6 Error-Correcting Codes

17 November 2023

Sebastian Wild

COMP526 (Fall 2023)
University of Liverpool
version **2023-11-13 16:46**

Learning Outcomes

1. Understand the context of *error-prone communication*.
2. Understand concepts of *error-detecting codes* and *error-correcting codes*.
3. Know and understand the *terminology of block codes*.
4. Know and understand *Hamming codes*, in particular (7,4) Hamming code.
5. Reason about the *suitability of a code* for an application.

Unit 6: *Error-Correcting Codes*

Outline

6 Error-Correcting Codes

6.1 Introduction

Noisy Communication

- most forms of communication are “noisy”
  - humans: acoustic noise, unclear pronunciation, misunderstanding, foreign languages

Noisy Communication

- most forms of communication are “noisy”
  - humans: acoustic noise, unclear pronunciation, misunderstanding, foreign languages

- How do humans cope with that?
  - slow down and/or speak up
  - ask to repeat if necessary

Noisy Communication

- most forms of communication are “noisy”
  - humans: acoustic noise, unclear pronunciation, misunderstanding, foreign languages

- How do humans cope with that?
  - slow down and/or speak up
  - ask to repeat if necessary

- But how is it possible (for us) to decode a message in the presence of noise & errors?

Bcaesue it semes taht ntaurul lanaguge has a lots fo ***redundancy*** *bilt itno it!*

Noisy Communication

- most forms of communication are “noisy”
  - humans: acoustic noise, unclear pronunciation, misunderstanding, foreign languages

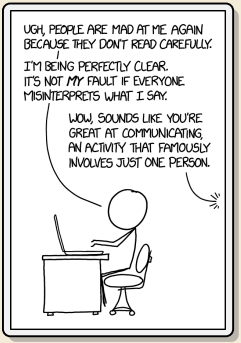
- How do humans cope with that?
  - slow down and/or speak up
  - ask to repeat if necessary

- But how is it possible (for us) to decode a message in the presence of noise & errors?

Bcaesue it semes taht ntaurul lanaguge has a lots fo ***redundancy*** *bilt itno it!*

⇝ We can

1. **detect errors** “This sentence has aao pi dgsdho gioasghds.”
2. **correct** (some) **errors** “Tiny errs ar corrrected automaticly.”
   (sometimes too eagerly as in the Chinese Whispers / Telephone)



Noisy Channels

- computers: copper cables & electromagnetic interference
- transmit a binary string
- but occasionally bits can "flip"

⇝ want a robust code

Noisy Channels

- computers: copper cables & electromagnetic interference
- transmit a binary string
- but occasionally bits can "flip"

⇝ want a robust code

- We can aim at
  1. **error detection** ⇝ can request a re-transmit
  2. **error correction** ⇝ avoid re-transmit for common types of errors

Noisy Channels

- computers: copper cables & electromagnetic interference
- transmit a binary string
- but occasionally bits can "flip"

⇝ want a robust code

- We can aim at
  1. **error detection** ⇝ can request a re-transmit
  2. **error correction** ⇝ avoid re-transmit for common types of errors
- This will require *redundancy*: sending *more* bits than plain message
  - ⇝ **goal:** robust code with lowest redundancy

that's the opposite of compression!

Clicker Question

What do you think, how many extra bits do we need to **detect** a **single bit error** in a message of 100 bits?

→ *sli.do/comp526*

Clicker Question

What do you think, how many extra bits do we need to **correct** a **single bit error** in a message of 100 bits?

→ *sli.do/comp526*