Course name: Coding Theory (Timing: T F, 9:30 - 11:15 AM).

Course Structure:

- A very brief overview: A robust way of storing information s.t. it is resilient to noise/errors.
  - wide practical applications (see below)
  - Immense theoretical importance, specifically in the area of algebra, combinatorics, information theory & computer science.

  we many "nontrivial" applications of basic facts from abstract & linear algebra, combinatorics, probability.

Overall, the purpose of the course is to provide an introduction to coding theory with an emphasis on its application in theoretical computer science.

Tentative topics:

- Basics of codes: Basics of linear codes
- Various bounds related to the existence of codes
- Construction of explicit codes with encoder & decoder

  for example: RS-codes, Expander codes, Reed-Muller codes etc.

- Code-concatenation
- List decoding with explicit constructions.
- Local-decoding with explicit constructions.
- Few more topics based on availability of time.

Prerequisite:

- Familiarity with basic algebra stuff related to groups, finite fields, linear algebra, the contents of typical undergrad Discrete Math & Algo class would be helpful.
- T1 ⇒ an inquisitive mathematical mind is sufficient
- T(1&2) ⇒ a strong interest with hardwork is also sufficient; we cover all the necessary details.

Grading Scheme:

- Endsem: 50%
- Midsem: 20%
- Assignment + Scribing lecture notes: 15% + 15%
  [Time for submitting 1st draft: Tuesday class by next Tuesday & Friday by next Friday
  & Also, need to make additional changes suggested by instructor]

References :

1. Venkatesan Guruswami, Atri Rudra, Madu Sudan: Essential Coding Theory (Main Text book).
2. Venkatesan Guruswami: 15-859Y: Coding Theory [Course offered in CMU .
3. Prahlad Harsha & Mrinal Kumar: CSS.318.1: Coding Theory [Course offered in TIFR]
4. Swastik Kopparty: 198:540: Error-Correcting Codes [Course offered in Rutgers]
5. Marry Wotters: CS250/EE387: Algebraic Error Correcting Codes [course offered in Standford]

Instructor email: besusumanta@gmail.com

TA : Ayanava Dasgupta (ayanavadasgupta_r@isical.ac.in).

Google Classroom Code: ofl4jjaj

22nd July, 2025: Basics of Codes & fundamental Question:

Goal: To understand the fundamental question in coding theory!

First we give an informal answer via following two examples:

Example:

You: What are you doing?

Friend: I bm blaying foetboll.

Can you decode what your friend's message?

[Error but get the message!]

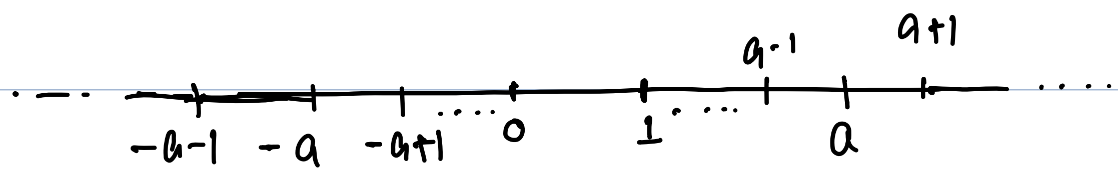
Why are we able to decode it correctly?

Reason:
- English as a language has some inherent redundancy (semantically & syntactically).
- Valid english messages are most of the time quite far from each other (semantically & syntactically).

The above two implies if errors happes in some restricted way, we can unique‌ly deode the message.

More formal Example: (Taken from a talk given by Swastik Koppatry):

- pick an integer $a \in \mathbb{N}$
- pick a number $x \in [-0.1, 0.1]$
- Give $a+x$.
- **Goal:** output a from $a+x$
- **Answer:** round$(a+x)$ / $[a+x]$ / $\lfloor a+x \rceil$: nearest integer function. (Explain)



Even if $x \in (-0.5, 0.5)$, the above approach works (Explain)

If $|x| \geq 0.5$. then we cannot find a from $a+x$.

Abstract property: "Relevant numbers" are only integers among all real numbers (Redundancy) & they are "well separated" from each other.

This exactly capture the essense of coding theory!

