15-853:Algorithms in the Real World

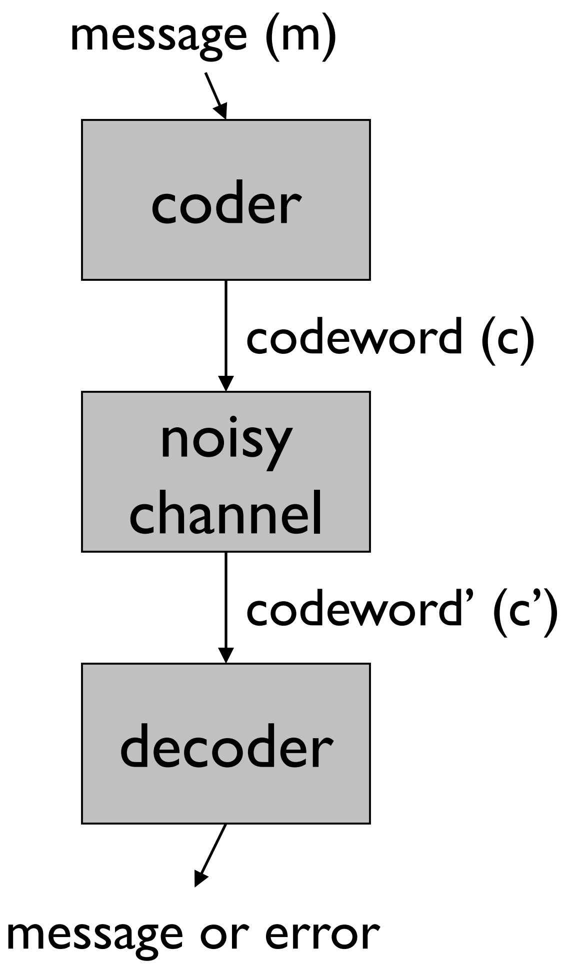
Error Correcting Codes (cont..)

Scribe volunteers: ?

Announcement:

Scribe notes sign up, template and instructions on the course webpage

Recap: Block Codes



Each message and codeword is of fixed size

Σ = codeword alphabet

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

C = "code" = set of codewords

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

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

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

Code described as: $\mathbf{(n,k,d)_q}$

Recap: Role of Minimum Distance

Theorem:

A code C with minimum distance "d" can:

1. detect any (d-1) errors
2. recover any (d-1) erasures
3. correct any <write> errors

Stated another way:

For s-bit error detection or erasure recovery: $d \geq s + 1$

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

To correct a erasures and b errors:

$$d \geq a + 2b + 1$$

Clarification

- Error model:
  1. Arbitrary/adversarial errors
     - Error can occur in “any” s code symbols
  2. Symmetric across alphabet values
- Role of minimum distance decoding
  - Think about which all points that a codeword can go to under error (spheres of Hamming radius s)
  - If spheres overlap, no decoding algorithm can decode
  - Closest codeword is the “correct” codeword.
    - So decoding is “min distance decoding”
  - Naïve way of achieving min-dist-decoding is brute force search across all codewords. There are efficient ways of getting to the closest codeword when codes have structure.

Recap: Linear Codes

If Σ is a field, then Σ^n is a vector space

Definition: C is a linear code if it is a linear subspace of Σ^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:

$$c = a_1 v_1 + a_2 v_2 + \ldots + a_k v_k \quad \text{where } a_i \in \Sigma$$

"Linear": linear combination of two codewords is a codeword.

Minimum distance = weight of least-weight codeword

Recap: Generator and Parity Check Matrices

Generator Matrix:

