MSRI-UP 2009 Coding Theory Seminar–Week 1

John B. Little

Department of Mathematics and Computer Science
College of the Holy Cross

June 15 - 18, 2009

Outline

A bit of history

- Beginning of coding theory as a mathematical and engineering subject came with a paper “A Mathematical Theory of Communication” by Claude Shannon (1948).
- Shannon lived from 1916 t0 2001, and spent most of his working career at Bell Labs and MIT.
- He also made fundamental contributions to cryptography and the design of computer circuitry in earlier work coming from his Ph.D. thesis.
- Other interests – inventing gadgets, juggling, unicycles, chess(!)

Shannon’s conceptual communication set-up

$$\begin{tikzcd}
\text{message} \arrow[d] & & \boxed{\text{noise}} \arrow[d] & & \\
\boxed{\text{encoder}} \arrow[r] & \boxed{\text{trans.}} \arrow[r] & \boxed{\text{channel}} \arrow[r] & \boxed{\text{rec.}} \arrow[r] & \boxed{\text{decoder}} \arrow[d] \\
& & & & \text{message}
\end{tikzcd}$$

Examples

This is a *very general* framework, incorporating examples such as

- communication with deep space exploration craft (Mariner, Voyager, etc. – the most important early application)
- storing/retrieving information in computer memory
- storing/retrieving audio information (CDs)
- storing/rerieving video information (DVD and Blu-Ray disks)
- wireless communication

A main goal of coding theory is the design of coding schemes that achieve *error control*: ability to detect and correct errors in received messages.

Notes

Error correction

Bienvenqdos a Cnlifornix, MSRI-UP sumyer 2009 studenxs!

- If you can read this message, then you're doing error correction!
- In all human languages like Spanish or English, words are usually "far enough apart" that even if some of a message is corrupted, it may still be intelligible
- *because* there are only a few legal words that are "close" to what is contained in the received message. (This is why autocorrection by spell-checking software is possible too!)
- In the systems used for other types of communication, similar robustness in the presence of noise is a very desirable feature that can be "designed in."

Notes

Mathematical setting

- Messages
  - are divided into "words" or blocks of a fixed length, k,
  - use symbols from a finite *alphabet* A with some number q of symbols
- Simplest case (also best adapted to electronic hardware) is an alphabet with two symbols: $A = \{0, 1\}$, identified with the finite field $\mathbb{F}_2$ (addition and multiplication *modulo 2* – so 1 + 1 = 0), but we will see others later also.
- Usually, all strings or k-tuples in $\mathbb{F}_2^k$ are considered as possible words that can appear in a message.

Notes

Encoding and decoding

To correct errors, *redundancy* must included in the encoded message. One way: encoded message consists of strings or n-tuples of fixed length $n > k$ over the same alphabet. Then encoding and decoding are functions:

$$E : \mathbb{F}_2^k \to \mathbb{F}_2^n$$

and

$$D : \mathbb{F}_2^n \to \mathbb{F}_2^k$$

where E is 1-1, and $D \circ E = I$ on $\mathbb{F}_2^k$.
(D might also take a "FAIL" value on some words in the complement of $Im(E)$ containing too many errors to be decodable.)
We call $C = Im(E)$ the set of *codewords*, or the *code*. Any C of this form is called a *block code of length n*.

Notes

Errors

- Channel errors replace a *codeword* x by a *received word* $x' \neq x$.
- For $A = \mathbb{F}_2$ with sum operation satisfying the usual algebraic rules (commutativity, associativity, existence of a 0 element and additive inverses), then $x' = x + e$, where $e \in \mathbb{F}_2^n$ is the *error vector*.
- The *weight*
$$\mathrm{wt}(e) = |\{i \mid e_i \neq 0\}|$$
determines how many entries of x are corrupted.
- Example: $\mathrm{wt}(11000101) = 4$.
- *Decoding* is the same as determining e (somehow), then subtracting it off to recover x.

Notes

Hamming distance

For errors to be correctable, codewords must be “separated.”

Definition (Hamming Distance)

Let $x, y \in \mathbb{F}_2^n$. Then

$$\begin{aligned} d(x,y) &= |\{i \in \{1,\ldots,n\} : x_i \neq y_i\}| \\ &= \mathrm{wt}(x-y). \end{aligned}$$

Example: $d(11000111, 10100101) = 3$
Fact: $d(x,y)$ is a *metric* or *distance function* (on the finite set $\mathbb{F}_2^n$). In particular, the *triangle inequality*:

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

holds for all $x, y, z \in \mathbb{F}_2^n$ (discussion today).

Notes

Error-Correcting Capacity

The Hamming distance measures a code's error detecting and error correcting capacity.

Theorem

Let C be a code in $\mathbb{F}_2^n$.

