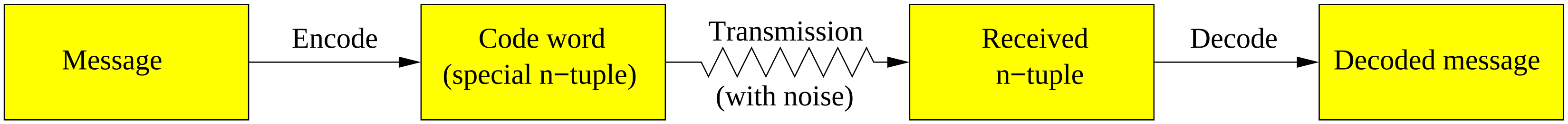
Elementary Coding Theory

Information Transmission



- The message is a binary string (m-tuple)

- The code word is also a binary string (n-tuple)

Errors

- Error - change in some of the bits in the code word
- Single error - change in only one bit of a code word

Easy Error Detection–Parity

- Add a parity check bit
- Message word + check bit = code word

Message Word (m-tuple)	Even Parity Check (n-tuple)	Odd Parity Check (n-tuple)
000 000	000 000 0	000 000 1
110 000	110 000 0	110 000 1
110 111	110 111 1	110 111 0

Single Error Detection

- Adding a parity check bit allows the detection of all single errors
- All single errors result in an error indication

Received 7-tuple	Decoded Word
001 000 1	001 000
101 010 0	Parity error
111 111 0	111 111
111 111 1	Parity error

Parity

- Even (or odd) parity checking is sufficient for most computer purposes
- Limitations:
  - Cannot detect some multiple errors
  - Cannot correct any errors

`110 010 1` Code word

`111 000 1` Code word

Maximum Likelyhood Decoding

- Assume transmission errors:
  - are rare
  - occur independently in each bit
- Therefore, 2 errors occur less frequently than 1, 3 errors occur less frequently than 2, etc.
- Maximum likelyhood decoding
  - Look for code word that was most likely transmitted

Simplest Error Correcting Code

The messages are either 0 or 1

Message	Code Word
0	000
1	111

Difference Matrix shows the number of bits a given 3-tuple is different from a code word

Code Word	000	001	010	011	100	101	110	111
000	0	1	1	2	1	2	2	3
111	3	2	2	1	2	1	1	0

Simplest Error Correcting Code (cont.)

- Encoding

Message	Code Word
`0`	`000`
`1`	`111`

- For single error correction, select closest code word from difference matrix

Enhanced Error Detection

Encoding

Message	Code Word
0	000
1	111

Alternatively, can detect up to two errors

But error correction then becomes impossible

111 $\rightsquigarrow$ 100 (transmission error)

Don't know if 111 or 000 was transmitted

Fundamental Principle of Coding Theory

The ability of a code to detect or to correct errors depends solely on its set of code words

Suppose 1100 and 0100 are code words in some code

1100 $\rightsquigarrow$ Error in bit one $\rightsquigarrow$ 0100

Received code would be decoded as an erroneous message

Suppose 1100 and 0101 are code words in some code

1100 $\rightsquigarrow$ Error in any single bit $\rightsquigarrow$ Can never be 0101

Hamming Distance

Let a and b be binary n-tuples. The number of places in which a and b differ is called the *Hamming distance* between a and b. The Hamming distance between tuples of different length is undefined.

$H(a,a) = 0$

If $H(a,b) = 0$, then $a = b$

Metric Properties

$a, b, c \in N_2^n$ (binary n-tuples)

- $H(a,b) \geq 0$
- $H(a,b) = 0 \leftrightarrow a = b$
- $H(a,b) = H(b,a)$
- $H(a,c) \leq H(a,b) + H(b,c)$

Minimum Distance

Consider a code whose code words are in N_2^n. The minimum distance, d, for the code is the minimum of Hamming distances $H(a,b)$ where a and b are distinct code words.

If $d = 1$, then the code cannot detect all transmission errors.

If $d = 2$, then the code can detect but not correct all single errors.

If $d \geq 3$, then the maximum likelyhood decoding scheme can correct all single errors.

Error Correcting Example

c_1 = 00000, c_2 = 01110, c_3 = 10111, c_4 = 11001

$d = ?$

Received 5-tuple = 11111 = r

c_i	$H(r, c_i)$
00000	5
01110	2
10111	1
11001	2

c_3 is the unique code word with minimum distance.

Group

A *group* is a mathematical structure consisting of a set and an operation, $[A, \cdot]$ with the following properties:

- For all $a, b \in A$, $a \cdot b \in A$ (closure)
- For all $a, b, c \in A$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ (associativity)
- There exists $e \in A$ such that for all $x \in A, e \cdot x = x = x \cdot e$ (identity)
- For all $x \in A$ there exists $y \in A$ such that $x \cdot y = e = y \cdot x$ (invertibility)

Group Codes

Group codes facilitate the construction of error correcting codes.

