①

Lecture 23: Information & Communication Theory Class.

There exists some code, if we pick codewords at random there are always situations where the code doesnt have enough codewords (sampling with replacement), the cardinality of the code is small. We could also end up with a BAD Code.

↳ 2^{nR} distinct vectors but they are close to each other (Some of them) & leads to large P(Error).

There is a non-zero probability that the code construction will not work. The theorem doesnt give us a guarantee. Next trial may work. Until we get a GOOD code

→ This is problem 1

Encoding & Decoding complexities are exponential in 'n'.

→ This is problem 2

Encoding: mapping : 2^{nR} is the size of the code. Every time we are communicating one codeword, we are sending nR bits (n times used)

for 2^{nR} items, the max Entropy $H(x) = \log 2^{nR} = nR$.

Encoding:

Map every nR length binary vector to a specific codeword. We will maintain a list of all possible nR length vectors & the n length codewords.

1 - 1 mapping

Implementing the Encoder via a table

msg vectors	→ Codeword
⋮	⋮
⋮	⋮
⋮	⋮

} 2^{nR} entries on both sides.

Searching in the table is 2^{nR} for the worst case. Table lookup process. Exponential in complexity of time & space. Assuming $O(1)$ time for each row.

Decoder:

Once it recieves y, it will find the nearest codeword from the recieved vector. Assuming the P(Error) to be really small, it will find a vector inside the np radius ball around the recieved vector 'y'.

→ Exponential time: $O(2^{nR})$ } → Decoding is exponentially complex (in n) for a random code as there are 2^{nR} codewords to search from.

The shannon Theorem for channel capacity is not enough for implementation.

Coding Theory Introduction:

→ Deals with actual construction of codes

→ The random code construction is not useful for implementation as

a) End up with a bad code.

b) Encoding & Decoding complexity is large.

We want codes which are good in rate P(error) & also has reasonable encoding & decoding complexity

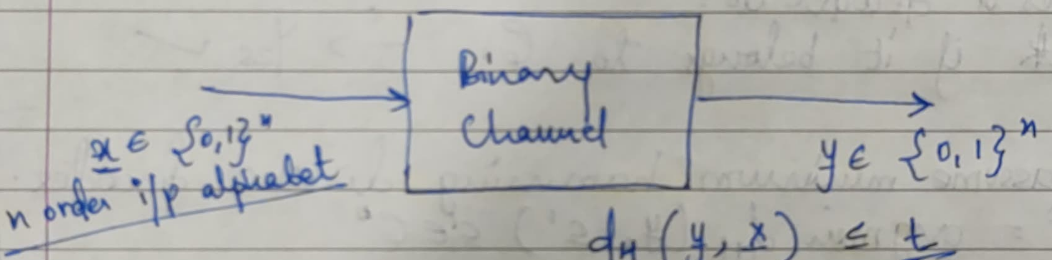
→ Linear codes have the above properties. Linear codes give good performance wrt Rate & P(Error) and low encoding & decoding complexity wrt random codes.

We will look at some simple examples of linear codes for Binary channel with worst case error model / Bounded error model.

→ Construction of codes which are useful for implementation is dealt with in <u>Coding Theory</u>

<u>Worst case error / Bounded error model for Binary Channel</u>

→ Let t, n be +ve integers such that $t \le n$.



$$d_H(y, x) \le t$$

For this channel we want to design a code $\mathcal{C}$ such that all t errors are corrected

t: Allowable no of errors the channel can cause in the rx vector.

→ Hamming distance is denoting the error.
→ No probability here.
→ We design the code to tackle any t errors.

→ This is different / stronger version to the probabilistic channel.

4

Example, $t = 1$:

Now suppose $\mathcal{C} = \{0, 1\}^n$. Let us construct a situation in which the decoder will surely make an error in decoding.

$\underline{c} = \underbrace{11 \ldots \ldots 1}_{n \text{ length}} \in \mathcal{C}$ has been transmitted,

suppose $y = 0\underbrace{111 \ldots 1}_{n-1}$. Is $y = 11 \ldots 1$ possible? Yes.

Can it make an error in decoding? Yes.
Depends on the decoder

What does a decoder do?
1) Check if it belongs to $\mathcal{C}$. $\longrightarrow$ Yes ✓

We will assume minimum hamming distance decoder.
$$\hat{c} = \arg\min d_H(y, \underline{c}') \quad \underline{c}' \in \mathcal{C}$$

for $y = (0, 1, 1 \ldots 1)$, $d_H = 0$ $\therefore$ It will assume it to true

$\min\limits_{\underline{c}' \in \mathcal{C}} d_H(y, c') = 0$ & this happens iff for $\underline{c}' = y$

$\therefore \underline{c}\hat{} = y$

This is a decoding error as $\underline{c}\hat{} \neq \underline{c}$

estimate from Decoder

Transmitted codeword.

→ It is clear that if $t \geq 1$, we cannot take the entire $\{0,1\}^n$ as the code.

→ So correcting any $t \geq 1$ errors requires us to pick proper subsets of $\{0,1\}^n$

→ Rate is inversely related to the size of $\mathcal{C}$.

→ We want to pick large subsets of $\{0,1\}^n$ as the code as we want to maximise

$$R = \frac{\log_2 |\mathcal{C}|}{n} \text{ bits/channel use.}$$

→ Picking large $\mathcal{C}$ ⇒ Codewords are 'closer' in hamming distance which means that they are more likely to cause decoding errors.

→ Tradeoff is needed - we will later obtain a bound called Hamming bound which describes this.

Lemma:

Let $\mathcal{C} \subseteq \{0,1\}^n$ be the chosen subset. Define

$$d_{min}(\mathcal{C}) \begin{bmatrix} \text{min distance} \\ \text{of the code } \mathcal{C} \end{bmatrix} = \min_{\substack{\underline{c}, \underline{c}' \in \mathcal{C} \\ \underline{c} \neq \underline{c}'}} d_H(\underline{c}, \underline{c}').$$

$\mathcal{C}$ can correct upto t-errors iff $d_{min}(\mathcal{C}) \geq 2t+1$

$d_{min}(\mathcal{C}) > 2t$.

Proof: (Exercise)

Think hamming balls of radius 't'.

If $d_{min} \geq 2t+1$, there is no decoding error upto 't' errors.

6

Using the hamming distance decoder, $\mathcal{C}$ is a code correcting upto 't' errors, prove $d_{min}(\mathcal{C}) \geq 2t+1$.

Only if statement.

If part:

Assume $d_{min}(\mathcal{C}) \geq 2t+1$

To prove : $\mathcal{C}$ can correct t-errors

Only if part:

Assume $\mathcal{C}$ can correct t-errors

To prove : $d_{min}(\mathcal{C}) \geq 2t+1$

Using Hamming distance decoder with the help of Pictorizing Hamming balls in $\{0,1\}^n$ space.