1. *If $d(u,v) \geq s+1$ for all distinct $u, v \in C$, then all error vectors of weight s less can be detected.*
2. *If $d(u,v) \geq 2t+1$ for all distinct $u, v \in C$, all error vectors of weight t or less will be corrected by the "nearest-neighbor" decoding function:*

$$D(x) = E^{-1}(c \in C : d(x,c) \text{ is minimal}).$$

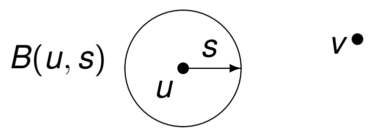
Notes

The proof

Proof: We write

$$B(u,s) = \{y \in \mathbb{F}_2^n : d(u,y) \leq s\},$$

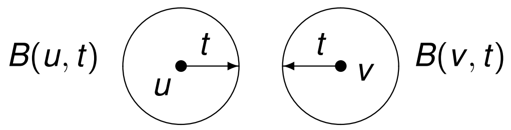
called the *closed ball with center u, radius s* for the Hamming distance.



Part 1: If $d(u,v) \geq s+1$ for all $u \neq v$ in C, then changing s entries in a codeword never produces another codeword.

Notes

The proof, continued



Part 2: Assume $d(u,v) \geq 2t+1$ for all $u \neq v \in C$. Then $B(u,t)$ and $B(v,t)$ must be *disjoint*. (If $y \in B(u,t) \cap B(v,t)$, then

$$d(u,v) \leq d(u,y) + d(y,v) \leq 2t$$

by the triangle inequality. Contradiction.) $\square$
Shows: If u is sent and $\mathrm{wt}(e) \leq t$, then $u+e$ is *still closer to u than any other codeword*, and nearest neighbor decoding will correct the error!

Notes

Minimum distance

The theorem shows that the *minimum distance* defined below is a very important parameter of codes.

Definition
Let C be a code in $\mathbb{F}_2^n$. The **minimum distance** *of C, denoted d or $d(C)$, is*

$$d = \min_{u \neq v \in C} d(u,v).$$

(That is, d gives the *smallest separation* between any two distinct codewords. The larger d is, the more widely separated all pairs of distinct codewords are.)

Notes

Comments

- For a general code C (where the set of codewords has no "extra structure"), to find d, it is necessary to compute Hamming distances between *all pairs* of distinct codewords.
- For instance, if $C = \{111001, 101010, 000110\}$, then $d(111001, 101010) = 3$, $d(111001, 000110) = 6$, $d(101010, 000110) = 3$, so $d(C) = 3$.
- If there are N codewords, this means $\binom{N}{2} = \frac{N(N-1)}{2}$ comparisons.

Notes

The theorem, reconsidered

- Can we always tell when errors occur?
- Clearly no, since the error could turn a codeword into a different codeword if it has large enough weight.
- If we use a code with $d = 2t + 1$, what happens if an error e with $\mathrm{wt}(e) > t$ is introduced by the channel?
- Nearest-neighbor decoding can *fail* and produce an incorrect decoding!
- Reason – the received word $u + e$ could be closer to a different $v \in C$ than to u: $d(u + e, v) < d(u + e, u)$(!)
- Are these possiblities likely to cause problems in our communications system?

Notes

More on channels

To answer a question like this, we really need to know more about the characteristics of the communications channel we are working with. Perhaps the simplest situation to understand and analyze is the class of *binary symmetric channels*, or BSC.

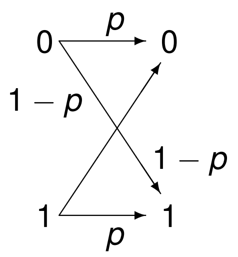
Assumptions:

- errors do not depend on the entries of the codeword transmitted
- bit-level errors are *random* – probability that a bit is sent correctly is some p, $0 < p < 1$, (so error probability is $1 - p$).
- bit level errors are *independent of each other* (knowing one bit was incorrect gives no information about others).

Notes

Schematic picture

The probabilities of a message bit on left being received as bit on right.



Note: “symmetric” refers to the symmetry between probabilities for 0s and 1s.

Notes

An example

Assume we have a BSC

- with $p = .995$ and $1 - p = .005$ (*that is:* out of every $10^3 = 1000$ bits transmitted, we expect 5 to be incorrect).
- capable of transmitting 10^6 bits per minute.

Say we want to send messages over the alphabet $\mathbb{F}_2$ divided into words of length 4.
Let's study three different scenarios.

- no encoding at all
- encoding with a "check bit"
- encoding each word of length 4 to a codeword of length 7, using a code with $d = 3$.

Notes

Scenario 1

With no error-control coding:

- Can send $10^6/4 = 250,000$ words each minute
- Expect about 5000 bit errors each minute
- Could have about 5000 (that is 2%) of the words incorrect (fewer, if errors "bunch up" in one word)
- In effect, we are using a code consisting of all of $\mathbb{F}_2^4$, so $d = 1$.
- By the theorem, we *cannot detect or correct any of the errors* (since error will *always* take a codeword to another codeword!)

