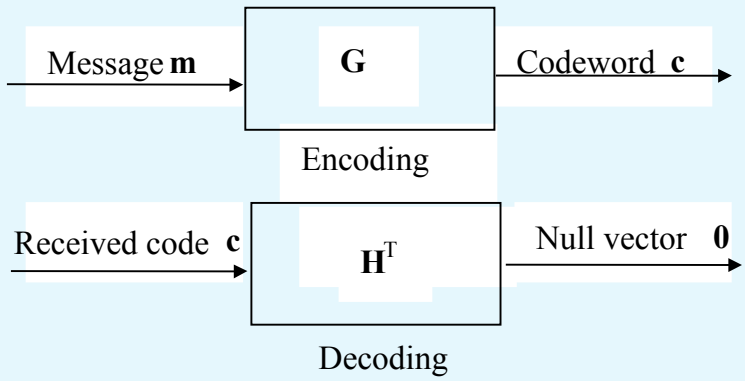
$$= \begin{bmatrix} P_{00}+P_{00} & P_{10}+P_{10} & P_{20}+P_{20} \\ P_{01}+P_{01} & P_{11}+P_{11} & P_{21}+P_{21} \\ P_{02}+P_{02} & P_{12}+P_{12} & P_{22}+P_{22} \end{bmatrix} = [0]$$

Summary

i. $\mathbf{c} = \mathbf{mG}$ Encoding

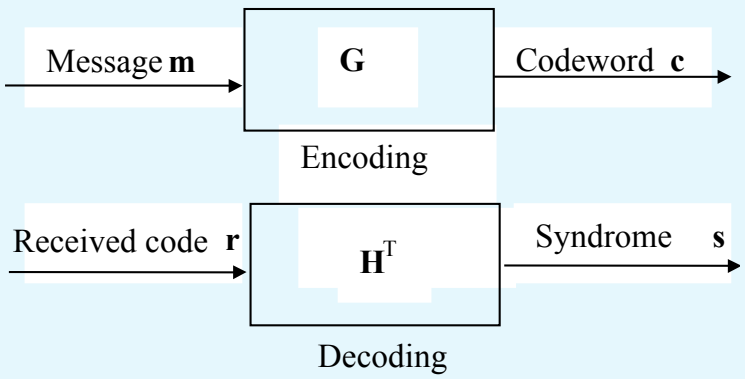
ii. $\mathbf{GH}^T = \mathbf{HG}^T = 0$

iii. $\mathbf{cH}^T = \mathbf{mGH}^T = 0$ Parity-check equation



Syndrome s

Let **r** be the received codeword that results from sending the codeword **c** overy a noisy channel.



Let $\mathbf{r} = \mathbf{c} + \mathbf{e}$ be the received code word

where $\mathbf{e} = \begin{bmatrix} e_0 & e_1 & e_2 & \cdots & e_{n-1} \end{bmatrix}$

$$e_i = \begin{cases} 1 & \text{if an error occurred at the } i^{th} \text{ location} \\ 0 & \text{otherwise} \end{cases}$$

Define the Syndrome $\mathbf{s}$ as $\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathbf{T}}$

$$= (\mathbf{c} + \mathbf{e})\mathbf{H}^{\mathbf{T}}$$
$$= \mathbf{c}\mathbf{H}^{\mathbf{T}} + \mathbf{e}\mathbf{H}^{\mathbf{T}}$$
$$= \mathbf{e}\mathbf{H}^{\mathbf{T}} \qquad \because \mathbf{c}\mathbf{H}^{\mathbf{T}} = 0$$

No error: $\mathbf{e} = [0 \quad 0 \quad 0 \quad \cdots \quad 0] \Rightarrow \mathbf{s} = [0]$ and $\mathbf{r}$ is a valid codeword

$\therefore \mathbf{c} = \mathbf{r}$

Error occurred: $\mathbf{e} \neq [0 \quad 0 \quad 0 \quad \cdots \quad 0] \Rightarrow \mathbf{s} \neq [0]$

Example 2

Consider a (7, 4) Linear Block Code

$$G = \left[\begin{array}{cccc:ccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$H = \left[\begin{array}{ccc:cccc} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}\right]$$

For $\mathbf{m} = [1\ 0\ 1\ 1]$, $\mathbf{c} = \mathbf{mG} = [0\ 0\ 1\ 1\ 0\ 1\ 1]$

No error, $\mathbf{r} = \mathbf{c}$, and $\mathbf{s} = \mathbf{c}\mathbf{H}^{\mathbf{T}} = [0\ 0\ 0]$

Let **c** suffered an error in the transmission such that the received vector $\mathbf{r} = [0\ 0\ \mathbf{0}\ 1\ 0\ 1\ 1]$

$= [0\ 0\ 1\ 1\ 0\ 1\ 1] + [0\ 0\ 1\ 0\ 0\ 0\ 0]$

$= \mathbf{c} + \mathbf{e}$

$$\mathbf{s} = \mathbf{cH}^{\mathbf{T}} = [0\ 0\ 0\ 1\ 0\ 1\ 1] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = [0\ 0\ 1] \quad (= \mathbf{eH}^{\mathbf{T}})$$

Note: Error in the 3rd bit results in **s** corresponding to the 3rd row of $\mathbf{H}^{\mathbf{T}}$.