Error correcting codes (simply, codes): are clever ways of representing data so that one recover the original message even if parts of it are corrupted;

Examples:

- QR-code scan.
- Transmitting (data) packets over internet:

dropped

[handled by a combination of sequence numbers & ACKs]

corruption

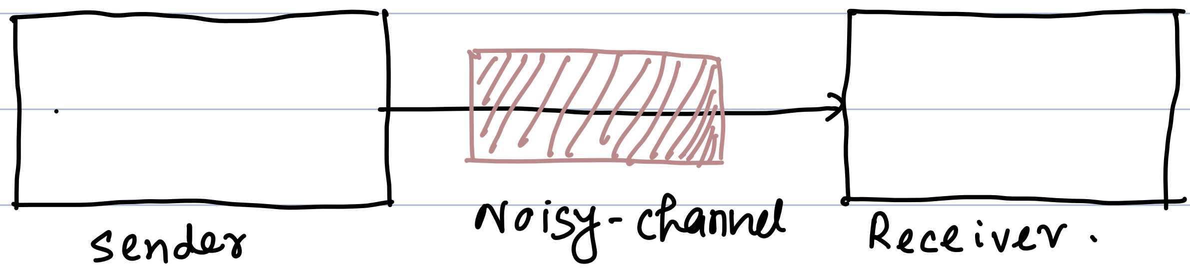
[handled by checksum]

(good in detection but poor in correction)

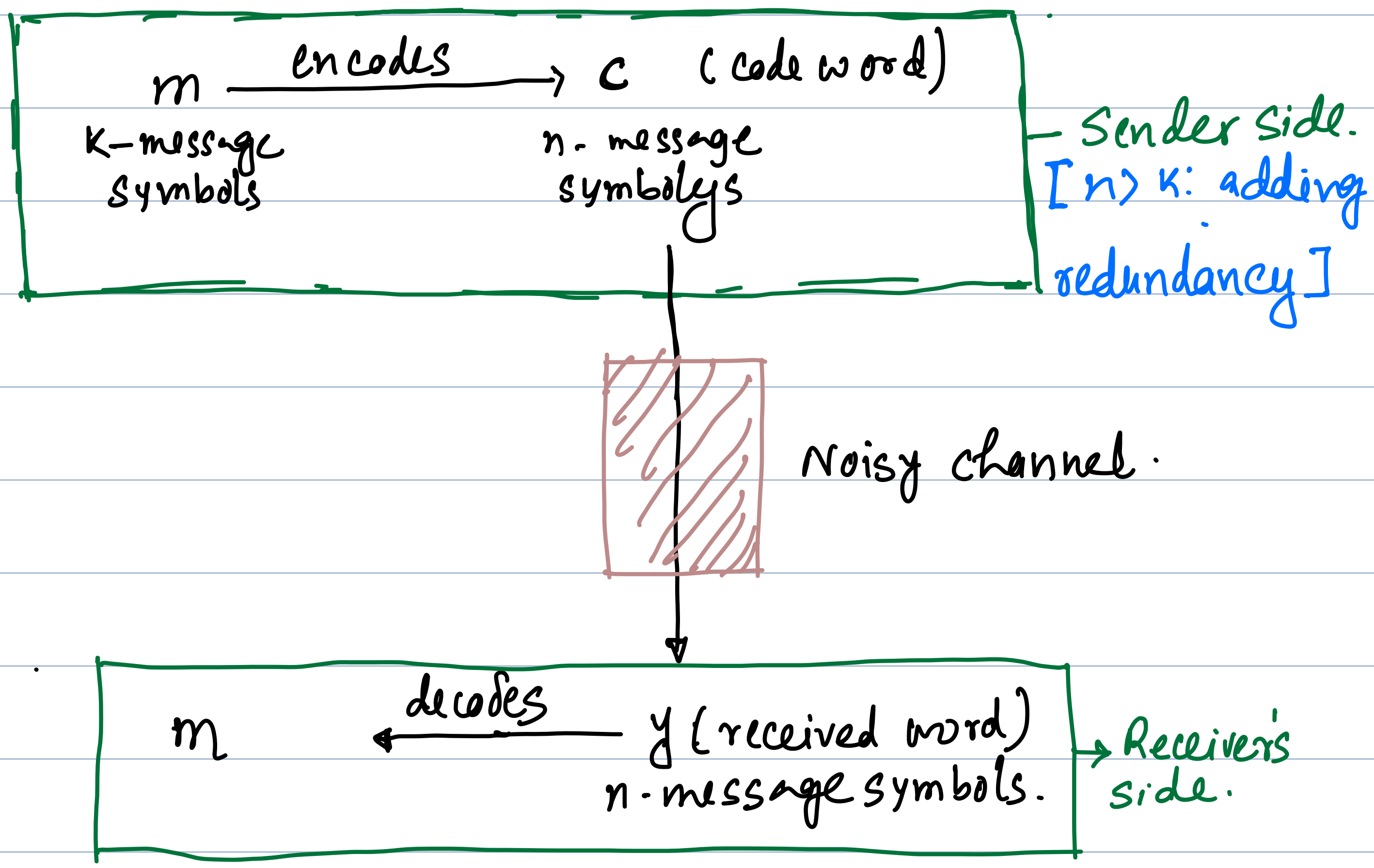
[works because retransmission allowed]