Notes

Scenario 2

With a "check bit"

- Say we add a bit to each word so that the number of 1's is always even (so for instance $1100 \to 11000$ and $1011 \to 10111$).
- The code C consists of the 16 words of length 5 with an even number of 1's. Have $d(C) = 2$.
- By the theorem, we can *detect* any single bit error in a received word, but we cannot necessarily correct it.
- Can send $10^6/5 = 200,000$ codewords each minute
- Still expect about 5000 bit errors per minute, so as many as 5000 words could contain errors.
- Question: How many *undetected* errors could there be?

Notes

Scenario 2, continued

- Our question comes down to asking: *what is the probability of an even number of errors in a received word*?
- We have (by independence assumption on the bit errors!)

$$P(2 \text{ bit errors}) = \binom{5}{2}(.995)^3(.005)^2 \doteq .000246$$

$$P(4 \text{ bit errors}) = \binom{5}{4}(.995)^1(.005)^4 \doteq 3.11 \times 10^{-9}$$

- Since the probability of 4 bit errors is so much smaller, we will neglect it.
- Expect about $(200,000)(.000246) \doteq 49$ words to have undetectable errors each minute.

Notes

Scenario 3

Assume we can map words of length 4 to 16 codewords of length $n = 7$ and code has $d = 3$. (We will see such a code later in the week!)

- By the theorem, any 2 bit errors in a received word can be *detected*, and any single bit error in a received word can be *corrected* by nearest-neighbor decoding.
- Can send $10^6/7 \doteq 142,857$ codewords per minute
- Now in the length 7 received words,

$$P(2 \text{ bit errors}) = \binom{7}{2}(.995)^5(.005)^2 \doteq .000512$$
$$P(3 \text{ bit errors}) = \binom{7}{3}(.995)^4(.005)^3 \doteq .00000429$$

  and the probability of more bit errors is negligibly small.

Notes

Scenario 3, continued

- So we expect about $(142857)(.00000429) \doteq .613$ undetectable errors per minute
- We expect about $(142857)(.000516) \doteq 74$ uncorrectable errors per minute.
- But by the previous, almost all of these would be detectable, so the receiver could ask for retransmission!
- Also, recall that in Scenario 2 no error *correction* was possible.

Notes

Conclusions

- By incorporating more sophisticated coding, it is definitely possible to reduce the rate at which “unrecoverable” errors occur.
- But the decision which scenario to use in a case like this would be made by comparing the *tradeoffs* between efficiency (message words transmitted per minute) and additional error detection and correction capability.
- Even though it draws heavily on topics in pure mathematics (as we will see), coding *practice* is definitely an engineering discipline (EE, computer engineering)!

Notes

Codes with “extra structure”

Codes where the set of codewords is an arbitrary subset of $\mathbb{F}_2^n$ are somewhat awkward to work with.

- (Usually) need the complete list of codewords to specify the code
- Computationally difficult to determine parameters like d
- No general algebraic encoding or decoding procedures known
- So, most codes used in practice have some *extra algebraic structure*
- Today, we will introduce *linear codes*, where the set of codewords is a *vector subspace* of $\mathbb{F}_2^n$

Notes

Linear algebra over a general field

- In your study of linear algebra so far, you may have only dealt with $\mathbb{R}^n$, vector spaces over the field $\mathbb{R}$, and so forth.
- Even though the algebra of $\mathbb{F}_2$ with $+, \cdot$ mod 2 is very simple, most of the good properties of $\mathbb{R}$ hold for $\mathbb{F}_2$ as well:
  1. Addition is associative and commutative, there is a 0, each element has an additive inverse
  2. Multiplication is associative and commutative, there is a 1, each nonzero element has a multiplicative inverse
  3. Multiplication distributes over addition
- Any set $\mathbb{F}$ with $+, \cdot$ operations satisfying these properties is called a *field*.
- Hence $\mathbb{F}_2$ is a field. More generally, $\mathbb{F}_p$ is a field for any prime p.

Notes

Linear algebra over a general field, continued

The field axioms are precisely the usual properties needed for the scalars associated to a *vector space* V, and we can define vector spaces over any field $\mathbb{F}$.
Example. Let $\mathbb{F} = \mathbb{F}_2$, and let $V = \mathbb{F}_2^n$. We have a vector sum operation on V defined by component-wise addition mod 2, e.g.

$$110011 + 010110 = 100101.$$

We also have a scalar multiplication by the scalars in $\mathbb{F}_2$:

$$0 \cdot (x_1, \ldots, x_n) = (0, \ldots, 0) \quad 1 \cdot (x_1, \ldots, x_n) = (x_1, \ldots, x_n).$$

This makes $\mathbb{F}_2^n$ a *vector space* over $\mathbb{F}_2$.

Notes

Linear algebra over a general field, continued

Most of the basic constructions from linear algebra carry over to vector spaces like $\mathbb{F}_2^n$.