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

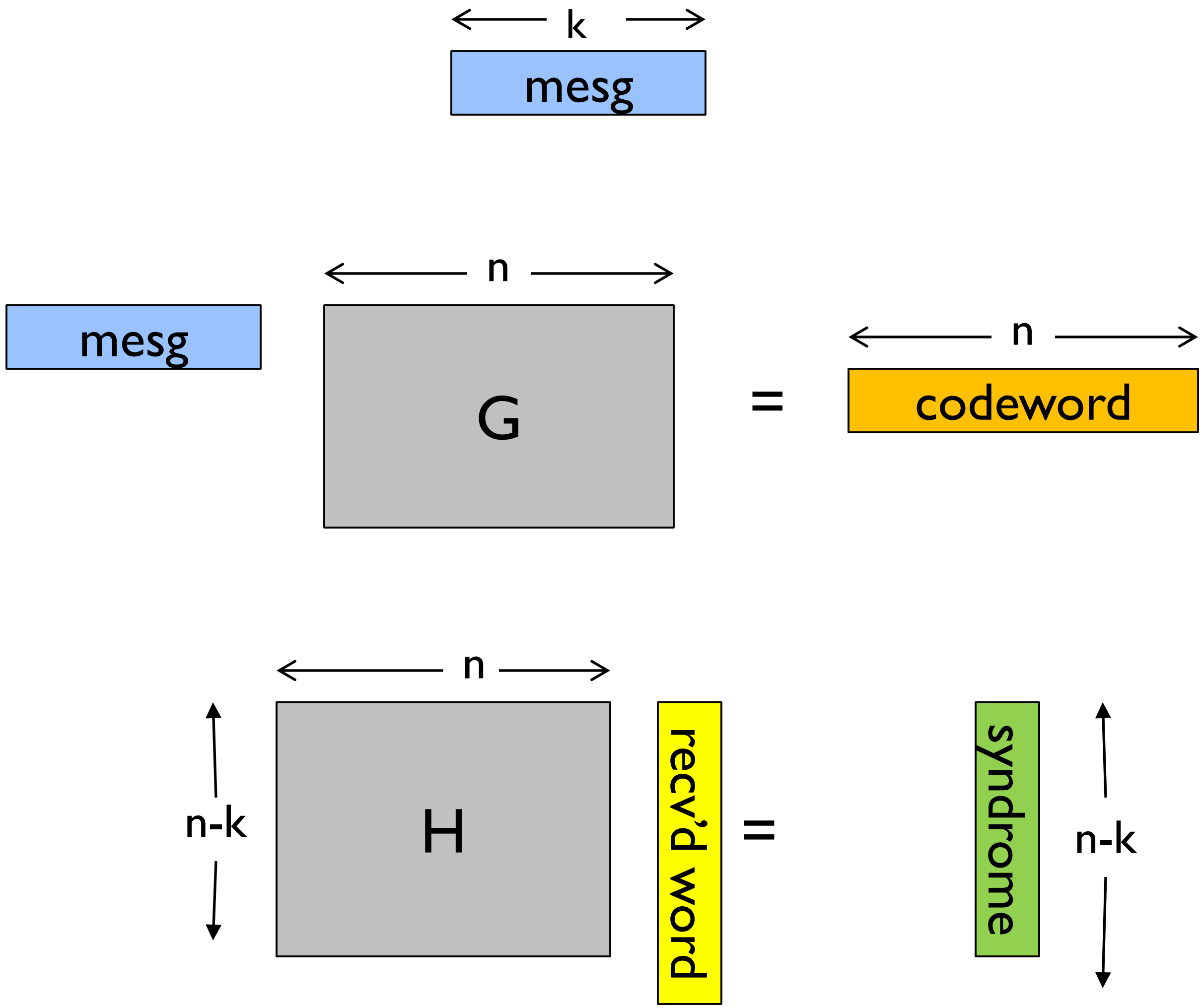
Made from stacking the spanning vectors

Parity Check Matrix:

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

(Codewords are the null space of H.)

These **always exist for linear codes**



if syndrome = 0, received word = codeword

else use syndrome to get back codeword

Recap: Linear Codes

Basis 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

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.

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

G is said to be in "**standard form**"

Relationship of G and H

Theorem: For binary codes, if G is in standard form $[I_k\ A]$ then $H = [A^T\ I_{n-k}]$

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

transpose

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

Relationship of G and H

Proof: <Board>

Two parts to prove:

1. Suppose that x is a message. Then $H(xG)^T = 0$.

2. Conversely, suppose that $Hy^T = 0$. Then y is a codeword.

Relationship of G and H

The above proof held only for $\mathbb{F}_2$.

Q: What about other alphabets?

For codes over a general field $\mathbb{F}_q$,

if G is of the standard form $[I_k, A]$

then the parity check matrix $H = [-A^T \; I_{n-k}]$

In the binary case, $-A = A$ and hence the principle is the same

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.

transpose

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

High level idea: for linear codes, distance equals least weight of non-zero codeword. And each codeword gives some collection of columns that must sum to zero.

