

## 15-853: Algorithms in the Real World

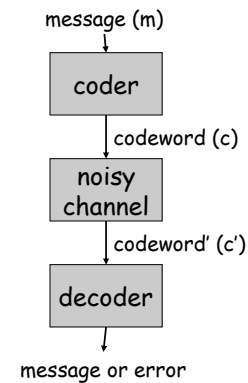
### Error Correcting Codes I

- Overview
- Hamming Codes
- Linear Codes

15-853

Page1

## General Model



Errors introduced by the noisy channel:

- changed fields in the codeword (e.g. a flipped bit)
- missing fields in the codeword (e.g. a lost byte). Called **erasures**

How the decoder deals with errors.

- **error detection** vs.
- **error correction**

15-853

Page2

## Applications

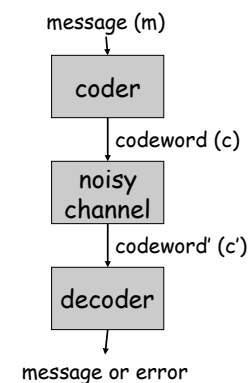
- **Storage:** CDs, DVDs, "hard drives",
- **Wireless:** Cell phones, wireless links, wimax (3G, 4G)
- **Satellite and Space:** TV, Mars rover, ...
- **Digital Television:** DVD, MPEG2 layover, DVB
- **Wired:** 10Gbase-T ethernet, ..

**Reed-Solomon** codes are the most used in practice, including most examples mentioned above, but **LDPC codes** are now becoming more widely used. Algorithms for decoding are quite sophisticated.

15-853

Page3

## Block Codes



Each message and codeword is of fixed size

$\Sigma$  = codeword alphabet

$k = |m|$   $n = |c|$   $q = |\Sigma|$

$C \subseteq \Sigma^n$  (codewords)

$\Delta(x, y)$  = number of positions s.t.  $x_i \neq y_i$

$d = \min\{\Delta(x, y) : x, y \in C, x \neq y\}$

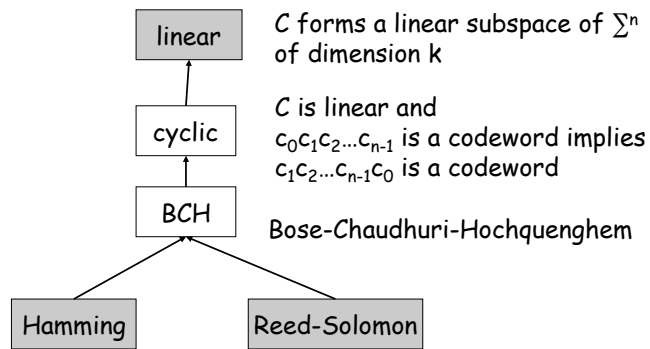
$s = \max\{\Delta(c, c')\}$  that the code can correct

Code described as:  $(n, k, d)_q$

15-853

Page4

## Hierarchy of Codes



These are all block codes.

15-853

Page5

## Binary Codes

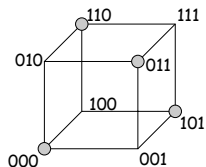
Today we will mostly be considering  $\Sigma = \{0,1\}$  and will sometimes use  $(n,k,d)$  as shorthand for  $(n,k,d)_2$ . In binary  $\Delta(x,y)$  is often called the **Hamming distance**.

15-853

Page6

## Hypercube Interpretation

Consider codewords as vertices on a hypercube.



- codeword
- $d = 2 = \text{min distance}$
- $n = 3 = \text{dimensionality}$
- $2^n = 8 = \text{number of nodes}$

The distance between nodes on the hypercube is the Hamming distance  $\Delta$ . The minimum distance is  $d$ .

001 is equidistance from 000, 011 and 101.