- A *linear combination* of vectors has the form $c_1x_1 + \cdots + c_nx_n$, where $c_i \in \mathbb{F}_2$, $x_i \in \mathbb{F}_2^n$.
- A subset W of $\mathbb{F}_2^n$ is a *vector subspace* of $\mathbb{F}_2^n$ if W is closed under all linear combinations.
- The *span* of a set $S \subset \mathbb{F}_2^n$, denoted $\langle S \rangle$, is the set of all linear combinations of vectors in S – a subspace of $\mathbb{F}_2^n$.
- A set S is said to be *linearly independent* if the only linear combination $c_1x_1 + \cdots + c_nx_n$ of vectors $x_i \in S$ that adds up to the zero vector is the combination with $c_i = 0$ for all i.
- Every vector subspace W in $\mathbb{F}_2^n$ has a *basis* – a set $S \subset W$ such that $\langle S \rangle = W$ and S is linearly independent.

Notes

Linear algebra over a general field, continued

- All bases for the same subspace W have the same number of elements.
- Hence it makes sense to define the *dimension* of W as $\dim(W) = |S|$, where S is any basis for W. We have $0 \leq \dim(W) \leq n$ for all W in $\mathbb{F}_2^n$.
- The set of solutions of any homogeneous system of linear equations in n variables with coefficients in $\mathbb{F}_2$ is a subspace of $\mathbb{F}_2^n$.
- The usual techniques of row operations on matrices and reduction to *row-reduced echelon form* work in the same way as over the field $\mathbb{R}$.

Notes

Linear codes

Definition

A **binary linear code** *of blocklength n is a vector subspace of* $\mathbb{F}_2^n$.

Definition

The **parameters** *of a binary linear code are* $[n, k, d]$, *where n is the block length, k is the dimension, and d is the minimum distance.*

Note: If a linear $[n, k, d]$ code C over $\mathbb{F}_2$ has a basis $S = \{x_1, \ldots, x_k\}$, then $C = \{c_1x_1 + \cdots + c_kx_k \mid c_i \in \mathbb{F}_2\}$, and linear independence means all these linear combinations are distinct, so we have 2^k different codewords.

Notes

Example

Let $S = \{11000, 10101, 01101, 11111\} \subset \mathbb{F}_2^5$.

- S is not a linear code itself, since it is not closed under linear combinations: $11000 + 11111 = 00111 \notin S$ for instance.
- S is not linearly independent: $11000 + 10101 + 01101 = 00000$
- Since we can rewrite the previous as $11000 + 10101 = 01101$ for instance, 01101 is *redundant* and can be omitted to get closer to a basis.

Notes

Example, continued

- If $a(11000) + b(10101) + c(11111) = 00000$, then

$$\begin{aligned} 0 &= a + b + c \\ 0 &= a + c \\ 0 &= b + c \\ 0 &= c \\ 0 &= b + c \end{aligned}$$

  Easy to see $a = b = c = 0$ so $S' = \{11000, 10101, 11111\}$ is linearly independent.
- Hence $\langle S \rangle = \langle S' \rangle$ is a linear code with $n = 5$, $k = 3$, containing these 8 code words: $\langle S' \rangle =$

  $\{00000, 11000, 10101, 11111, 01101, 00111, 01010, 10010\}$

Notes

What about d?

What is the minimum distance of our example code: $C = \langle S' \rangle = \{00000, 11000, 10101, 11111, 01101, 00111, 01010, 10010\}$?

- Note, when we compute the Hamming distance $d(u, v)$ for $u \neq v \in C$, then $d(u, v) = \mathrm{wt}(u - v) = \mathrm{wt}(u + v)$.
- But since C is a linear code, $u + v \neq 0 \in C$ as well.
- Hence we will have $d = \min_{w \neq 0 \in C} \mathrm{wt}(w) = 2$, so C is a $[5, 3, 2]$ code over $\mathbb{F}_2$.
- The same reasoning shows in general:

Proposition

Let C be a linear code in $\mathbb{F}_2^n$. Then $d(C) = \min_{w \neq 0 \in C} \mathrm{wt}(w)$.

(For linear codes, *minimum distance = minimum weight of a nonzero codeword*.)

Notes

Generator matrices

Definition

An $k \times n$ matrix G is a **generator matrix** *for a linear code C if its rows form a basis for C.*

A code of dimension $k > 1$ has many generator matrices.

- Given any spanning set S for C, we can find a generator matrix of a special form (echelon form) by forming the $|S| \times n$ matrix M whose rows are the vectors in S,
- taking M to RREF by elementary row operations,
- discarding any zero rows to form a new matrix G. The remaining rows are linearly independent and span C, hence G is a generator matrix.

Notes

Example

Let $S = \{11011, 10010, 01001\}$. Let's find an echelon-form generator matrix for $C = \langle S \rangle$ by the process described above.

- start with $M = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$
- add row 1 to row 2, then row 2 to row 3 to yield $\begin{pmatrix} 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
- add row 2 to row 1 and discard row 3 (all zeros): $\begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$ The final matrix is an echelon form generator matrix for a $[5, 2, 2]$ code.