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.

If some set S of d-1 columns were linearly dependent, then

$$\sum_{i \in S} c_k H_k = 0$$

But then y which has zeroes on coordinates outside S, and c_i for each coordinate $i \in S$ satisfies $Hy = 0$,
so is codeword of weight < d, a contradiction.

Conversely, distance d means there's a codeword y of weight d, which means $Hy = 0$ and hence the columns of H for the non-zero coordinates of y are linear dependent.

Dual Codes


Jacques Hadamard
(1865-1963)

For every code with

$G = [I_k\ A]$ and $H = [A^T\ \ I_{n-k}]$

we have a **dual code** with

$G = [I_{n-k}\ A^T]$ and $H = [A\ I_k]$

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

Dual Codes

For every code with

$G = [I_k \; A]$ and $H = [A^T \; I_{n-k}]$

we have a **dual code** with

$G = [I_{n-k} \; A^T]$ and $H = [A \; I_k]$

Irving Reed

David Muller

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

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

Note that these codes are highly redundant, with very low rate. Where would these be useful?

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

Dual Codes

For every code with

$G = [I_k\ A]$ and $H = [A^T\ I_{n-k}]$

we have a **dual code** with

$G = [I_{n-k}\ A^T]$ and $H = [A\ I_k]$

Dual of (7, 4, 3) Hamming code has generator matrix

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

Note: every non-zero r-bit vector appears as a column.

Lemma: this is a $(2^r - 1, r, 2^{r-1})$ code.
Proof: <discuss>

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 $s \leq (d-1)/2$ errors, every syndrome value corresponds to a unique set of error locations.

Proof: HW exercise.

Keep table of all these syndrome values. Has q^{n-k} entries, each of size at most n (i.e. keep a bit vector of locations).

Generic algorithm: not efficient for large values of (n-k)!

(Better algorithms exists for special codes.)

Consider a (5,2) linear block code:

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

Its standard array table:

	codewords				syndrome
	00000	10101	01110	11011	**000**
	00001	10100	01111	11010	**001**
	00010	10111	01100	11001	**010**
error vectors with same syndrome	00100	10001	01010	11111	**100**
	01000	11101	00110	10011	**110**
	10000	00101	11110	01011	**101**
	11000	01101	10110	00011	**011**
	10010	00111	11100	01001	**111**

Example drawn from Bill Cherowitzo's notes.

Another very useful bound: Singleton bound

Theorem: For every $(n, k, d)_q$ code, $n \geq k + d - 1$

Proof:

<board>

Codes that meet Singleton bound with equality are called **Maximum Distance Separable (MDS)**

Maximum Distance Separable (MDS)

Q: Are Hamming codes MDS? <board>

Only two binary MDS codes!

Q: What are they?

1. Repetition codes
2. Single-parity check codes

Need to go beyond the binary alphabet!

(We will need some number theory for this)

Number Theory Outline

Groups

- Definitions, Examples, Properties
- Multiplicative group modulo n

Fields

- Definition, Examples
- Polynomials
- Galois Fields

Number theory is crucial for arithmetic over finite sets.

Groups

A **Group** (G,*,I) is a set G with operator * such that:

1. **Closure**. For all $a,b \in G$, $a * b \in G$
2. **Associativity.** For all $a,b,c \in G$, $a*(b*c) = (a*b)*c$
3. **Identity.** There exists $I \in G$, such that for all $a \in G$, $a*I=I*a=a$
4. **Inverse.** For every $a \in G$, there exist a unique element $b \in G$, such that $a*b=b*a=I$

An **Abelian or Commutative Group** is a Group with the additional condition

5. **Commutativity.** For all $a,b \in G$, $a*b=b*a$

Examples of groups

Q: Examples?

- Integers, Reals or Rationals with Addition
- The nonzero Reals or Rationals with Multiplication
- Non-singular n x n real matrices with Matrix Multiplication
- Permutations over n elements with composition
  [0→1, 1→2, 2→0] o [0→1, 1→0, 2→2] = [0→0, 1→2, 2→1]

Often we will be concerned with **finite groups**, I.e., ones with a finite number of elements.

(We will start with finite groups in the next lecture)