DD2442 AUTUMN 2014 LECTURE 8

ERROR-CORRECTING CODES

Want to transmit message x over noisy channel

Encode message with redundancy so that it can be recovered even if transmission distorted

$$x \xrightarrow{\text{encode}} \underset{\text{codeword}}{E(x)} \xrightarrow{\text{channel}} \underset{\text{received word}}{y} \xrightarrow{\text{decode}} D(y)$$

<u>Want</u>:

- $D(y) = x$ even if some noise was introduced
- Encoding and decoding efficient
- Encoding doesn't blow up message too much

Some commonly considered channels:

- Binary symmetric channel:
  Bits $\{0,1\}$ get flipped with probability p
- Binary erasure channel:
  Bits get erased with probability p, in which case "?" is received
- Worst case:
  An evil adversary is allowed to flip a fraction p of the bits in some malicious way

Coding theory initiated by two papers

Shannon 1948: "A mathematical theory of communication"

Hamming 1950: "Error detecting and error correcting codes"

Both papers in Bell System Technical Journal — Shannon and Hamming were colleagues at Bell Labs

Shannon analysed exactly what can and cannot be done for binary symmetric channel, binary erasure channel, and many other channels.
Serious drawback: Highly non-constructive results.

Hamming focused on constructive results
Explicit (and efficient) encoding and decoding functions + analysis of transmission guarantees

We will focus on worst-case guarantees
(perhaps most well-studied channel, at least in TCS?)

NOTATION AND TERMINOLOGY

Codes are over ALPHABET Σ. Always $q = |\Sigma|$
BINARY CODES $\Sigma = \{0,1\}$
Other alphabets will also be of interest

HAMMING DISTANCE $\Delta(x,y)$ $x,y \in \Sigma^n$

$$\Delta(x,y) = \left| \{ i \in [n] \mid x_i \neq y_i \} \right|$$

This is indeed a metric

$$\Delta(x,y) = \Delta(y,x)$$

$$\Delta(x,y) \geq 0 \quad \text{with equality iff } x=y$$

$$\Delta(x,z) \leq \Delta(x,y) + \Delta(y,z)$$

For now, identify code with the image of the encoding function.
Will get back to encoding / decoding later

ERROR-CORRECTING CODE simply subset

$C \subseteq \Sigma^n$ $\quad x \in C$ CODEWORD

MINIMUM DISTANCE of code $\Delta(C) = \min_{\substack{x,y \in C \\ x \neq y}} \{\Delta(x,y)\}$

Fundamental parameters for $C \subseteq \Sigma^n$

1) BLOCK LENGTH n
2) MESSAGE LENGTH $k = \log_q |C|$
   (There are q^k possible messages that can be sent; can think of encoding function $E : \Sigma^k \to \Sigma^n$)
3) MINIMUM DISTANCE $d = \Delta(C)$
   (will affect possibility of correcting errors)
4) ALPHABET SIZE $q = |\Sigma|$

We say that C is an $(n,k,d)_q$ - code

Broad goals for $(n,k,d)_q$ - codes

a) Given k, d, q, minimize block length n
b) Given n, d, q, maximize message length k
c) Given n, k, q, maximize distance d
d) Given n, k, d, minimize alphabet size q

(a) - (c) obvious — small block length, large message size, and large distance always desirable

(d) not so clear. But empirically, the smaller the q, the harder it is to get good parameters. And in some sense, at least from a CS perspective it seems $q=2$ would be the really interesting case

In fact, often a good idea to
(i) build great code for large q;
(ii) find clever way of reducing q without destroying other parameters

We will hopefully see some examples of this

Hamming considered 3 properties of a code C
1) Minimum distance
2) Error detection capacity
C is e-error detecting if, under the promise that no more than e errors occur during transmission, it is always possible to detect whether errors have occurred or not, and e maximal with this property
Requires minimum distance $e+1$
3) Error correction capacity
C is t-error correcting if, under promise that no more than t errors occur, it is always possible to ~~locate~~ locate and correct errors (information-theoretically, but not necessarily efficiently) and t is maximal with this property

t-error correcting code has min distance $2t+1$ or $2t+2$

We will focus for now on distance (and thus error-correction capability)

Later discuss how to solve algorithmic task of error correction

Goals for lectures on error-correcting codes

- Cover some basic concepts, terminology, notation, facts
- Build some nice codes using vector spaces and polynomials over finite fields
- Prove some upper bounds on what kind of code constructions are possible
- Define/understand what "good codes" means (codes that get somewhat close to best possible)
- Give increasingly better constructions of codes (polynomials will play a key role)
- See that very good codes can be constructed that are also extremely efficient to encode and decode (expander graphs will play a key role)

Somewhere in the middle will have an interlude with a guest lecturer speaking about other topic (due to calendar constraints – such is life)

Let us start by warming up...

