15-853:Algorithms in the Real World

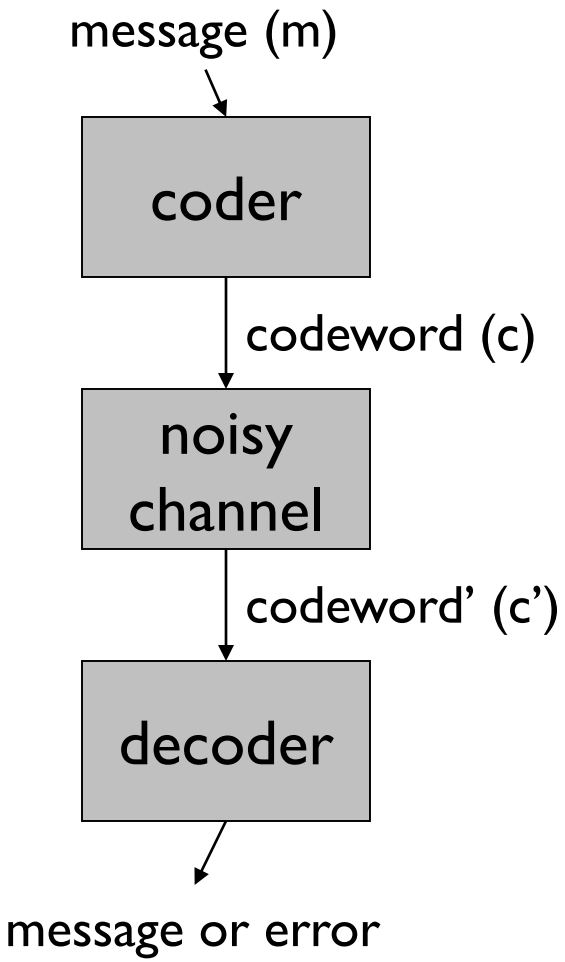
Error Correcting Codes (cont..)

Continuing with
primer on number theory

Announcement:

Slight modification in scribing structure

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)

Δ**(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: Linear Codes

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

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

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

i.e. every codeword can be written as:

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

"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: $C = \{ xG \mid x \in \sum^k \}$

Made from stacking the spanning vectors

Parity Check Matrix:

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

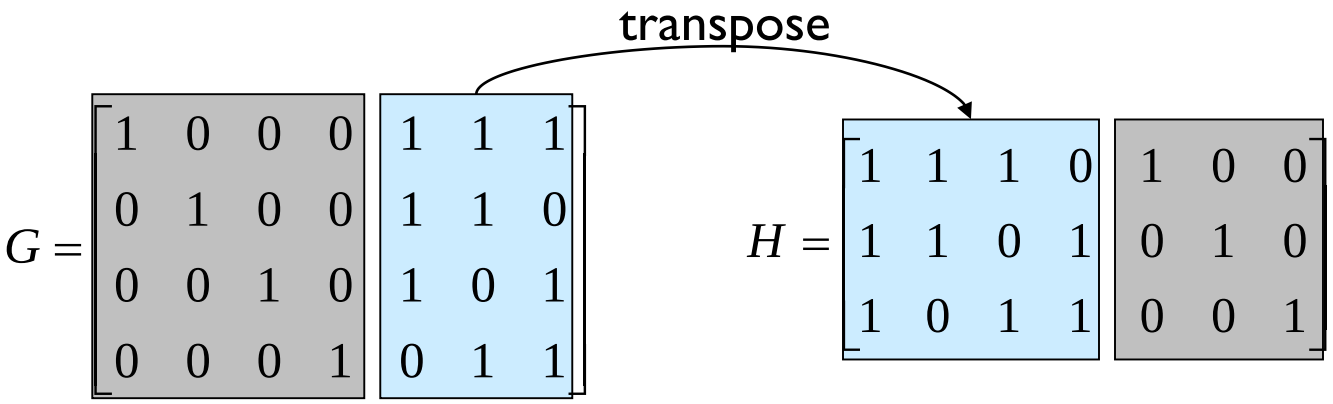
(Codewords are the null space of H.)

These **always exist for linear codes**

Recap: Relationship of G and H

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

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



Recap: 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 $H = [A \; I_k]$

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.

Recap: Properties of Syndrome and connection to error locations

Hy^T is called the **syndrome** (0 if a valid codeword).

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.

Recap: Singleton bound and MDS codes

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

Codes that meet Singleton bound with equality are called

Maximum Distance Separable (MDS)

Only two binary MDS codes!

1. Repetition codes
2. Single-parity check codes

Need to go beyond the binary alphabet!