Notes

Encoding with G

Given a generator matrix G for a linear code C, we have a simple way to do encoding from $\mathbb{F}_2^k$ to $C \subset \mathbb{F}_2^n$. Namely, since the rows of G form a basis for C, we can take entries of vectors in $\mathbb{F}_2^k$ as the scalars in linear combinations of the rows of G. This can even be done in a nice, matrix algebra way by computing $v \mapsto vG$ (the product of a $1 \times k$ and a $k \times n$ is $1 \times n$, a codeword). For instance, from the last example,

$$11 \mapsto \begin{pmatrix} 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Notes

The dual code

Given a spanning set (or a basis) $S = \{x_1, \ldots, x_k\}$ for a code C, we can also consider the set of $v \in \mathbb{F}_2^n$ such that $x_i \cdot v = 0$ for $i = 1, \ldots, k$. Here $x_i \cdot v$ is the formal *dot product*:

$$(x_{i1}, \ldots, x_{in}) \cdot (v_1, \ldots, v_n) = x_{i1}v_1 + \cdots + x_{in}v_n.$$

Note that

$$x_i \cdot v = 0 \text{ for all } i \Leftrightarrow x \cdot v = 0 \text{ for all } x \in C$$

Definition

The set of solutions v of the system of linear equations $x \cdot v = 0$ for all $x \in C$ is called the **dual code** *of C and denoted $C^\perp$.*

Notes

More on the dual code

Proposition

If C is a linear code of dimension k and blocklength n, then $C^\perp$ is a linear code of dimension $n-k$ and blocklength n.

Proof: If $u, v \in C^\perp$ and $c \in \mathbb{F}_2$, then $x \cdot u = x \cdot v = 0$ for all $x \in C$. But then $0 = x \cdot u + x \cdot cv = x \cdot (u + cv)$. This shows that $u + cv \in C^\perp$. Hence $C^\perp$ is also a linear code. The claim about the dimension can be seen by considering the RREF system of linear equations corresponding to a RREF generator matrix G for C. It will also follow from the next discussion about parity check matrices. □

Notes

Parity-check matrices

The dot product $x \cdot v$ can also be computed as:

$$x \cdot v = \begin{pmatrix} x_1 \cdots x_n \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

If the *columns* of an $n \times \dim(C^\perp)$ matrix H are a basis for $C^\perp$, then for any generator matrix G for C, $GH = 0$.

Definition

A **parity-check matrix** *for a linear code C is a matrix H whose columns form a basis for $C^\perp$.*

Note: $H = (1, \dots, 1)^t$ for the check-bit code we saw yesterday.

Notes

Algorithm for determining a parity-check matrix H

For linear codes over $\mathbb{F}_2$, there is a very nice process (Algorithm 2.5.7 in text) for determining a parity-check matrix:

- Start from a RREF generator matrix G for C.
- Let X be the $k \times (n-k)$ matrix obtained by deleting the columns containing the leading entries on each row.
- In the rows of H corresponding to the leading columns of G, place the rows of X, in order.
- In the other $n-k$ rows, place the rows of an $(n-k) \times (n-k)$ identity matrix, in order.
- Easiest case is if $G = (I_k|X)$. Then $H = \begin{pmatrix} X \\ I_{n-k} \end{pmatrix}$. (If the leading columns of G are not the first k columns, things get "mixed up" more.)

Notes

Example 1

Say $G = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$ from previous example. Then $G = (I_2|X)$, where $X = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. According to Algorithm 2.5.7,

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

It's easy to check that $GH = 0$. Moreover, from G, we can see that $C^\perp$ has dimension $\dim(C^\perp) = 3 = 5 - \dim(C)$.

Notes

Example 2

Now, say $G = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$ with leading entries in columns 1,3,4, and $X = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$. Then according to the algorithm

- We start with

$$\begin{pmatrix} 1 & 1 & 0 \\ & & \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ & & \\ & & \end{pmatrix}$$

Notes

Example 2, continued

- Then fill in with rows from a 3×3 identity matrix:

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

Again, can check that $GH = 0$. In this case, both C and $C^{\perp}$ have $n = 6$ and $k = 3$.

Notes

Syndrome decoding – an introduction

We saw last time that encoding could be done for linear codes using a generator matrix and matrix multiplication. This uses much less information (especially for codes with large n and k) than the methods which would apply for general (i.e. non-linear) codes. There we would essentially have to store a table of *all the code words* and look up the appropriate codeword for each $v \in \mathbb{F}_2^k$ each time we wanted to encode a word.

Today, we will see that there is a corresponding simplification of the process of *decoding* for linear codes, using a parity-check matrix.