Some simple codes

$d = 1$ Trivial. Can use identity for encoding which yields $(n, n, 1)_q$ - code

$d = 2$ Already interesting
Append parity of all the message bits
(If messages differ in just one coordinate parities will also be different.
For general q, identify $\Sigma = \mathbb{Z}_q$
(additive group of integers mod q)
and use sum of all message symbols
~~ase~~ as check symbol. Yields $(n, n-1, 2)_q$ - code

$d = 3$ Non-trivial. Will get back to this

Perhaps $(n, n-2, 3)_q$ - code possible?
Or more generally $(n, n+1-d, d)_q$ - code?
Nope. $d=3$ counter-example, as proven by Hamming

To discuss this, we need to develop some more terminology and concepts.

Associate Σ_i with field $\mathbb{F}_q$ (so q prime power). Think of codewords as living in vector space $\mathbb{F}_q^n$

$\mathcal{L} \subseteq \mathbb{F}_q^n$ is a LINEAR SUBSPACE if
$\forall x, y \in \mathcal{L} \quad \forall \alpha \in \mathbb{F}_q$
- $x + y \in \mathcal{L}$
- $\alpha x \in \mathcal{L}$

C is a LINEAR CODE if $C \subseteq \mathbb{F}_q^n$ is a linear subspace of $\mathbb{F}_q^n$.

We denote that code is linear by using square brackets: C is an $[n,k,d]_q$-code if C is an $(n,k,d)_q$-code that is linear

We can use linear algebra to succinctly represent linear codes

GENERATOR MATRIX

Can specify linear subspace by basis linearly independent set of vectors $x_1, \ldots, x_k \in \Sigma_i^n = \mathbb{F}_q^n$ such that

$$C = \left\{ \sum_{i=1}^{k} \alpha_i x_i \;\middle|\; \alpha_i \in \mathbb{F}_q \right\}$$

~~The~~ A GENERATOR MATRIX $G \in \mathbb{F}_q^{k \times n}$ for C is a matrix with the <u>rows</u> of G forming a basis of C

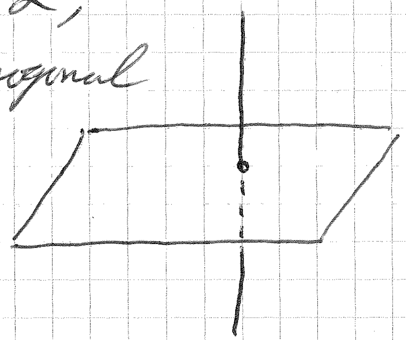
$$C = \left\{ \alpha G \;\middle|\; \alpha \in \mathbb{F}_q^k \right\}$$

PARITY CHECK MATRIX

The NULL SPACE of a linear subspace $\mathcal{L} \subseteq \mathbb{F}_q^n$ is $\mathcal{L}^\perp = \{ y \in \mathbb{F}_q^n \mid \langle x, y \rangle = 0 \ \forall x \in \mathcal{L} \}$

FACT 1 The nullspace $\mathcal{L}^\perp$ of a linear subspace $\mathcal{L} \subseteq \mathbb{F}_q^n$ is a linear subspace of $\mathbb{F}_q^n$ of dimension $n - \dim(\mathcal{L})$

This might seem geometrically obvious.
If $\mathcal{L} \subseteq \mathbb{R}^3$ has dimension 2, then $\mathcal{L}$ plane and $\mathcal{L}^\perp$ orthogonal line



But $\langle \cdot, \cdot \rangle$ is not a true inner product

Example Suppose $q = 2$, $n = 2$
$$C = \{(0,0), (1,1)\}$$
Then $C^\perp = C$

Yes, succumbing to the notation in Madhu Sudan's ECC lecture notes, we will now often write vectors as rows

$C^\perp$ is also a linear code; the DUAL CODE

Let $H' \in \mathbb{F}_q^{(n-k) \times n}$ be a generator matrix for $C^\perp$. Let $H = (H')^T$

Then $H \in \mathbb{F}_q^{n \times (n-k)}$ is a PARITY CHECK MATRIX of C

FACT 2 $C = \{ y \in \mathbb{F}_q^n \mid yH = 0 \}$

This is in some sense obvious, but here is an explanation why it is obvious.