(We will need some number theory for this)

Recap: 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 $a,b \in G$, $a*b=b*a$

Recap: Examples of groups

- 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
  $[0\to1, 1\to2, 2\to0] \circ [0\to1, 1\to0, 2\to2] = [0\to0, 1\to2, 2\to1]$

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

Groups based on modular arithmetic

The group of positive integers modulo a prime *p*

$Z_p^* \equiv \{1, 2, 3, \ldots, p\text{-}1\}$ $\quad\quad$ $*_p \equiv$ multiplication modulo p

Denoted as: $(Z_p^*, *_p)$

Required properties

1. Closure. Yes.
2. Associativity. Yes.
3. Identity. 1.
4. Inverse. Yes. (HW)

Example: $Z_7^* = \{1,2,3,4,5,6\}$

$1^{-1} = 1$, $2^{-1} = 4$, $3^{-1} = 5$, $6^{-1} = 6$

Fields

A **Field** is a set of elements F with binary operators * and + such that

1. (F, +) is an **abelian group**
2. (F \ I_+, *) is an **abelian group** the "multiplicative group"
3. **Distribution**: a*(b+c) = a*b + a*c
4. **Cancellation**: a*I_+ = I_+

Example: The reals and rationals with + and * are fields.

The **order (or size)** of a field is the number of elements.

A field of finite order is a **finite field**.

Finite Fields

$\mathbb{Z}_p$ (p prime) with + and * mod p, is a **finite** field.

1. $(\mathbb{Z}_p, +)$ is an **abelian group** (0 is identity)
2. $(\mathbb{Z}_p \setminus 0, *)$ is an **abelian group** (1 is identity)
3. **Distribution**: a*(b+c) = a*b + a*c
4. **Cancellation**: a*0 = 0

We denote this by $\mathbb{F}_p$ or GF(p)

Are there other finite fields?

What about ones that fit nicely into bits, bytes and words (i.e with 2^k elements)?

Polynomials over $\mathbb{F}_p$

$\mathbb{F}_p[x]$ = polynomials on x with coefficients in $\mathbb{F}_p$.

- Example of $\mathbb{F}_5[x]$: $f(x) = 3x^4 + 1x^3 + 4x^2 + 3$
- $\deg(f(x)) = 4$ (the **degree** of the polynomial)

Operations: (examples over $\mathbb{F}_5[x]$)

- Addition: $(x^3 + 4x^2 + 3) + (3x^2 + 1) = (x^3 + 2x^2 + 4)$
- Multiplication: $(x^3 + 3) * (3x^2 + 1) = 3x^5 + x^3 + 4x^2 + 3$
- $I_+ = 0$, $I_* = 1$
- + and * are associative and commutative
- Multiplication distributes and 0 cancels

Do these polynomials form a field?

Division and Modulus

Long division on polynomials ($\mathbb{F}_5[x]$):

$$
\begin{array}{r}
\boxed{1x+4} \\
x^2+1 \;\overline{)\; x^3+4x^2+0x+3} \\
\underline{x^3+0x^2+1x+0} \\
4x^2+4x+3 \\
\underline{4x^2+0x+4} \\
\boxed{4x+4}
\end{array}
$$

$$(x^3+4x^2+3)/(x^2+1)=(x+4)$$

$$(x^3+4x^2+3)\bmod(x^2+1)=(4x+4)$$

$$(x^2+1)(x+4)+(4x+4)=(x^3+4x^2+3)$$

Polynomials modulo Polynomials

How about making a field of polynomials modulo another polynomial?

This is analogous to $\mathbb{F}_p$ (i.e., integers modulo another integer).

Need a polynomial analogous to a prime number…

Definition: An **irreducible polynomial** is one that is not a product of two other polynomials both of degree greater than 0.

e.g. $(x^2 + 2)$ for $\mathbb{F}_5[x]$

Galois Fields

The polynomials $\mathbb{F}_p[x] \bmod p(x)$ where

1. $p(x) \in \in \mathbb{F}_p[x]$, p(x) is irreducible and
2. deg(p(x)) = n

form a finite field.

Q: How many elements?

Such a field has p^n elements.

These fields are called **Galois Fields** or **GF(p^n)** or $\mathbb{F}_{p^n}$

The special case n = 1 reduces to the fields $\mathbb{F}_p$.

The special case p = 2 is especially useful for us.

GF(2^n)

$\mathbb{F}_{2^n}$ = set of polynomials in $\mathbb{F}_2[x]$ modulo
irreducible polynomial $p(x) \in \mathbb{F}_2[x]$ of degree n.

Elements are all polynomials in $\mathbb{F}_2[x]$ of degree $\leq n-1$.

Has 2^n elements.

Natural correspondence with bits in $\{0,1\}^n$.

Elements of $\mathbb{F}_{2^8}$ can be represented as **a byte**, one bit for each term.

E.g., $x^6 + x^4 + x + 1 = 01010011$

GF(2^n)

$\mathbb{F}_{2^n}$ = set of polynomials in $\mathbb{F}_2[x]$ modulo
irreducible polynomial $p(x) \in \mathbb{F}_2[x]$ of degree n.

Elements are all polynomials in $\mathbb{F}_2[x]$ of degree $\leq n-1$.

Has 2^n elements.

Natural correspondence with bits in $\{0,1\}^n$.

