# COSC 290 Discrete Structures

## Lecture 9: Error-correcting codes

Prof. Michael Hay
Monday, Sep. 18, 2017
Colgate University

## Plan for today

1. Problem Set
2. Error-correcting codes
3. A more abstract view of codes



# Problem Set

## Problem Set 6

Let's go over 3 problems from the problem set. Which three should we do?

## Error-correcting codes

## Basic setup

- Sender wants to transmit message $m \in \{0, 1\}^k$
- Message $m$ encoded as a n-bit codeword $c \in \mathcal{C} \subseteq \{0, 1\}^n$
- Codeword $c$ is transmitted over a noisy channel which may corrupt message.
- Receiver receives $c'$, a (possibly corrupted) n-bit string.
- Reciever decodes $c'$ into message $m'$



## Error detection and correction

Instead of sending $k$-bit message directly, a *larger* n-bit codeword is sent.

The goal: design an encoding scheme with these properties...

- Error Correction if a "small" number of bits are corrupted, the receiver correct those bits and recover message $m$
- Error Detection if a "medium" number of bits are corrupted, the receiver can at least detect corruption (and, say, request retransmission)



## Applications

- Digital storage (Reed-Solomon codes and CDs/DVDs)
- Internet
- Deep-space telecommunications
- Related ideas are used to verify transactions in Bitcoin (blockchain)

## Performance measures and Goals

**Performance measures**:

- Error tolerance: the maximum number of bits that can be corrupted yet receiver can still recover original message.
- Rate: ratio between message length and codeword length, $k/n$.

**Goals**: high error tolerance, high rate



## Example: repetition code

A size $\ell$ repetition code takes message $m$, and sends $\ell$ copies of $m$.

Example:

- Suppose $m \in \{0, 1\}^2$ and $\ell = 3$.
- If message $m = 10$ then $c = 10\ 10\ 10$.
- Suppose the receiver gets $c' = 10\ 10\ 11$,
  - can the receiver detect an error? how?
  - correct an error? how?



## Poll: repetition code

A size $\ell$ repetition code takes message $m$, and sends $\ell$ copies of $m$.

How many errors can a size $\ell$ repetition code tolerate? What is its rate?

[A) It can tolerate 1 error, and its rate is $1/k$

[B) It can tolerate $\ell$ errors, and its rate is $1/k$

[C) It can tolerate 1 error, and its rate is $1/\ell$

[D) It can tolerate $\ell$ errors, and its rate is $1/\ell$

[E) It can tolerate $\ell$ errors, and its rate is $k/\ell$



## A more abstract view of codes

## Error correcting codes: an abstract view

A code is a set $\mathcal{C} \subseteq \{0,1\}^n$ where $|\mathcal{C}| = 2^k$.

- Encoding: a bijective function $encode : \{0,1\}^k \to \mathcal{C}$ maps $k$-bit messages to codewords in $\mathcal{C}$. Both the sender and receiver know this function.
- Error detection: a function $hasError(c')$ returns True if $c' \notin \mathcal{C}$ and False otherwise.
- Error correction: choose $c \in \mathcal{C}$ that is *closest* to $c'$ and then applies the inverse of *encode*.

Question: if $hasError(c')$ returns False, does this mean *no error* has occurred?



## Distance measure for bit strings

Let $x, y \in \{0,1\}^n$ be two $n$-bit strings. The Hamming distance between $x$ and $y$, denoted $\Delta(x,y)$, is the number of positions in which $x$ and $y$ differ.

$$\Delta(x,y) := |\{ i \in 1,2,\ldots,n : x_i \neq y_i \}|$$

Example:

- $x = 1000011$
- $y = 1\underline{1}000\underline{0}1$
- $\Delta(x,y) = 2$



## Example

Here is an example code where $\mathcal{C}\,\{100111, 101010, 010110, 010111\}$. Since $|\mathcal{C}| = 2^2$, we can use code to send 2-bit messages.

| $m$ | $c = encode(m)$ |
|---|---|
| 00 | 10 01 11 |
| 01 | 10 10 10 |
| 10 | 01 01 10 |
| 11 | 01 01 11 |



## Poll: minimum distance

Suppose the receiver gets $c' = 10\ 10\ 11$, can the receiver detect an error? If so, can receiver correct the error?

A) The receiver can never be 100% certain there was an error.

B) The receiver knows there's an error, but cannot correct it.

C) The receiver knows there's an error, and would correct it to be 10 01 11.

D) The receiver knows there's an error, and would correct it to be 10 10 10.

E) None of the above / more than one of the above.

| $m$ | $c = encode(m)$ |
|---|---|
| 00 | 10 01 11 |
| 01 | 10 10 10 |
| 10 | 01 01 10 |
| 11 | 01 01 11 |

## Minimum Distance

The minimum distance of code $\mathcal{C}$ is the smallest Hamming distance between two distinct codewords in $\mathcal{C}$.

$$\min\{\,\Delta(x,y) : x,y \in \mathcal{C} \text{ and } x \neq y\,\}$$

| $m$ | $c = encode(m)$ |
|---|---|
| 00 | 10 01 11 |
| 01 | 10 10 10 |
| 10 | 01 01 10 |
| 11 | 01 01 11 |



## Poll: decoding a message

Consider this code $\mathcal{C}$?

| $m$ | $c = encode(m)$ |
|---|---|
| 00 | 10 01 11 |
| 01 | 10 10 10 |
| 10 | 01 01 10 |
| 11 | 01 01 11 |

What is its minimum distance?

A) 0

B) 1

C) 2

D) 3

The minimum distance of code $\mathcal{C}$ is the smallest Hamming distance between two distinct codewords in $\mathcal{C}$.

$$\min\{\,\Delta(x,y) : x,y \in \mathcal{C} \text{ and } x \neq y\,\}$$



## Theorem: minimum distance and detecting/correcting errors

If the minimum distance of a code $\mathcal{C}$ is $2t+1$, then $\mathcal{C}$ can detect $2t$ errors and correct $t$ errors.

Proofs on board