MTH2002
Coding Theory

Semester A, 2024/25

Yuri Santos Rego

University of Lincoln

UNIVERSITY OF LINCOLN
MATHEMATICS

Week 2, Lecture 2

Today

Week 2: Goals

Week 2

1. Lecture 1: Geometry of the Hamming distance, and the Distance Theorem.
2. Lecture 2: Chances of correct decoding, parameters, and the Main Problem.

Last time

Recalling:

Distance Theorem (22)

Let C be a code with minimal distance $d_{\min}(C)$. Then the following statements hold:

1. If $t \in \mathbb{N}$ and $d_{\min}(C) \geq t + 1$, then C detects t errors.
2. If $k \in \mathbb{N}$ and $d_{\min}(C) \geq 2k + 1$, then C corrects k errors.

Corollary (27) to the Distance Theorem

Let C be a code and write $\mathbf{d}$ for $d_{\min}(C)$. Then C can detect up to $d - 1$ errors and correct up to $\lfloor \frac{d-1}{2} \rfloor$ errors, where $\lfloor \ \rfloor : \mathbb{R} \to \mathbb{Z}$ denotes the **integer part function**: $\lfloor r \rfloor = \max\{z \in \mathbb{Z} \mid z \leq r\}$.

Remarks: 'decoding' correctly

What is decoding?

Let $C \subseteq A^n$ be a code in a space of words A^n (of length n) over an alphabet A.

Decoding

For the purposes of our module, a decoding process means choosing a valid codeword $c \in C$ given a word $w \in A^n$ that we have received upon transmission of a valid codeword $c_0 \in C$.

(The chosen word $c \in C$ is called the **decoded word** for the sent word $c_0 \in C$.)

In case the word received w **is** a **valid codeword**, then the choice is always $c = w$.

What is decoding?

Example

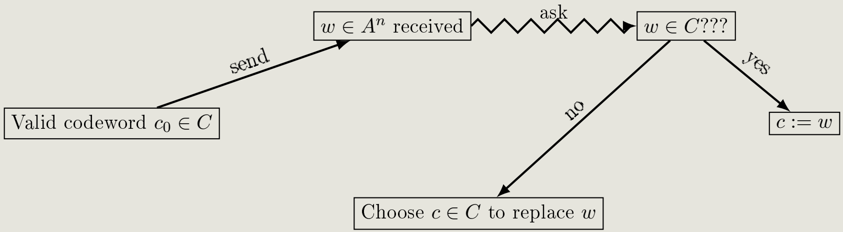
$A = \{0, 1\}, n = 3, C = \{000, 111, 101\}$.

- $c_0 = 111 \in C$ sent, $w = 011 \in A^n$ received (not valid!) $\rightsquigarrow$ decoding means choosing $c \in C$ to replace w, for example $c = 111$ by **nearest neighbour decoding strategy**;
- $c_0 = 111 \in C$ sent, $w = 101 \in A^n$ received $\rightsquigarrow$ decoding means choosing $c = w \in C$ because w is valid in ths case.

What is decoding?

$C \subseteq A^n$ a code in a space of words A^n (of length n) over an alphabet A.

Schematically



In particular: **incorrect decoding** happens if **undetected** transmission errors occur.

Chance of incorrect decoding

Notation

Given a code $C \subseteq A^n$ and a word $w \in A^n$, denote by

- $P_{\text{corr.dec.}}(w)$ the probability of correctly decoding w,
- $P_{\text{incorr.dec.}}(w)$ the probability of **in**correctly decoding w. (Thus $P_{\text{incorr.dec.}}(w) = 1 - P_{\text{corr.dec.}}(w)$ by principles of probability!)
- $P_{\text{undetect}}(w)$ the probability of not detecting errors when w is transmitted.

Natural goal: Design codes C for which $P_{\text{incorr.dec.}}(w)$ and $P_{\text{undetect}}(w)$ are as low as possible for all words $w \in A^n$.

Issue: $P_{\text{incorr.dec.}}(w)$ might not be so straightforward to compute!

Example

$A = \mathbb{F}_2 = \{0, 1\}$, $n = 5$, $C = \{00000, 11111\}$, with transmission in a symmetric channel with symbol error probability $p = 0.004$.

Say $c \in C$ is sent and $w \in \mathbb{F}_2^5$ is received. Out of the five symbols of w, the following can happen:

(I) all five are correct (i.e., $w = c$), with probability $(1-p)^5$;

(II) one is wrong, in which case this can happen in $\binom{5}{1}$ ways, each with chance $p(1-p)^4$;

(III) two are wrong, which can happen in $\binom{5}{2}$ ways, each with chance $p^2(1-p)^3$;

Example

(IV) three are wrong, which can happen in $\binom{5}{3}$ ways, each with chance $p^3(1-p)^2$;

(V) four are wrong, which can happen in $\binom{5}{4}$ ways, each with chance $p^4(1-p)$;

(VI) all five are wrong, which can happen with probability p^5.

But looking at $C = \{00000, 11111\}$, the **Distance Theorem** tells us that C corrects up to 2 errors — thus, in the first three cases, we can safely choose the correct substitute for w.

In other words: we **correctly** decode w if case (I), (II) or (III) occurs.

Chance of incorrect decoding

Example

So: we correctly decode w if case (I), (II) or (III) occurs.

The probability of (I) happening is $(1-p)^5$, the probability of (II) happening is $\binom{5}{1} \cdot p \cdot (1-p)^4 = 5p(1-p)^4$, and the probability of (III) happening is $\binom{5}{2} \cdot p^2 \cdot (1-p)^3 = 10p^2(1-p)^3$.