- When Retransmission is not allowed: cell phone calling, space communication. satellite broadcast;
- Error correction in non-communication set up:
  1. Data storage in CD, DVDs, hard disk;
  2. Banking system.

Error-Correction Set up:



Goal: sending messg $m = (m_1, \ldots, m_k)$ from Sender to receiver.



Question 1. (Trade-off b/w redundancy & error-correction)

How much redundancy do we need to correct a given amount of error?

Maximizing error correction & minimizing redundancy are contradictory goals:

Aim: Formalizing this question!

Step 1: determine the optimal trade off!

Step 2: Constructing codes that can achieve it!

Definitions:

- **Code:** Alphabet: a finite subset Σ, $q = |\Sigma|$

A code of block length n over an alphabet Σ is a subset of Σ^n, where.

Σ^n
- → a set of sequences $(v_1, v_2, \ldots, v_n)$ where $v_i \in \Sigma$
- → a set of vector tuples: helpful when Σ has some structure, in particular, field.
- → set of functions $f: [n] \to \Sigma$; helpful when coordinates has some structure
  e.g. $[n]$ can a finite field of size n or group of size n.

Alternate view: For any code $C \subseteq \Sigma^n$ with $|C| = M$, think C is a mapping of form:

$$C : [M] \to \Sigma^n.$$

where $[M] = \{1, 2, \ldots, M\}$.

Dimension of Code: $\quad k = \log_q |C|.$

Examples:

$\Sigma = \{0,1\}$ (i.e., binary codes)

- **Parity Code:** $C_\oplus : \underset{\in \{0,1\}^4}{(x_1, x_2, x_3, x_4)} \mapsto (x_1, x_2, x_3, x_4, x_1 \oplus \cdots \oplus x_4)$

  $\oplus$: XOR operation $:=$ addition modulo 2.

- **Repetition Code:** repeat every message bit to a fix no. of times.

  $C_{3,rep} : (x_1, x_2, x_3, x_4) \mapsto (x_1, x_1, x_1, x_2, x_2, x_2, \ldots, x_4, x_4, x_4)$

In both the examples, $|C| = 2^4 = 16$;

dimension: 4;

messg space: $\{0,1\}^4$

Formalizing Redundancy: C is a code over Σ.

- block length: n, i.e. $C \subseteq \Sigma^n$
- dimension: $k = \log_q |C|$ where $q = |\Sigma|$

One natural way: $n - k$

- **Problem:** Consider two codes:

$k = 2$, $n = 4$ $\qquad$ $k = 100$, $n = 102$

Cannot distinguish these two case.

first code uses 1 bits of redundancy, whereas the second code uses 0.02 bits of redundancy per message bit. That is, in the relative sense, first code uses more redundancy compare to the second code.

Definition (Rate of a Code): $R = \frac{k}{n}$

- avg amount of information in each of the n symbols transmitted over the channel;
- Higher rate $\Longrightarrow$ lesser redundancy; Complement of redundancy (Use $C_{\oplus}$, $C_{rep,3}$ as examples)