~~Rows~~ Columns of H are basis for $C^\perp$

Hence, if $yH = 0$, then $y \in (C^\perp)^\perp$

Clearly all $x \in C$ satisfy this,

so $C \subseteq (C^\perp)^\perp$

But $\dim(C) = \dim((C^\perp)^\perp)$ by Fact 1,

so $C = (C^\perp)^\perp$.

Generator matrix
Parity check matrix $\Big\}$ both $O(n^2 \log q)$-size representation of C

Fairly compact

Can be shown that representations are essentially equivalent computationally

Can compute one from the other using linear algebra

So why have both? Generator matrix would seem to be sufficient... But parity check matrices can show | useful since they | properties of a code

Applications of parity check matrix

PROPOSITION 3

Error-detection for linear codes can be done efficiently using the parity check matrix

Proof

Suppose received word y transmission of codeword from distance-d code

Guarantee that fewer than $d-1$ errors in transmission.

Compute yH

If $yH = 0$, then $y \in C$ and no error occurred

If $yH \neq 0$, then y is not a codeword and an error must have occurred

Problem Given linear code C, compute its min distance $\Delta(C)$

NP-hard problem ...

But using parity check matrices can help us design codes with reasonable minimum distance

The HAMMING WEIGHT of $x \in \mathbb{F}_q^n$ is

$$wt(x) = \left| \{ i \mid x_i \neq 0 \} \right|$$

i.e. $wt(x) = \Delta(x, 0)$.

To find min distance of linear code, find non-zero codeword x of smallest weight such that $x \cdot H = 0$. Then $wt(x) = \Delta(C)$

PROPOSITION 4

Let C linear code with parity check matrix H. Then $\Delta(C)$ equals smallest d such that there exist d linearly dependent rows in H.

Proof Note that this is equivalent to the claim

$$\Delta(C) = \min_{x \neq 0} \{ wt(x) \mid xH = 0 \}$$

Let h_i ith row of H

Let $x_{i_1}, \ldots, x_{i_d}$ non-zero coordinates of x

Then $\sum_{j=1}^{d} x_{i_j} \cdot h_{i_j} = 0$ and hence $\{ h_{i_j} \mid j \in [d] \}$ is a linearly dependent set of rows. In the other direction, any collection of d linearly dependent rows by definition yield x of weight d s.t. $xH = 0$.

So we need to show $\min \{ wt(x) \mid x \neq 0, xH = 0 \} = \Delta(C)$.

Clearly $\Delta(C) \leq wt(x)$ since 0 is a codeword (by linearity) and $\Delta(x, 0) = wt(x)$.

Suppose $y \neq z \in C$ minimize $\Delta(y, z)$

then $y - z \in C$ by linearity and

$$\Delta(C) = \Delta(y, z) = \Delta(y - z, 0) = wt(y - z)$$

HAMMING CODES

Now we can construct codes with minimum distance $d = 3$

For simplicity, let's focus on $\mathbb{F}_2 = \{0, 1\}$

Want to construct matrix $H \in \mathbb{F}_q^{n \times (n-k)}$ such that if $wt(x) = 1$ or 2 then $xH \neq 0$

Fix $\ell = n-k$ and try to maximize n

Let h_i = ith row of parity check matrix

If $wt(x) = 1$ then $x = e_i$ and $xH = h_i$

Hence if all rows in parity check matrix are non-zero then min dist of code ≥ 2

If $wt(x) = 2$ then $x = e_i + e_j$ for some $i \neq j$. and $xH = h_i + h_j$. We thus need $h_i \neq h_j$ for all $i \neq j$.