For  $s$ -bit error detection  $d \geq s + 1$

For  $s$ -bit error correction  $d \geq 2s + 1$

15-853

Page7

## Error Detection with Parity Bit

A  $(k+1,k,2)_2$  systematic code

**Encoding:**

$$m_1m_2\dots m_k \Rightarrow m_1m_2\dots m_kp_{k+1}$$

$$\text{where } p_{k+1} = m_1 \oplus m_2 \oplus \dots \oplus m_k$$

$d = 2$  since the parity is always even (it takes two bit changes to go from one codeword to another).

**Detects one-bit error** since this gives odd parity

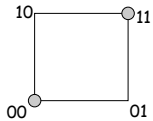
**Cannot be used to correct 1-bit error** since any odd-parity word is equal distance  $\Delta$  to  $k+1$  valid codewords.

15-853

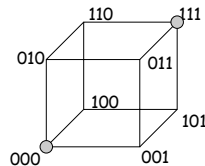
Page8

## Error Correcting One Bit Messages

How many bits do we need to correct a one bit error on a one bit message?



2 bits  
0 → 00, 1 → 11  
( $n=2, k=1, d=2$ )



3 bits  
0 → 000, 1 → 111  
( $n=3, k=1, d=3$ )

In general need  $d \geq 3$  to correct one error. Why?

15-853

Page9

## Example of $(6,3,3)_2$ systematic code

message	codeword
000	000000
001	001011
010	010101
011	011110
100	100110
101	101101
110	110011
111	111000

**Definition:** A **Systematic code** is one in which the message appears in the codeword

15-853

Page10

## Error Correcting Multibit Messages

We will first discuss **Hamming Codes**

Detect and correct 1-bit errors.

Codes are of form:  $(2^r-1, 2^r-1-r, 3)$  for any  $r > 1$

e.g.  $(3,1,3)$ ,  $(7,4,3)$ ,  $(15,11,3)$ ,  $(31, 26, 3)$ , ...

which correspond to 2, 3, 4, 5, ... "parity bits" (i.e.  $n-k$ )

The high-level idea is to "localize" the error.

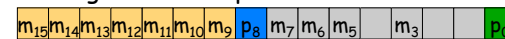
Any specific ideas?

15-853

Page11

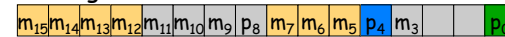
## Hamming Codes: Encoding

Localizing error to top or bottom half 1xxx or 0xxx



$$p_8 = m_{15} \oplus m_{14} \oplus m_{13} \oplus m_{12} \oplus m_{11} \oplus m_{10} \oplus m_9$$

Localizing error to x1xx or x0xx



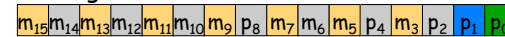
$$p_4 = m_{15} \oplus m_{14} \oplus m_{13} \oplus m_{12} \oplus m_7 \oplus m_6 \oplus m_5$$

Localizing error to xx1x or xx0x



$$p_2 = m_{15} \oplus m_{14} \oplus m_{11} \oplus m_{10} \oplus m_7 \oplus m_6 \oplus m_3$$

Localizing error to xxx1 or xxx0

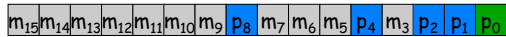


$$p_1 = m_{15} \oplus m_{13} \oplus m_{11} \oplus m_9 \oplus m_7 \oplus m_5 \oplus m_3$$

15-853

Page12

## Hamming Codes: Decoding



We don't need  $p_0$ , so we have a (15,11,?) code.