Notes

Key property of H

By what we saw yesterday, if H is a parity-check matrix for a code C,

$$u \in C \Leftrightarrow uH = 0$$

Proof: $\Rightarrow$: The columns of H form a basis for $C^\perp$, so $uH = 0$, since $u \cdot h = 0$ for all columns h in H.
$\Leftarrow$: if $uH = 0$, then $u \cdot h = 0$ for all $h \in C^\perp$. So $u \in (C^\perp)^\perp$. Now, $C \subseteq (C^\perp)^\perp$, and

$$\dim(C^\perp)^\perp = n - (n - k) = k.$$

Hence $(C^\perp)^\perp = C$. $\Box$

Notes

Background on syndromes

- If a codeword u is sent and $u+e$ is received, then
$$(u+e)H = uH + eH = 0 + eH = eH$$
depends only on the error.
- We call $s = (u+e)H = eH$ the *error syndrome*
- The set of all words y with $yH = s$ for a given $s \in \mathbb{F}_2^{n-k}$ is $e + C = \{e + u \mid u \in C\}$ (a *coset* of the code C).
- (**Proof**: $e + C \subseteq \{y \mid yH = eH\}$ is clear from the above. On the other hand, if $yH = eH$, then $(y-e)H = 0$, so $y - e = u \in C$ and $y = e + u \in e + C$. $\square$)

Notes

- For a BSC, for instance, the most likely errors are the errors of smallest weight.
- This suggests that we should look for the *smallest-weight* vectors in each coset $e + C$.

- Proposition
Let $d(C) = 2t+1$ or $2t+2$. Then if there is a word of weight $\leq t$ in a coset $e+C$, that word is ***unique****.*
  - **Proof:** If $e+u$ and $e+v$ both have weight $\leq t$, then $\mathrm{wt}((e+u)+(e+v)) = \mathrm{wt}(u+v) \leq 2t$. But $u+v \in C$. Hence $u+v = 0$ so $e+u = e+v$. $\square$

Notes

The coset leader table and decoding

- The smallest-weight vector(s) in a coset $e + C$ are called the *coset leader(s)*.
- The coset leader table (or *standard decoding array*, SDA as in our text) is a table giving the coset leaders for each coset (or each possible syndrome).
- By the last proposition, if $d(C) = 2t + 1$ or $2t + 2$, and the error has weight $\leq t$, then the corresponding entry of the coset leader table will have just *one* entry.
- Syndrome decoding algorithm:
  - compute the syndrome $s = yH$ of the received word y,
  - look up the coset leader(s) in the table for the coset corresponding to s,
  - (if the leader is unique) subtract (or add – same thing in $\mathbb{F}_2$!) the coset leader from the received word.

Notes

Example – syndrome decoding

Consider the $[6, 3, 3]$ code C with $G = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$.

We have

$$C = \{000000, 100111, 010110, 001101, \\ 110001, 101010, 011011, 111100\}.$$

(Given what we have seen so far, we actually have to do this to see that $d = 3$ – note that the nonzero codewords of smallest weight have weight 3.)
Since $3 = 2 \cdot 1 + 1$, we have $t = 1$, and we know this code can correct any single bit error in a received word.

Notes

Syndrome decoding example, continued

- Now we construct the coset leader table (SDA) for C.
- Since G is in RREF, by Algorithm 2.5.7 from yesterday we have

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

- So, the syndromes $s = yH$ will be in $\mathbb{F}_2^3$, and there will be $2^3 = 8$ different possible syndromes.

Notes

Syndrome decoding example, continued

- The coset of the syndrome $s = 000$ is C itself (this is always the case)
- The coset of the syndrome $s = 100$ is the set of solutions of $yH = 100$, or

$$\begin{aligned} y_1 + y_2 + y_3 + y_4 &= 1 \\ y_1 + y_2 + y_5 &= 0 \\ y_1 + y_3 + y_6 &= 0 \end{aligned}$$

- (Shortcut!) Think of y_4, y_5, y_6 as the “basic variables” and y_1, y_2, y_3 as the free variables. This gives the coset

$$\{000100, 100011, 010010, 001001, 110101, 101110, 011111, 111000\}$$

$= 000100 + C$.

Notes

Syndrome decoding example, continued

Continuing in the same way, we get the coset leader table

syndrome	corresponding coset – leader(s) underlined
000	$\underline{000000}$, 100111, 010110, 001101, 110001, 101010, 011011, 111100
100	$\underline{000100}$, 100011, 010010, 001001, 110101, 101110, 011111, 111000
010	$\underline{000010}$, 100101, 010100, 001111, 110011, 101000, 011001, 111110
001	$\underline{000001}$, 100110, 010111, 001100, 110000, 101011, 011010, 111101
110	000110, 100001, $\underline{010000}$, 001011, 110111, 101100, 011101, 111010
101	000101, 100010, 010011, $\underline{001000}$, 110100, 101111, 011110, 111001
011	$\underline{000011}$, $\underline{100100}$, 010101, 001110, 110010, 101001, $\underline{011000}$, 111111
111	000111, $\underline{100000}$, 010001, 001010, 110110, 101101, 011100, 111011

Note: Each word of length 6 occurs exactly once in the table.
Only the coset of $s = 011$ fails to have a unique coset leader.

Also: Only the coset leaders need to be stored (not the whole cosets).