For this (7, 4) code, a single error in the i^{th} bit would give **s** equal to the i^{th} row of $\mathbf{H}^{\mathbf{T}}$. The code is capable of correcting a single error.

Double errors: $\mathbf{e} = [0\ 1\ 1\ 0\ 0\ 0\ 0]$

$\mathbf{s} = \mathbf{c} = \mathbf{e} = [0\ 1\ 1] \neq [0]$

$= 7^{th}$ row of $\mathbf{H}^{\mathbf{T}}$ and is interpreted as an error in the 7^{th} bit of the received word.

- Error is detectable but not correctable.
- More than 2 errors might give $\mathbf{s} = [0]$ and in that case the errors are not detectable!

Syndrome decoding for multiple errors: Problem

In syndrome decoding, the receiver forms $\mathbf{s} = \mathbf{rH}^T$.

$\mathbf{s} = \mathbf{rH}^T = 0$ implies transmitted codeword $\mathbf{c} = \mathbf{r}$

$\mathbf{s} = \mathbf{rH}^T \neq 0$ Find the corresponding error pattern(s) that gives rise to $\mathbf{s}$ and $\mathbf{c} = \mathbf{r} + \mathbf{e}$

The process is a bit more complicated as was illustrated for a double error in the (7, 4) code.

Consider the (7, 4) code with

$$\mathbf{G} = \left[\begin{array}{ccc:cccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right]$$

For $\mathbf{r} = [1\ 0\ 0\ 0\ 0\ 1\ 1]$

$\mathbf{s} = \mathbf{rH}^T = [1\ 0\ 0]$

But $\mathbf{s} = \mathbf{rH}^T = \mathbf{eH}^T$, let $\mathbf{e} = [e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6 \quad e_7]$

$$\mathbf{s} = [1\,0\,0] = [e_1\ e_2\ e_3\ e_4\ e_5\ e_6\ e_7] \begin{bmatrix} 1&1&1 \\ 1&1&0 \\ 1&0&1 \\ 0&1&1 \\ 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{bmatrix}$$

$$\Rightarrow \quad 1 = e_1 + e_2 + e_3 + e_5$$
$$0 = e_1 + e_2 + e_4 + e_6$$
$$0 = e_1 + e_3 + e_4 + e_7$$

3 equations and 7 unknowns. $\therefore$ Unique solution for **e** is not possible.

For **s** = [1 0 0], the possible error patterns are

2^4

e_1	e_2	e_3	e_4	e_5	e_6	e_7
0	0	0	0	1	0	0
1	0	0	1	0	0	0
1	1	1	0	0	0	0
0	1	1	0	1	1	1
⋮	⋮					

Error detection and error correction capabilities

Hamming weight of a code word **c**, $w(\mathbf{c})$, is the number of nonzero elements in the code vector. For example,

$\mathbf{c} = [11001010111] \Rightarrow w(\mathbf{c}) = 7$

Hamming distance, $d(\mathbf{c_1},\mathbf{c_2})$, between a pair of code vectors and that have the same number of elements is defined as the number of location in which their respective elements differ. Say,

$\mathbf{c_1}$=[1011001] , $\mathbf{c_2}$ =[0111101] $\Rightarrow d(\mathbf{c_1}, \mathbf{c_2}) = 3$

Minimum distance

Minimum distance, d_{min}, of a block code is the smallest distance between any pair of code words in the code. For example,

Codewords	weight **w(c)**		
000000	0		$d(\mathbf{c}_1,\mathbf{c}_2)= 3$
110001	3		$d(\mathbf{c}_1,\mathbf{c}_2) = 3$
101010	3		$d(\mathbf{c}_1,\mathbf{c}_2) = 4$
011011	4		:
011100	3		:
101101	4	Look for	:
110110	4	the smallest distance $d(\mathbf{c}_1,\mathbf{c}_2)$	
000111	3		

Hamming weight and Minimum distance

Theorem

Minimum distance of a linear block code is the smallest Hamming weight of the nonzero code vectors in the code.