Adding it all up and recalling $p = 0.004$, the chance of **correctly** decoding w is
$P_{\text{corr.dec.}}(w) = (1-p)^5 + 5p(1-p)^4 + 10p^2(1-p)^3 \approx 0.999999364$,

so the chance of **<u>in</u>correctly** decoding w is
$P_{\text{incorr.dec.}}(w) = 1 - P_{\text{corr.dec.}}(w) \approx 0.000000636$.

Chance of incorrect decoding

- Remark: In cases (IV) and (V) — i.e., three or four errors — we know errors occurred but could not decode correctly with certainty. (Maybe could ask for retransmission.)
- Note how the *Distance Theorem* (thus the **minimal distance** $d_{\min}(C)$) came into play in our considerations.
- The previous example reinforces the following natural observation:

  The higher the minimal distance is for a code, the better!

  (Concretely: smaller probability of incorrect decoding.)

Parameters and Main Problem

Parameters of a code

Motivated by the previous examples and lectures, we look at numerical attributes of codes that reveal information about its efficiency.

Definition 30 (Parameters of a code)

Let n, M, d and q be natural numbers. A code C called an $(n, M, d)_q$-code when

- the underlying alphabet used for C has q symbols,
- each codeword in C has length n,
- C itself has M codewords in total (i.e., $M = \#C$), and
- d is its minimal distance (i.e., $d = d_{\min}(C)$).

The numbers n, M, d and q are called *parameters* of C.

If the number of symbols q is not important in the context, we sometimes write just (n, M, d) and say that C is an (n, M, d)-code.

Example 31

- $C_3 = \{00000,\ 01101,\ 10110,\ 11011\}$ from the first lecture is a $(5,4,3)_2$-code.
- $C = \{00000,\ 11111\}$ from the previous example is a $(5,2,5)_2$-code.
- If $C = \mathbb{F}_3^2 = \{00,\ 01,\ 10,\ 11,\ 02,\ 20,\ 22,\ 21,\ 12\}$ (from Example 19), then C is a $(2,9,1)_3$-code.

Example 32 (Repetition codes)

Recall *repetition codes* from the practicals: say C is formed from a q-ary alphabet A by considering messages of length m and repeating each of these m symbols k times to their left.

The resulting repetition code C is a $(m \cdot k, q^m, k)_q$-code.

[For instance, $A = \mathbb{F}_3 = \{0, 1, 2\}$, initial messages of length 2, and repeat symbols twice (so $k = 2$).

Thus there are 9 possible initial messages, namely {00, 01, 10, 11, 02, 20, 22, 21, 12}, and the resulting repetition code C is given by
$C = \{0000, 0011, 1100, 1111, 0022, 2200, 2222, 2211, 1122\} \subset \mathbb{F}_3^4$.

This is a $(4, 9, 2)_3$-code.]

What makes up an 'efficient' code?

From our investigations so far: A 'good' code should...

- ... minimise the chances of incorrectly decoding words, which can be done by increasing its error detection and error correction capabilities. By the **Distance Theorem**, this means that its **minimal distance should be large**;
- ... **have** relatively **small length**, because the less symbols we have to transmit, the faster is the transmission;
- ... **have** relatively **large size** (that is, total number of codewords), because this means we can transmit a wide variety of messages.

Main Issue of Code Design

The second and third requirements above are conflicting! The smaller the length, the smaller the possible codewords we can form!!! D-:

What makes up an 'efficient' code?

A 'good' $(n, M, d)_q$-code should...

- ... **have small** d (to detect and correct many errors);
- ... **have** relatively **small** n (to speed up transmission);
- ... **have** relatively **large** M (to permit wide variety of messages).

Main Problem of Coding Theory

Given a q-ary alphabet, a length n, and a desired minimal distance d, design an $(n, M, d)_q$-code for which its total number of codewords M **is as large as possible**.

Notation

Given q, n, and d as above, we write $M_q(n, d)$ for the largest possible such M. In other words, the Main Problem of Coding Theory asks us to find $M_q(n, d)$ once the parameters q, n and d have been given.

Examples of $M_q(n, d)$

Recall: $M_q(n, d)$ = largest possible number of codewords in a code of length n with minimum distance d and over a q-ary alphabet.

Proposition 33

$M_q(n, 1) = q^n$.

Proof.

Let A be a q-ary alphabet.
Suppose C is some arbitrary code with $\#C = M_q(n, 1)$. Then — by definition — $C \subseteq A^n$, hence $M_q(n, 1) = \#C \leq q^n$.
On the other hand, we can design a code with $M_q(n, 1) \geq q^n$: simply take C to be A^n, in which case $d_{\min}(C) = d_{\min}(A^n) = 1$.
Therefore $M_q(n, 1) \geq q^n$.
Combining the results: $q^n \leq M_q(n, 1) \leq q^n$, i.e., $M_q(n, 1) = q^n$. $\square$

Examples of $M_q(n,d)$

Recall: $M_q(n,d)$ = largest possible number of codewords in a code of length n with minimum distance d and over a q-ary alphabet.

Proposition 34

$M_q(n,n) = q$.

Proof.

Let A be a q-ary alphabet.
Suppose C is some arbitrary code with $\#C = M_q(n,n)$. Since the length n agrees with the minimal distance n, any two codewords of C differ in **every** entry. In particular, the first entry of each codeword is distinct and so $M_q(n,n) = \#C \leq q$.
Conversely, design a code with $M_q(n,n) \geq q$: take C as the repetition code from messages of length 1 over A repeating the symbols n times — this gives an $(n,q,n)_q$-code as seen in Example 32. Thus $M_q(n,n) \geq q$. $\square$

Next time...

- Symmetries of codes;
- The Sphere Packing Bound;
- A redesign strategy: parity check.

I wish you a great weekend!

34