Notes

Syndrome decoding example, continued

- Say the word $y = 101111$ is received.
- Start by computing

$$s = yH = (1,0,1,1,1,1)\begin{pmatrix}1&1&1\\1&1&0\\1&0&1\\1&0&0\\0&1&0\\0&0&1\end{pmatrix} = (1,0,1).$$

- From the coset leader table, the corresponding coset has leader $\ell = 001000$.
- So we decode to $y + \ell = 101111 + 001000 = 100111$. Can see this is a codeword!

Notes

Syndrome decoding example, continued

Question: What would happen if we received a word like 001110?

We have $s = (0,0,1,1,1,0)\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 0\\ 1 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} = (0,1,1)$

This is the case where there is not a unique coset leader. What would the decoder need to do in this case?

Notes

Syndrome decoding example, concluded

- The main “down-side” of this method may not be apparent. Realistic codes might have n, k in the 100s or 1000s.
- The coset leader table has at least 2^{n-k} words of length n. Something like $2^{500} \simeq 10^{50}$ is *really huge!*
- However, the syndromes correspond to the integers $0, 1, \dots, 2^{n-k} - 1$, so they come with a natural *ordering*. We could use a variant of the *binary search*, in which the length of the list to be searched is cut down by a factor of 2 each time.
- No more than $n - k$ comparisons of the syndrome with a key to the coset leader table would be needed.
- To summarize – big list of coset leaders, but efficient searching methods are possible!

Notes

Introduction – bounds on codes

We have seen a few small examples of codes at this point, but none of them would be especially attractive for applications. Since the larger d is, the more bit errors a code can detect and correct, it is natural to ask:

Given an n and a k, what is the best possible d for a linear code over $\mathbb{F}_2$?

Today, we want to develop several *bounds* on the parameters of codes that (even if they do not guarantee the existence of codes), at least give some idea of what is possible.

Notes

Richard Hamming

Our first bound was developed by Richard Hamming, who

- worked on the Manhattan Project that developed the atomic bomb in WWII,
- later worked at Bell Labs from 1946 to 1976 on various problems in communications
- at Bell Labs, he was a colleague of Claude Shannon, (the "father of coding theory").
- Hamming died in 1998 after receiving numerous honors for his pioneering work.
- We already saw Hamming's name in the *Hamming distance* on $\mathbb{F}_2^n$.

Notes

The Hamming balls, again

- Recall from our Theorem on Monday that if $d = 2t + 1$ or $2t + 2$ for a code, then the closed balls of radius t around distinct codewords must be disjoint:

$$u \neq v \in C \Rightarrow B(u,t) \cap B(v,t) = \emptyset.$$

- The number of words in one such ball is given by:

Proposition

Let $u \in \mathbb{F}_2^n$. Then $|B(u,t)| = \sum_{i=0}^{t} \binom{n}{i}$.

- **Proof:** The words in $B(u,t)$ are obtained from u by "flipping" at most t of the bits in u. There are $\binom{n}{i}$ ways to select a collection of i bits to "flip." □

Notes

The Hamming (sphere-packing) bound

If $d = 2t + 1$ or $2t + 2$, then $B(u,t)$ for $u \in C$ are pairwise disjoint, but their union also must "fit" inside $\mathbb{F}_2^n$. Hence, we have proved:

Theorem (Hamming bound)

Let C be a code with blocklength n and $d = 2t + 1$ or $2t + 2$. Then

$$|C| \leq \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}.$$

(This holds for non-linear codes as well, since we have not used linearity properties in any way.)

Notes

Deductions from the Hamming bound

Question: What is the largest number of codewords a code with $n = 7$ and $d = 5$ can have?

- We have $5 = 2 \cdot 2 + 1$, so $t = 2$. Hence $\sum_{i=0}^{2} \binom{7}{i} = 1 + 7 + 21 = 29$.
- We have $|C| \leq \frac{2^7}{29} \doteq 4.4$.
- Since $|C|$ must be an integer, this says $|C| \leq 4$.
- **Caution:** This does not show that such a code exists or how to find one. *Challenge:* Can you find one? It is relatively easy to see that *C cannot be linear*.

Notes

Another example

Question: What is the largest d possible for a linear code with $n = 23$ and $k = 11$?

- We must have $|C| = 2^{11}$, so $2^{11} \leq \frac{2^{23}}{\sum_{i=0}^{t} \binom{23}{i}}$.
- So $\sum_{i=0}^{t} \binom{23}{i} \leq 2^{12} = 4096$.
- $\sum_{i=0}^{3} \binom{23}{i} = 2048$, but $\sum_{i=0}^{4} \binom{23}{i} = 10903$.
- So, t is at most 3, and $d \leq 2 \cdot 3 + 2 = 8$.
- In this case, $[23, 11, 8]$ codes do exist (from an online table of best-known codes that we will see next week!)

Notes

The Singleton bound

- Our next result applies only for linear codes.

