WELCOME TO CODING THEORY!!

LECTURER: MADHU Sudan
madhu@mit.edu

TA: SWASTIK KOPPARTY
swastik@mit.edu

URL: http://people.csail.mit.edu/madhu/ST08

Mailing List: Exists. Mail madhu/swastik to get added
ADMINISTRIVIA

- 3-0-9 IT Level Credit, but no EC's, no TQE's. (Anything is possible by petition)

- To pass you must
  - turn in all psets (3-4)
  - do a project
  - scribe one lecture
  - participate actively in lectures

- CHECKLIST
  - fill up Signup Sheet
  - Signup for scribing early
    + read instructions on web site.
Error-Correcting Codes [Hamming/Shannon]

Hamming's Problem (roughly)

- Want to store bits on magnetic storage device

- Bits get corrupted, 0 → 1 or 1 → 0 but rarely. Say one in every block of 63 bits.

- How can you store information so that it is not lost?

*(Why 63? Will see....)*
Naive Solution

Repeat every bit 3 times

Encoding

message

Codeword

majority

errors

Decoding

received word
Good News:

- Can encode 21 bit message as 63 bit codeword.

- Encoding/Decoding simple (polytime computable).

Bad News:

- Rate \( \frac{\text{message length}}{\text{codeword length}} = \frac{21}{63} = \frac{1}{3} \)

Not so great!

Can we do better?

How much better?

Will encoding/decoding be easy?
Hamming Solution - 1

- Break message into 4 bit chunks
- Encode each chunk as follows:

$$\mathbf{m} \rightarrow \mathbf{m} \cdot \mathbf{G}$$

where $$\mathbf{m} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \in \{0,1\}^4$$

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \ 0 & 1 & 0 & 0 & 1 & 0 \ 0 & 0 & 1 & 0 & 0 & 1 \ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- Surprising Property:

  $$\forall \mathbf{m}_1 \neq \mathbf{m}_2, \ \mathbf{m}_1 \mathbf{G} \neq \mathbf{m}_2 \mathbf{G}$$ differ in 

  $$\geq 3$$ bits [Will prove later]
- **Rate**: 4 bits → 7 bits
  36 bits → 63 bits
  \[ \text{Rate} \approx \frac{4}{7} \quad \text{(will do better)} \]

- **Encoding** — simple

- **Decoding? More Magic**

- received word
  
  7 bits

\[
\begin{bmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
0 & 1 & 1 \\
1 & 0 & 0 \\
1 & 0 & 1 \\
1 & 1 & 0 \\
1 & 1 & 1 \\
\end{bmatrix}
\]
\( x_H = \text{index of flipped bit in binary!} \)

\[ \text{Voila!!} \]

- Is this the best one can do?
- No!

**Hamming Solution 2:**

```
\[ E \]
\[ G \]
\[ 6 \]
```

```
\[ H \]
\[ 6_3 \]
```

```
\[ 6 \]
```

```
\[ 6_3 \]
```

```
\[ E \]
```

```
\[ G \]
```

```
\[ 6 \]
```

```
\[ H \]
```

```
\[ 6_3 \]
```
Properties:

1. \( \forall m_1 \neq m_2 \in \{0, 1\}^{57} \)

   \( m_1, G \) & \( m_2, G \) differ in \( \geq 3 \) coordinates.

2. If a received word \( w \) has \( \leq 1 \) bit flip, then \( w H = \text{index of flipped coordinate} \).

Conclude:

- Can achieve rate = \( \frac{57}{63} \)

- Is this best possible?

\[ \text{[Hamming]}: \text{YES! No Encoding/Decoding scheme does better!} \]
Summary of Hamming’s Work

- Construction of “Error-correcting Code”
- Method for Encoding/Decoding
- Proof/Investigation of Optimality/Limits
  [Modelling critical to prove “optimality”]

Note: [Shannon] has all the same features with slightly different model/emphasis.
Will see next week.
Hamming's Notions:

1. Hamming Distance:
   - Let $\Sigma$ be some finite set
     - $\Sigma = \{0, 1\}$ in earlier example,
     - $\Sigma = \{0, 1, 2\}$ (bytes) in CDs
   - Let $\Sigma^n$ be set of $n$-letter words
     - over $\Sigma$ [Ambient Space]
   - For $x, y \in \Sigma^n$
     - $\Delta(x, y) = \#$ coordinates where $x, y$ differ
     - $\Delta(x, y) = |\{i \mid x_i \neq y_i\}|$
\[ \Delta = \text{Hamming Distance} \]

\[ S(x, y) = \frac{\Delta(x, y)}{n}. \quad [\text{relativized distance}] \]

- **Fact:** Hamming distance is a metric
  1. \( \Delta(x, y) = 0 \iff x = y \)
  2. \( \Delta(x, y) = \Delta(y, x) \)
  3. \( \Delta(x, y) + \Delta(y, z) \geq \Delta(x, z) \)

- **Conclude:** Can bring in geometric intuition to think about Hamming distance.
Hamming Notions (Contd.): Codes

\[ C \subseteq \mathcal{E}^n \quad \text{[set of codewords under some encoding]} \]

- \( C \) is \( t \)-error correcting if any pattern of up to \( t \) errors can be corrected [by some, possibly inefficient, decoding method]

- Formally:
  - \( \text{Ball}(x,t) = \{ y \in \mathcal{E}^n \mid \Delta(x,y) \leq t \} \)
  - \( C \) is \( t \)-error correcting if \( x = y \in C \) \( \text{Ball}(x,t) \cap \text{Ball}(y,t) = \emptyset \).
C is \( e \)-error detecting if whenever
\[ 1 \leq \# \text{ symbol errors} \leq e, \]
it can be detected that errors have occurred.

Formally:
\[ \forall x \in C \quad \text{Ball}(x, e) \cap C = \{x\}. \]
\[ \Delta(C) \quad (\text{Distance of Code}) \]

\[ = \min \left\{ \Delta(x, y) \right\} \]

\[ x \neq y \quad \text{for } x, y \in C \]

[Hamming] \[ x, y \in C \]

**Proposition:** \( C \) is \( t \) error-correcting \iff \( C \) is \( 2t \) error-detecting \iff Distance of \( C \) is \( \geq 2t + 1 \).
Proposition: for \( x \in \mathbb{Z}^n \) \( \exists \leq = \exists_{0,1}\exists \) \( \Rightarrow \exists (x,t) = \sum_{i=0}^{t} \frac{n}{i} \equiv \text{Vol}(n,t) \)

Proposition: \( \exists = \exists_{0,1}\exists \): if \( C \) is \( t \) error-correcting then

\[ |C| \leq \frac{2^n}{\text{Vol}(n,t)} \]

Can use above to conclude

\[ \text{Rate} = \frac{57}{63} \text{ is optimal in our example} \]
Rest of this course:

Follow Hamming's Plan

- Construct Codes (correcting more errors)
- Show Limitations
- Construct decoding algorithms

(Non-Hamming Part)

- See how codes are generally useful
  (in Math. & CS)
Some Claims & Disclaimers

- Motivation is more mathematical & less engineering; Not a substitute for communication/coding course, but complementary.

- It is a graduate class. Some math maturity is required.

- E.g. lots of texts – we will follow none!

- Hopefully, will have fun.