After transmission, we generate

$$b_8 = p_8 \oplus m_{15} \oplus m_{14} \oplus m_{13} \oplus m_{12} \oplus m_{11} \oplus m_{10} \oplus m_9$$

$$b_4 = p_4 \oplus m_{15} \oplus m_{14} \oplus m_{13} \oplus m_{12} \oplus m_7 \oplus m_6 \oplus m_5$$

$$b_2 = p_2 \oplus m_{15} \oplus m_{14} \oplus m_{11} \oplus m_{10} \oplus m_7 \oplus m_6 \oplus m_3$$

$$b_1 = p_1 \oplus m_{15} \oplus m_{13} \oplus m_{11} \oplus m_9 \oplus m_7 \oplus m_5 \oplus m_3$$

With no errors, these will all be zero

With one error  $b_8 b_4 b_2 b_1$  gives us the error location.

e.g. **0100** would tell us that  $p_4$  is wrong, and

**1100** would tell us that  $m_{12}$  is wrong

## Hamming Codes

Can be generalized to any power of 2

- $n = 2^r - 1$  (15 in the example)
- $(n-k) = r$  (4 in the example)
- $d = 3$  (discuss later)
- Can correct one error, but can't tell difference between one and two!
- Gives  $(2^r-1, 2^r-1-r, 3)$  code

Extended Hamming code

- Add back the parity bit at the end
- Gives  $(2^r, 2^r-1-r, 4)$  code
- Can correct one error and detect 3
- (not so obvious)

## Lower bound on parity bits

How many nodes in hypercube do we need so that  $d = 3$ ?

Each of the  $2^k$  codewords eliminates  $n$  neighbors plus itself, i.e.  $n+1$

$$2^n \geq (n+1)2^k$$

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

$$n \geq k + \lceil \log_2(n+1) \rceil$$

In previous hamming code  $15 \geq 11 + \lceil \log_2(15+1) \rceil = 15$

Hamming Codes are called **perfect codes** since they match the lower bound exactly

## Lower bound on parity bits

What about fixing 2 errors (i.e.  $d=5$ )?

Each of the  $2^k$  codewords eliminates itself, its neighbors and its neighbors' neighbors, giving:  $1 + \binom{n}{1} + \binom{n}{2}$

$$2^n \geq (1 + n + n(n-1)/2)2^k$$

$$n \geq k + \log_2(1 + n + n(n-1)/2)$$

$$\geq k + 2\log_2 n - 1$$

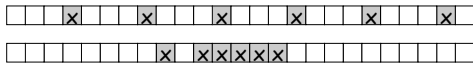
Generally to correct  $s$  errors:

$$n \geq k + \log_2(1 + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{s})$$

## Lower Bounds: a side note

The lower bounds assume adversarial placement of bit errors.

In practice errors are likely to be less than random, e.g. evenly spaced or clustered:



Can we do better if we assume **regular errors**?

We will come back to this later when we talk about **Reed-Solomon** codes. In fact, this is a major reason why Reed-Solomon codes are used much more than Hamming-codes.

## Linear Codes

If  $\Sigma$  is a field, then  $\Sigma^n$  is a vector space

**Definition:**  $C$  is a linear code if it is a linear subspace of  $\Sigma^n$  of dimension  $k$ .

This means that there is a set of  $k$  independent vectors  $v_i \in \Sigma^n$  ( $1 \leq i \leq k$ ) that span the subspace.

i.e. every codeword can be written as:

$$c = a_1 v_1 + \dots + a_k v_k \quad a_i \in \Sigma$$

The sum of two codewords is a codeword.

## Linear Codes

Vectors for the  $(7,4,3)_2$  Hamming code:

	$m_7$	$m_6$	$m_5$	$p_4$	$m_3$	$p_2$	$p_1$
$v_1 =$	1	0	0	1	0	1	1
$v_2 =$	0	1	0	1	0	1	0
$v_3 =$	0	0	1	1	0	0	1
$v_4 =$	0	0	0	0	1	1	1

How can we see that  $d \leq 3$ ?