- Theorem (Singleton bound)
  Let C be a linear code with parameters $[n, k, d]$. Then

$$d \leq n - k + 1.$$

- Codes meeting this bound are called *MDS* (maximum distance separable) codes

Notes

Proof of the Singleton bound

Proof:

- The words of C are the scalars in linear combinations of the *rows* of a parity-check matrix that add up to 0.
- Hence, if C has $d(C) = d$, then *all sets of $d - 1$ rows of H are linearly independent, but some set of d rows of H is linearly dependent.*
- H is an $n \times (n - k)$ matrix that has rank $n - k$. Hence by the above, $d - 1 \leq n - k$. $\square$
- This proof shows that C is MDS if and only if *all* sets of $n - k$ rows of H are linearly independent.

Notes

Minimum distance via parity-check matrix

The observation relating d and the rows of the parity-check matrix is sufficiently important to rate a separate statement.

Theorem

Let H be an $n \times (n-k)$ parity check matrix for a linear code C. If H has the property that all collections of $d-1$ rows are linearly independent, but some set of d rows is linearly dependent, then C has minimum distance d.

Notes

Comments on Singleton

- The Singleton bound can also be rearranged to say that for a linear code $k \leq n - d + 1$.
- The Singleton bound is often much less “tight” than the Hamming bound.
- For example, consider the case $n = 23$, $k = 11$ that we saw before. Hamming $\Rightarrow d \leq 8$.
- But Singleton only gives $d \leq 23 - 11 + 1 = 13$. (True, but not too close to the best d that is actually possible!)
- On the other hand, there are situations (especially for codes over larger alphabets), where MDS codes (codes with $d = n - k + 1$ actually do exist!

Notes

The Gilbert-Varshamov bound

- The Hamming and Singleton bounds both give *upper bounds* on $|C|$ (or k for linear codes).
- Our next result gives a *lower bound* that says, in effect, the best possible code is *at least as good as the bound*.

- Theorem (Gilbert-Varshamov Bound)
There exists an $[n,k,d]$ linear code for any combination of parameters satisfying

$$2^{n-k} > \sum_{i=0}^{d-2} \binom{n-1}{i}.$$

Notes

Proof of Gilbert-Varshamov

Proof:

- The proof consists of showing that we can construct a parity-check matrix H for a code of minimum distance d under this hypothesis.
- By the above, we need to construct an $n \times (n-k)$ matrix with the property that all collections of $d-1$ rows are linearly independent.
- The construction will be *inductive*. Start by picking any nonzero first row.
- Now assume that we have constructed the first i rows of H; call the submatrix H_i.

Notes

Proof of Gilbert-Varshamov, continued

- Given the i rows of H_i, there are at most

$$\binom{i}{0} + \binom{i}{1} + \cdots + \binom{i}{d-2}$$

vectors in $\mathbb{F}_2^{n-k}$ that are linear combinations of the rows of H_i.

- On the other hand, by hypothesis,

$$\begin{aligned} 2^{n-k} &> \binom{n-1}{0} + \binom{n-1}{1} + \cdots + \binom{n-1}{d-2} \\ &\geq \binom{i}{0} + \binom{i}{1} + \cdots + \binom{i}{d-2} \end{aligned}$$

Notes

Proof of Gilbert-Varshamov, concluded

- Hence, there are always vectors in $\mathbb{F}_2^{n-k}$ *left over!*
- We can choose any one of them as the next row for H while retaining the property that all collections of $d-1$ rows are linearly independent.
- Continue in this way until H has n rows. $\square$

Notes

Example

Question: Does there exist a linear $[15, 6, 5]$ code over $\mathbb{F}_2$?

- By Gilbert-Varshamov, $2^{15-6} = 2^9 = 512$.
- $\sum_{i=0}^{3} \binom{14}{i} = 1 + 14 + 91 + 364 = 470$.
- Since, $512 > 470$, the answer is *yes*!
- The process of the proof of Gilbert-Varshamov could be used to construct a parity-check matrix H for such a code.

Notes

Perfect codes

Definition

Codes for which the Hamming bound is achieved:

$$|C| = \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}}$$

are known as **perfect codes**.

We will conclude today by considering a famous family of perfect codes, also found by Richard Hamming.

Notes

Hamming codes

- Let H be a parity-check matrix whose rows consist of all $2^r - 1$ *nonzero* vectors in $\mathbb{F}_2^r$ (in some order).
- For instance, if $r = 3$ we could take

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

- Note that by construction all sets of 2 rows of H are linearly independent (since they are distinct).

Notes

Hamming codes, continued

- By our theorem on d and the parity-check matrix, the corresponding code has $d = 3$.
- The other parameters are $n = 2^r - 1$, and $k = 2^r - r - 1$.
- Hamming codes are perfect codes since $t = 1$, so:

$$|C| = 2^{2^r - r - 1} = \frac{2^{2^r - 1}}{1 + 2^r - 1}.$$

- The integer $r \geq 1$ is arbitary, so we have an infinite family of perfect codes with parameters $[2^r - 1, 2^r - r - 1, 3]$.