Lecture 12 — February 21

Lecturer: David Tse *Scribe: Rishi S, Cem K, John C, Liu F, Leonard B, Vivek B*

12.1 Outline

- Repetition Codes
- Polar Codes

12.2 The Road to Capacity

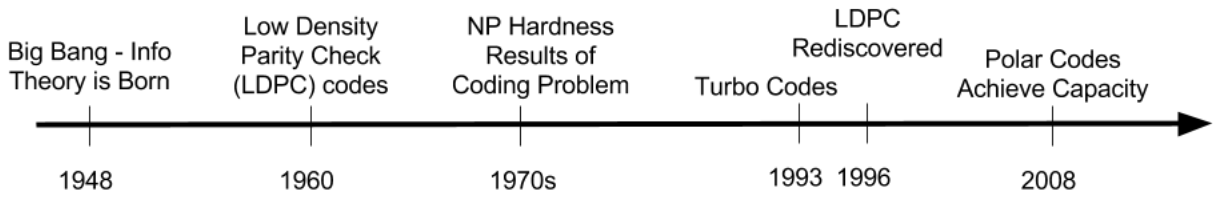

Figure 12.1: Timeline of developments towards efficient capacity achieving codes.

12.2.1 Timeline of Codes

Using the random coding argument, Shannon proved the existence of codes which achieve capacity. However, these capacity achieving codes were not computationally tractable and engineers spent the ensuing six decades trying to develop computationally efficient capacity achieving codes. Significant progress was made along the way, with a number of codes that *empirically* could achieve rates very close to capacity (Turbo Codes, LDPC Codes). Finally, with the development of polar codes in 2008 by Erdal Arıkan, the search came to an end as these codes achieve capacity while having encoding and decoding complexities of $O(n \log n)$, where n is the block length.

In this lecture and the next, we will revisit repetition codes through the lens of linear codes, and then modify them to obtain polar codes.

12.2.2 Recap – Random Coding and Random Linear Codes

The random coding argument (and its associated random code) established that a codebook chosen uniformly at random over the typical sequences achieves capacity. However, the time and space complexities for encoding and decoding is exponential in the size of its message length.

The first of these two problems i.e, the complexity of encoding was solved fairly quickly by the restriction to random *linear* codes. These codes are defined by a generator matrix G and the codeword for any message is defined by a matrix multiplication with G. For linear codes, we only need to store the generator matrix for encoding, which has a quadratic space complexity, a huge improvement over the exponentially large codebook required for general random codes.

However, the issue of decoding still remains a problem, as the decoding scheme for random linear codes remains a combinatorially hard problem of the form:

$$\hat{u}^k = \min_{u^k \in \{0,1\}^k} ||y^n - Gu^k||_H,$$

where y^n is the received message, and we are looking for the codeword $x^n = G\hat{u}^k$ that has the minimum Hamming distance from the received message. Thus, random linear codes do not serve as computationally efficient capacity achieving codes.

The search for efficient capacity achieving codes would animate research in information theory and coding theory for the next 60 years, culminating in 2008 with the discovery of polar codes; in this quest, many very interesting mathematical and engineering tools were developed, but we will skip all of this interesting material and jump to the end of the journey.

12.3 Polar Codes

12.3.1 Another Look at Repetition Codes

We develop polar codes through the lens of 'modified' repetition codes. To do so, we first take a new look at repetition codes. From now on, let $P : \{0, 1\} \to Y$ be a symmetric channel with capacity $C(P)$.

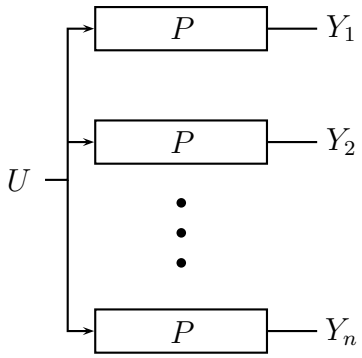

Figure 12.2: A repetition code makes n uses of the channel for input $U \sim Unif\{0, 1\}$.