## Generator and Parity Check Matrices

**Generator Matrix:**

A  $k \times n$  matrix  $G$  such that:  $C = \{xG \mid x \in \Sigma^k\}$

Made from stacking the spanning vectors

**Parity Check Matrix:**

An  $(n - k) \times n$  matrix  $H$  such that:  $C = \{y \in \Sigma^n \mid Hy^T = 0\}$

Codewords are the nullspace of  $H$

These **always exist for linear codes**

## Advantages of Linear Codes

- Encoding is efficient (vector-matrix multiply)
- Error detection is efficient (vector-matrix multiply)
- **Syndrome** ( $Hy^T$ ) has error information
- Gives  $q^{n-k}$  sized table for decoding  
Useful if  $n-k$  is small

15-853

Page21

## Example and "Standard Form"

For the Hamming (7,4,3) code:

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

By swapping columns 4 and 5 it is in the form  $[I_k, A]$ .  
A code with a matrix in this form is **systematic**, and  
 $G$  is in "**standard form**"

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

15-853

Page22

## Relationship of $G$ and $H$

If  $G$  is in standard form  $[I_k, A]$   
then  $H = [A^T, I_{n-k}]$

**Example** of (7,4,3) Hamming code:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \xrightarrow{\text{transpose}} H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

15-853

Page23

## Proof that $H$ is a Parity Check Matrix

Suppose that  $x$  is a message. Then

$$H(xG)^T = H(G^T x^T) = (HG^T)x^T = (A^T I_k + I_{n-k} A^T)x^T = (A^T + A^T)x^T = 0$$

Now suppose that  $Hy^T = 0$ . Then  $A^T_{i,*} \cdot y^T_{[1..k]} + y^T_{k+i} = 0$  (where  $A^T_{i,*}$  is row  $i$  of  $A^T$  and  $y^T_{[1..k]}$  are the first  $k$  elements of  $y^T$ ) for  $1 \leq i \leq n-k$ . Thus,  $y_{[1..k]} \cdot A_{*,i} = y_{k+i}$  where  $A_{*,i}$  is now column  $i$  of  $A$ , and  $y_{[1..k]}$  are the first  $k$  elements of  $y$ , so  $y_{[k+1..n]} = y_{[1..k]}A$ .

Consider  $x = y_{[1..k]}$ . Then  $xG = [y_{[1..k]} \mid y_{[1..k]}A] = y$ .

Hence if  $Hy^T = 0$ ,  $y$  is the codeword for  $x = y_{[1..k]}$ .

15-853

Page24

## The d of linear codes

**Theorem:** Linear codes have distance  $d$  if every set of  $(d-1)$  columns of  $H$  are linearly independent, but there is a set of  $d$  columns that are linearly dependent.

**Proof:** if  $d-1$  or fewer columns are linearly dependent, then for any codeword  $y$ , there is another codeword  $y'$ , in which the bits in the positions corresponding to the columns are inverted, that both have the same syndrome (0).

If every set of  $d-1$  columns is linearly independent, then changing any  $d-1$  bits in a codeword  $y$  must also change the syndrome (since the  $d-1$  corresponding columns cannot sum to 0).

15-853

Page25

## Dual Codes

For every code with

$$G = I_k, A \quad \text{and} \quad H = A^T, I_{n-k}$$

we have a **dual code** with

$$G = I_{n-k}, A^T \quad \text{and} \quad H = A, I_k$$

The dual of the Hamming codes are the **binary simplex codes**:  $(2^{r-1}, r, 2^{r-1}-r)$

The dual of the extended Hamming codes are the **first-order Reed-Muller codes**.

Note that these codes are **highly redundant** and can fix many errors.

15-853

Page26

## NASA Mariner:

Deep space probes from  
1969-1977.

Mariner 10 shown



Used  $(32,6,16)$  Reed Muller code ( $r = 5$ )

Rate =  $6/32 = .1875$  (only 1 out of 5 bits are useful)

Can fix up to 7 bit errors per 32-bit word

15-853

Page27

## How to find the error locations

$Hy^T$  is called the **syndrome** (no error if 0).

In **general** we can find the error location by creating a table that maps each syndrome to a set of error locations.

**Theorem:** assuming  $s \leq 2d-1$  every syndrome value corresponds to a unique set of error locations.

**Proof: Exercise.**

Table has  $q^{n-k}$  entries, each of size at most  $n$  (i.e. keep a bit vector of locations).

15-853

Page28