A code whose code words are binary n-tuples is a group code if the sum in N_2^n of any two code words is again a code word.

The addition is a component-wise mod 2 addition

$+_2$	0	1
0	0	1
1	1	0

If c is a code word, then $c + c = \mathbf{0}$ (where $\mathbf{0}$ is the element of N_2^n consisting of all zeros.

Weight

The *weight* of a binary n-tuple a is the number of 1s in the n-tuple.

$W(1101) = 3$, $W(10001) = 2$, $W(111) = 3$, $W(00000) = 0$

$W(a) = H(a, 0)$

$H(a, b) = W(a + b)$

Let d be the minimum distance for a group code. Then d also equals the minimum of the weights of all code words except $\mathbf{0}$.

Multiplication Mod 2

$\cdot_2$	0	1
0	0	0
1	0	1

Parity Check Matrices

Let H be an $n \times r$ binary matrix.

Suppose that the code words for a code consist of all binary n-tuples c such that $c \cdot H = \mathbf{0}_r$. $c \in N_2^n$, and $\mathbf{0}_r \in N_2^r$.

Parity Check Matrices

Example:

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

3-tuple	$c \cdot H$	Code word?
000	00	yes
001	01	no
010	10	no
011	11	no
100	11	no
101	10	no
110	01	no
111	00	yes

$d = 3$—single error correcting or double error detecting

Parity Check Matrices (cont.)

Another example:

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

3-tuple	$c \cdot H$	Code word?
000	0	yes
001	1	no
010	0	yes
011	1	no
100	1	no
101	0	yes
110	1	no
111	0	yes

$d = 1$—not even single error detecting!

Group Homomorphism

Let $[N_2^n, +_2]$ be a group.

Let $[N_2^n, +_2] \to [N_2^r, +_2]$ be a homomorphism. (This homomorphism f maps wider bitstrings to narrower bitstrings.)

$\ker f$ is the set of elements in $[N_2^n, +_2]$ that map to to $\mathbf{0}_r$ under f.

$\ker f$ includes $\mathbf{0}_n$, all of its elements are invertible, it is closed, and associatively obviously still holds; therefore, $\ker f$ is the set of code words in some group code.

Canonical Parity Check Matrix

If in H the last r rows form the $r \times r$ identity matrix, then H is a *canonical parity check matrix*.

$$\begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \cdot c_1 + 0 \cdot c_2 + 1 \cdot c_3 \end{bmatrix} = \begin{bmatrix} c_1 + c_3 \end{bmatrix} = [0]$$

This means the number of 1s in the first and third places is even; thus, an even parity check is performed on bits 1 and 3.

Another Example

$$\begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} (c_1 + c_2) & (c_1 + c_3) \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

This means an even parity check is being performed on bits 1 and 2, and an even parity check is being performed on bits 1 and 3.

Minimum Code Weight

The minimum weight of the code = the minimum number of rows in H that add to $\mathbf{0}_r$.

Hamming Codes

To generate a single error correcting code for $N_2^m = N_2^{n-r}$ (a subgroup of N_2^n):

- The dimension of H is $n \times r$
- no two rows of H can be the same (add to $\mathbf{0}_r$)
- each row in H has r elements
- there can be no more than 2^r rows
- no row can contain $\mathbf{0}_r$, so number of rows $\leq 2^r - 1$
- $n \leq 2^r - 1$
- $m = n - r \leq 2^r - r - 1$

A Hamming code is *perfect* if $m = 2^r - r - 1$.

Hamming Code Example

$$m = 2$$

$$n = 5$$

$$r = 3$$

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

Hamming Code Example (cont.)

$$\begin{bmatrix} c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} (c_1 + c_2 + c_3) & (c_2 + c_4) & (c_1 + c_2 + c_5) \end{bmatrix}$$

c_1	c_2	c_3	c_4	c_5
0	0	0	0	0
0	1	1	1	1
1	0	1	0	1
1	1	0	1	0

$$C = \{00000, 01111, 10101, 11010\}$$

Decoding

Let c be the received code word.

- If $c \cdot H = \mathbf{0}_r$, then strip off the r check bits and interpret the m message bits as the original message.

- If $c \cdot H \neq \mathbf{0}_r$, then at least one of the bits is non-zero. Find the row in H that matches the received bogus code word. The number of the matching row indicates the bit position of the error in the received code word.

Decoding Example

Received Code Word = `01111`

Decoding:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Interpretation: Message was `01`

Decoding Example 2

Received Code Word = `01101`

Decoding:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

Interpretation: Non-zero result: `010` which matches row 4 in H; therefore, error is in bit 4, the code word should have been `01111`, and message was `01`