The smallest Hamming weight from the previous example except **c**= [0 0 0 0 0 0 0] is 3, therefore, $d_{min}= 3$

$$d_{\min} = \min\{d(\mathbf{c}_i, \mathbf{c}_j)\} = \min\{w(\mathbf{c}_i)\}$$

Proof

- .The minimum distance is the same as the smallest Hamming weight of the difference between any pair of code vector (i.e. $\min\{d(\mathbf{c}_i, \mathbf{c}_j)\} = \min\{w(\mathbf{c}_i - \mathbf{c}_j)\}$)
- From the closure property of linear block code, the sum (or difference) of two code vectors is another code vector.(i.e. $\min\{w(\mathbf{c}_i - \mathbf{c}_j)\} = \min\{w(\mathbf{c}_i)\}$)

Note: The best strategy for the decoder is to pick the code vector closest to the received vector, that is, the one for which the Hamming distance $d(\mathbf{c_i}, \mathbf{r})$ is the smallest.

Theorem

A linear block code with a minimum distance $d_{\min}$ can correct up to $[(d_{\min} - 1) / 2]$ errors and detect up to $(d_{\min} - 1)$ errors in each code word where [*] denotes the largest integer no greater than *, i.e., [3.5] = 3. Thus, in the previous example,

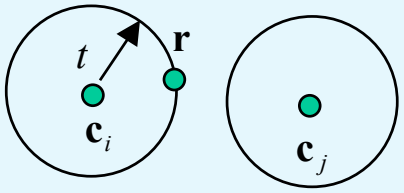
$d_{\min} = 3 \;\; \therefore \;\; [(3 - 1) / 2] = 1$

$(3 - 1) = 2$

Therefore, this (7, 4) block code can correct 1 error and detect up to 2 errors.

Interpretation:

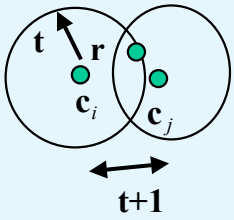
Case I $d(\mathbf{c}_i, \mathbf{c}_j) \geq 2t + 1$



error can be corrected

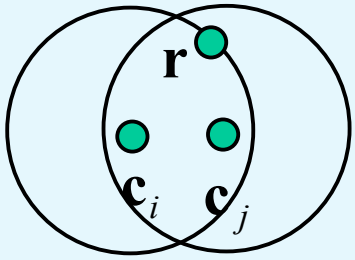
(*t* is the distance between received codeword **r** and the transmitted codeword **c**)

Case II $d(\mathbf{c}_i, \mathbf{c}_j) \geq t + 1$



error cannot be corrected but can be detected

Case III $d(\mathbf{c}_i, \mathbf{c}_j) < t+1$



error cannot be detected

Syndrome decoding for multiple errors: Solution

Consider a (n, k) code

There are 2^k possible codewords: $\mathbf{c_1}$, $\mathbf{c_2}$, $\mathbf{c_3}$,…, $\mathbf{c_{2k}}$

Received vector **r** has 2^n possible values. Decoder has to relate **r** with one of the 2^k valid code words.

Divide the 2^n possible received codewords into 2^k disjoint subsets in such a way that each subset contains only 1 code vector $\mathbf{c_i}$. Then the received codeword **r** is decoded into $\mathbf{c_i}$ if it is in the ^{i}th subset.

Construction of subsets: standard array

The subsets constitute is constructed as follows:

1. The 2^k code words are placed in a row with the all-zero codeword $\mathbf{c_1}$ as the left most element.
2. An error pattern $\mathbf{e_2}$ is picked and placed under $\mathbf{c_1}$, and a second row is formed by adding $\mathbf{e_2}$ to each of the remaining code vectors in the first row; it is important that the error pattern chosen as the first element in a row has not previously appeared in the standard array.
3. Step 2 is repeated until all the possible error patterns have been accounted for.

The array is called standard array.

$\mathbf{c_1}$	$\mathbf{c}_2$	$\mathbf{c}_3$	...	$\mathbf{c}_{2^k}$
$\mathbf{e_2}$	$\mathbf{c}_2 + \mathbf{e_2}$	$\mathbf{c}_3 + \mathbf{e_2}$	...	$\mathbf{c}_{2^k} + \mathbf{e_2}$
$\mathbf{e}_3$	$\mathbf{c}_2 + \mathbf{e}_3$	$\mathbf{c}_3 + \mathbf{e}_3$	...	$\mathbf{c}_{2^k} + \mathbf{e}_3$
$\vdots$	$\vdots$	$\vdots$	...	$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{c}_2 + \mathbf{e}_{2^{n-k}}$	$\mathbf{c}_3 + \mathbf{e}_{2^{n-k}}$	...	$\mathbf{c}_{2^k} + \mathbf{e}_{2^{n-k}}$

The rows of the standard array are called co-sets and the first element in each row is called co-set leader. **The co-set leaders are chosen to be the error patterns that are most likely to occur**. In the case of a binary symmetric channel, the smallest the Hamming weight of an error pattern the more likely it is to occur. Accordingly, the standard array should be constructed with each coset leader having the minimum Hamming weight in its coset.