Addition over $\mathbb{F}_2$ corresponds to xor.

- Just take the xor of the bit-strings (bytes or words in practice). This is dirt cheap.

Multiplication over GF(2^n)

If n is small enough can use a table of all combinations.

The size will be 2^n x 2^n (e.g. 64K for $\mathbb{F}_{2^8}$)

Otherwise, use standard shift and add (xor)

Note: dividing through by the irreducible polynomial on an overflow by 1 term is simply a test and an xor.

e.g. 0111 mod 1001 = 0111

1011 mod 1001 = 1011 xor 1001 = 0010

^ just look at this bit for $\mathbb{F}_{2^3}$

Finding inverses over GF(2^n)

Again, if n is small just store in a table.

- Table size is just 2^n.

For larger n, use Euclid's algorithm.

- This is again easy to do with shift and xors.

Euclid's Algorithm

Euclid's Algorithm:

gcd(a,b) = gcd(b,a mod b)

gcd(a,0) = a

"Extended" Euclid's algorithm:

- Find **x** and **y** such that **ax + by = gcd(a,b)**
- Can be calculated as a side-effect of Euclid's algorithm.
- Note that **x** and **y** can be zero or negative.

This allows us to find **a^{-1} mod n**, for **a** $\in Z_n^*$

Q: Any idea how?

In particular return **x** in **ax + ny = 1**.

Polynomials with coefficients in GF(p^n)

Recall that $\mathbb{F}_{p^n}$ was defined in terms of coefficients that were themselves fields (*i.e.*, $\mathbb{F}_p$).

We can apply this **recursively** and define:

$\mathbb{F}_{p^n}[x]$ = polynomials on x with coefficients in $\mathbb{F}_{p^n}$.

- Example of $\mathbb{F}_{2^3}[x]$:
  - $f(x) = 001x^2 + 101x + 010$
    Where 101 is shorthand for x^2+1.

Polynomials with coefficients in GF(p^n)

We can make a finite field by using an irreducible polynomial M(x) selected from $\mathbb{F}_{p^n}[x]$.

For an order m polynomial and by abuse of notation we write: **GF(GF(p^n)^m)**

Q: How many elements?

p^{nm} elements.

Note: all finite fields are isomorphic to GF(p^n) for some p,n so GF(GF(2^8)^4) is just another representation of GF(2^{32}).

This representation, however, has practical advantages.

The operations are more modular, easier to implement.

We now have the number theory basics in place to learn about codes over alphabets beyond binary.

Two Codes

Hamming codes are binary $(2^r\text{-}1\text{–}1, 2^r\text{-}1\text{-}r, 3)$ codes.

Basically $(n, n - \log n, 3)$

Hadamard codes are binary $(2^r\text{-}1, r, 2^{r-1})$.

Basically $(n, \log n, n/2)$

The first has great rate, small distance.

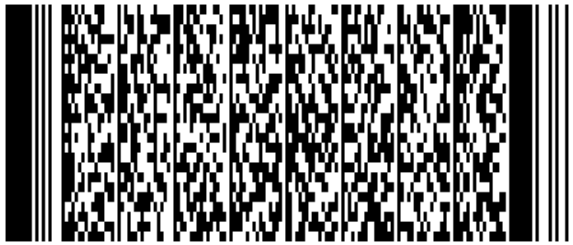
The second has poor rate, great distance.

Can we get $\Omega(n)$ rate, $\Omega(n)$ distance?

Yes. Let's see how...

Reed-Solomon (RS) Codes

Irving S. Reed and Gustave Solomon



PDF-417

QR code

Aztec code

DataMatrix code

2-dimensional Reed-Solomon bar codes

images: wikipedia

RS code: Polynomials viewpoint

Message: $[a_{k-1}, \ldots, a_1, a_0]$ where $a_i \in GF(q^r)$

Consider the polynomial of degree k-1

$$P(x) = a_{k-1} x^{k-1} + \cdots + a_1 x + a_0$$

RS code:

Codeword: [P(1), P(2), …, P(n)]

To make the i in p(i) distinct, need field size $q^r \geq n$

That is, need sufficiently large field size for desired codeword length.

Minimum distance of an (n, k) RS code

Theorem: RS codes have minimum distance d = n-k+1

Proof: Any ideas?

Hint: Is it a linear code?

1. *RS is a linear code:* if we add two codewords corresponding to P(x) and Q(x), we get a codeword corresponding to the polynomial P(x) + Q(x). Similarly any linear combination..
2. *So look at the least weight codeword*. It is the evaluation of a polynomial of degree k-1 at some n points. So it can be zero on only k-1 points. Hence non-zero on at most (n-(k-1)) points. This means distance at least n-k+1
3. Apply Singleton bound

Meets Singleton bound: RS codes are MDS

Generator matrix of RS code

Q: What is the generator matrix?

<board>

“Vandermonde matrix”

Special property of Vandermonde matrices:
Full rank (columns linearly independent)

Vandermonde matrix: Very useful in constructing codes.