<u>Conclusion</u> H is parity check matrix for code of distance ≥ 3 if all rows distinct and non zero

Largest n = # distinct non-zero vectors in $\mathbb{F}_2^{\ell}$

there are $2^{\ell} - 1$ such vectors. This proves the following proposition

PROPOSITION 5 For every $\ell \in \mathbb{N}^+$, $\ell \geq 2$ there exists an $[2^{\ell}-1, 2^{\ell}-\ell-1, 3]_2$ - code, called the (binary) Hamming code

Can let rows in H be integers $1, 2, \ldots, 2^{\ell}-2, 2^{\ell}-1$ written in binary

If $k = 2^l - l - 1$, then $l = \Theta(\log k)$
thus, Hamming codes are obtained
by adding $\Theta(\log k)$ parity check bits.

Min distance 3 $\Rightarrow$ error correction capacity 1.

Given received word with one error
how can we locate position of error?

1) Brute force Flip each bit in y until
find $y' = y + e_i$ with $y'H = 0$. Then
error is in position i
$O(n^3)$ algorithm

2) Use parity check matrix in a smarter
way. One error in position i $\Leftrightarrow$
$y = x + e_i$
Since $x \in C$ we have
$yH = (x + e_i)H = e_i H = h_i$
But h_i is integer i written in binary
$O(n^2)$ algorithm.

Is it necessary to add logarithmically
many bits to be able to (not only detect but)
correct a single error?
Hamming proved that the
answer is <u>yes</u>!

Let $B_q(y,r) = \{x \in \mathbb{F}_q^n \mid \Delta(x,y) \le r\}$ denote the HAMMING BALL of radius r centred at y in $\mathbb{F}_q^n$

Let $Vol_q(r,n)$ denote the volume of any radius-r ball ~~i.e.~~ in $\mathbb{F}_q^n$, i.e.

$$Vol_q(r,n) = |B_q(y,r)|$$
$$= \sum_{i=0}^{r} \binom{n}{i}(q-1)^i$$

For binary codes (i.e., $\mathbb{F}_2^n$ for linear codes) we drop subscript

THEOREM 6 (HAMMING BOUND)
If an $(n,k,d)_q$-code exists, then

$$q^k \, Vol_q\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n\right) \le q^n$$

<u>Proof</u> If x,y distinct codewords of $(n,k,d)_q$-code and if we let $r = \left\lfloor \frac{d-1}{2} \right\rfloor$, then $B(x,r)$ and $B(y,r)$ are disjoint.

Since $\bigcup_{x \in C} B(x,r) \subseteq \Sigma^n$ and sets on LHS are all disjoint we get

$$\sum_{x \in C} B(x,r) =$$
$$q^k \, Vol_q(r) \le q^n$$

$\boxtimes$

For $d = 3$, $q = 2$ we have

$$\mathrm{Vol}(1, n) = 1 + n$$

and Thm 4 yields

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

For the Hamming code in Prop 5 we have

$$n = 2^l - 1 \qquad \text{blocklength}$$
$$k = 2^l - l - 1 \qquad \text{message length}$$

and $2^k(n+1) = 2^{2^l - l - 1} \cdot 2^l = 2^{2^l - 1} = 2^n$

so ~~of~~ Hamming codes meet Hamming bound exactly. (Can define q-ary Hamming codes and this is true for them as well)

Thus, balls around codewords in Hamming code fill up ambient vector space $\mathbb{F}_q^n$ perfectly

DEF 7
An $(n,k,d)_q$-code is PERFECT if it meets the Hamming bound exactly, i.e., if

$$q^k \, \mathrm{Vol}_q\left(\left\lfloor \frac{d-1}{2} \right\rfloor, n\right) = q^n$$

Hamming codes are perfect
Two specific codes by Golay [1949] are perfect

No other perfect codes exist!
[van Lint 1971], [Tietavainen 1973]

So Hamming bound rules out $(n, n-d+1, d)_2$ - code already for $d = 3$. (Note this applies to binary codes $q=2$)

Can we do ~~better~~ for larger alphabets?
Can we get $(n, n-d+1, d)_q$ - codes for q large enough?

Why not be even more ambitious and ask for $k > n-d+1$?!

Because, as we will see next lecture, $k \leq n-d+1$ is an upper bound regardless of alphabet size.

Will also see next lecture that there are codes that meet this bound
So-called REED SOLOMON CODES
constructed by evaluating univariate polynomials over finite fields.