Thus, the procedure for decoding a linear block code is

1. Compute $\mathbf{s} = \mathbf{r}\mathbf{H}^T$.

2. Within the coset characterized by the syndrome **s**, identify the coset leader (i.e., the error pattern with the largest probability of occurrence); call it $\mathbf{e}_o$.

3. Compute $\mathbf{c}=\mathbf{r}+\mathbf{e}_o$.

Example 3

Consider the (6, 3) linear code generated by the following matrix: $\mathbf{G} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$ .The standard array of this code is

coset leader							
000000	011100	101010	110001	110110	101101	011011	000111
100000	111100	001010	010001	010110	001101	111011	100111
010000	001100	111010	100001	100110	111101	001011	010111
001000	010100	100010	111001	111110	100101	010011	001111
000100	011000	101110	110101	110010	101001	011111	000011
000010	011110	101000	110011	110100	101111	011001	000101
000001	011101	101011	110000	110111	101100	011010	000110
100100	111000	001110	010101	010010	001001	111111	100011

Example 4

Construction of a single-error correction code

need $d_{min} = 3$ (or more)

when single error occurs, the i^{th} bit corresponds to the i^{th} row of $\mathbf{H^T}$

Therefore,

i. Choose the rows of $\mathbf{H^T}$ to be all distinct.

ii. Make sure that the first $(n\text{-}k)$ rows of $\mathbf{H^T}$ is $\mathbf{I_{n-k}}$.

iii. Do not use [000…0] for any row of $\mathbf{H^T}$.

$$\mathbf{H}^{\mathbf{T}} = \begin{bmatrix} I_{n-k} \\ P \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ P_{00} & P_{01} & P_{02} & \cdots & P_{0,n-k-1} \\ P_{10} & P_{11} & P_{12} & \cdots & P_{1,n-k-1} \\ \vdots & & \vdots & & \\ P_{k-1,0} & P_{k-1,1} & P_{k-12} & \cdots & P_{k-1,n-k-1} \end{bmatrix}}_{n-k} \begin{matrix} \} n-k \\ \} k \end{matrix} \} n$$

Since there are $(n\text{-}k)$ columns, we can have a possible of $2^{n\text{-}k}$ distinct rows. So, use the first rows for , then fill in the rest of the matrix arbitrarily but make sure that all the rows are different. Finally, make sure that none of the rows is [0 0 0 . . . 0].

Question: Given k, how can n be determined?

Now, n rows of $\mathbf{H^T}$ must be distinct.

$$\Rightarrow 2^{n-k} - 1 \geq n$$
$$(n-k) \geq \log_2(n+1)$$
$$\underline{n \geq k + \log_2(n+1)}$$

$\therefore$ Given k, the minimum value of n is $k + \log_2(n+1)$.

Design a (n, 8) code with single error correction capability,

$$n \geq 8 + \log_2(n+1)$$

Minimum value of n that satisfies the equation is 12. Hence a (12,8) code.

$$\therefore H^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} I_{n-k} \\ \cdots \\ P \end{bmatrix}$$

E.47

Check: For the 8-bit data **m** = [1 1 1 0 1 0 1 1], the 12-bit code word **c** is

$$\mathbf{c} = \mathbf{mG} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \left[\begin{array}{cccc:cccccccc} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

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

E.48

Assume that the received bit is in error, i.e.,

$\mathbf{r}$ = [1 0 1 0 **0** 1 1 0 1 0 1 1]

$$\mathbf{s} = \mathbf{r}\mathbf{H}^T = [101001101011] \begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \\ \cdots\cdots \\ 1100 \\ 0110 \\ 0011 \\ 1001 \\ 1010 \\ 0101 \\ 1110 \\ 0111 \end{bmatrix} = [1100]$$

$\mathbf{s}$ = [1 1 0 0] corresponds to the 5th row of $\mathbf{H}^T$

$\therefore$ the bit of $\mathbf{r}$ is in error and

$\mathbf{c} = \mathbf{r}+\mathbf{e}$ = [101001101011]+[000010000000]

= [101011101011] and the message $\mathbf{m}$ is

[11101011]