Special Topic: Error-Correcting Codes
Lale Özkahya

Resources:
https://courses.grainger.illinois.edu/cs598ak/sp2012/
Coding Theory

- Error-Correcting codes are used to compensate for noise and interference in communication.
- We consider the problem of transmitting bits (or maybe symbols from some small discrete alphabet).
- We only consider interference that consists of flipping bits.
- I.e. if I want to transmit the string 0101, the receiver might get 1101 but not 010.
- Amount of noise = number of bits flipped.
Coding Theory

- In this model, the transmitter wants to send \( m \) bits: message is an element of \( \{0,1\}^m \).
- If the transmitter wants the receiver to correctly receive the message in presence of noise, she should send \( n > m \) bits in such a way that the receiver can figure out what the original message was.
- Formally, we have an encoding function \( C: \{0,1\}^m \rightarrow \{0,1\}^n \) and for \( x \in \{0,1\}^m \) \( C(x) \) is its codeword.
- The receiver gets \( C(x)+e \), where \( e \) is a binary error vector. Tries to decode it, and hopefully get back \( x \).
Coding Theory

- Ratio $m/n$ is called the RATE of the code. (How many bits transmitted for each message bit) We want codes of high rate.
- Naïve first attempt: send every bit 3 times
- Rate is $m/3m = 1/3$.
- If only one bit was flipped then the receiver would be able to figure out which one it was.
- Very inefficient!
Hamming Codes

- What would you do if you had to work on the weekends?

- First idea in coding theory was parity bit: allows to detect one error, not correct it!
Wants to send $b_1, \ldots, b_m$

Constructs $b_{m+1} = \sum_{i=1}^{m} b_i$

Sends $b_1, \ldots, b_m, b_{m+1}$

Receiver can detect one error but cannot find where it is. Cannot detect more.

Receives $b_1 + 1, \ldots, b_m, b_{m+1}$
Hamming Codes

- Hamming codes combine parity bits in an interesting way to allow the receiver to correct one error.
- Can also detect but not correct two errors.
Hamming Codes

Wants to send \( b_3 b_5 b_6 b_7 \)

Constructs \( b_4 = b_5 + b_6 + b_7 \)
\( b_2 = b_3 + b_6 + b_7 \)
\( b_1 = b_3 + b_5 + b_7 \)

Sends \( b_1 b_2 b_3 b_4 b_5 b_6 b_7 \)

Receives \( b_1 b_2 b_3 b_4 b_5 (b_6 + 1) b_7 \)
The Asymptotic Case

- As number of bits to be transmitted became larger, an asymptotic approach was in order.
- We view error-correcting code as a mapping $C: \{0,1\}^m \rightarrow \{0,1\}^n$ for $n>m$.
- For $x \in \{0,1\}^m$ $C(x)$ is its codeword.
- Often identify $C$ with the set of codewords.
- Reminder: rate is $m/n$. 

The Asymptotic Case

• Hamming distance \( dist(c^1, c^2) \) between two codewords \( c^1, c^2 \) is number of bits in which they differ.

• Minimum distance of code is
  \[
  d = \min_{c^1 \neq c^2 \in C} dist(c^1, c^2)
  \]

• Large \( d \) => able to correct many errors (any number less than \( d/2 \)), proof
• Large \( d \) is good!!
The Asymptotic Case

- Minimum relative distance is $\delta = d/n$

- Possible to keep both the rate $m/n$ and the min relative distance bounded below by constants, as $n$ grows.

- Sequence of codes $C_1, C_2, \ldots$ (increasing message lengths) is asymptotically good if there are absolute constants $r$ and $\delta$:
  \[ r(C_i) \geq r \quad \text{and} \quad \delta(C_i) \geq \delta \]
Random Codes

- We will see that Random Linear Code is asymptotically good w.h.p.
- Two ways to define random linear codes
  - Choose rectangular \{0, 1\} matrix \( M \) at random and set \( C = \{ c : Mc = 0 \} \)
  - Instead, we choose \( m \)-by-\( n \) matrix \( M \) with independent uniformly choses \{0, 1\} entries and then set \( C(b) = Mb \)
- Code maps \( m \) bits to \( n \) bits, it is linear and has rate \( m/n \).
Random Codes

- Call our random code $C_M$
- Minimum distance of linear code is simplified:

$$\text{dist}(c^1, c^2) = \text{dist}(0, c^1 - c^2) = \text{dist}(0, c^1 + c^2)$$

- Linearity ensures that if $c^1, c^2$ codewords, so is $c^1 + c^2$.
- So, minimum distance is

$$\min_{0 \neq b \in \{0,1\}^m} \text{dist}(0, Mb) = \min_{0 \neq b \in \{0,1\}^m} |Mb|$$

$|s|$ is number of ones in $s$, called weight of $s$
Random Codes

**Theorem.** Let $M$ be a random $m$-by-$n$ matrix. For any $d$, the probability that $C_M$ has minimum distance at least $d$ is at least

$$1 - \frac{2^m}{2^n} \sum_{i=0}^{d} \binom{n}{d}$$

Prove that for every non zero rate $r=m/n$, asymptotically good codes exist (Gilbert-Varshamov bound).
Reed-Solomon Codes

- Key codes for coding theory
- Not binary codes. Symbols are elements of a finite field. We consider prime fields for now, $F_p$.
- These are numbers modulo prime $p$, they can be added, multiplied and divided.
Reed-Solomon Codes

- Message \((f_1, \ldots, f_m)\) in Reed-Solomon code over \(F_p\) is identified with polynomial of degree \(m-1\) :
  \[ Q(x) = \sum_{i=1}^{m-1} f_{i+1}x^i \]

- Codeword constructed from evaluating \(Q\) over every element of the field. That is, codeword is \(Q(0), Q(1), \ldots, Q(p-1)\)
Reed-Solomon Codes

- **Theorem.** The minimum distance of Reed-Solomon code is at least $p^m$

- However, not asymptotically good if we use $\log p$ bits to represent each field element. Code has length $p \log p$ but can correct only $< p$ errors (there is way around it)

- Next time: error correcting codes from expanders

- Next next time: construct expanders from error correcting codes.