The above view of repetition codes in Figure 12.2 casts new light on the known properties of repetition codes:

1. **Reliability:** The error probability of repetition codes, $p_e \overset{n\to\infty}{\longrightarrow} 0$. Equivalently, we may interpret this in terms of the mutual information between U and the output of the channel:

$$\begin{aligned} I(U; Y_1, Y_2, \ldots, Y_n) &= H(U) - H(U|Y_1, Y_2, \cdots, Y_n) \\ &= 1 - H(U|Y_1, Y_2, \cdots, Y_n) \\ &\overset{n\to\infty}{\longrightarrow} 1. \end{aligned}$$

2. **Rate:** The rate of communication is approximately 1 bit for every n uses of the channel. This is small compared to the maximum possible communication rate $nC(P)$. In the next section, we will modify this coding scheme to make up the difference and get more bits through the channel.

12.3.2 Repetition code for $n = 2$

Let us consider the repetition code for $n = 2$ on the BEC(p) channel with capacity $C = 1 - p$ (refer to Figure 12.3)

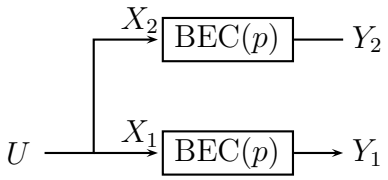

Figure 12.3: Repetition code for $n = 2$ for BEC(p) channel for input $U \sim \text{Unif}\{0, 1\}$. **Channel inputs:** $X_1 = X_2 = U$.

We lower bound the mutual information term

$$\begin{aligned} I(U; Y_1, Y_2) &= I(U; Y_1) + I(U; Y_2|Y_1) \\ &\geq I(U; Y_1) \\ &= C. \end{aligned} \tag{12.1}$$

The last equality in equation (12.1) follows from the fact that the uniform distribution U achieves capacity on symmetric channels. On the other hand, we upper bound the same mutual information term,

$$\begin{aligned} I(U; Y_1, Y_2) &= I(X_1, X_2; Y_1, Y_2) \\ &\overset{(a)}{\leq} 2C, \end{aligned} \tag{12.2}$$

where inequality (a) follows from the definition of capacity i.e, $\max_{p(x_1,x_2)} I(X_1, X_2; Y_1, Y_2) = 2C$.

Thus, from equation (12.2), we conclude that the repetition code in Figure 12.3 does **not** achieve capacity and has a gap of $2C - I(U; Y_1, Y_2)$. In the next subsection, we shall try closing the gap by modifying the above coding scheme in Figure 12.3.

12.3.3 Squeezing in More Bits

Consider the following coding scheme with two inputs U and V:

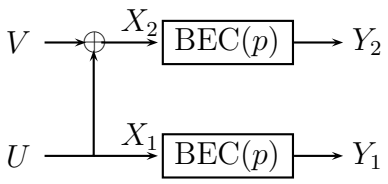

Figure 12.4: Modified coding scheme for $U, V \overset{i.i.d.}{\sim} \text{Unif}\{0,1\}$. **Channel inputs:** $X_1 = U,\ X_2 = U \oplus V$.

Compared to the coding scheme in Figure 12.3, the coding scheme in Figure 12.4 changes the input X_2 from $X_2 = U$ to $X_2 = V \oplus U$, where $V \sim Unif\{0,1\}$. It is easy to see that $\{U, V\}$ and $\{X_1, X_2\}$ have a bijection, and further coupling this with the fact $X_1, X_2 \overset{i.i.d}{\sim} Unif\{0,1\}$, we have

$$\begin{aligned} I(U,V;Y_1,Y_2) &= I(X_1,X_2;Y_1,Y_2) \\ &= 2C = 2(1-p). \end{aligned} \tag{12.3}$$

From equation (12.3), we see that the coding scheme in Figure 12.4 achieves capacity.
Now we change course to obtain a different interpretation of the term $I(U,V;Y_1,Y_2)$, and we begin by splitting the mutual information term using the chain rule

$$I(U,V;Y_1,Y_2) = I(U;Y_1,Y_2|V) + I(V;Y_1,Y_2). \tag{12.4}$$

The first term $I(U;Y_1,Y_2|V)$ is nothing but the rate achieved by the repetition code for $n = 2$, and on evaluating it further we obtain

$$\begin{aligned} I(U;Y_1,Y_2|V) &= H(U|V) - H(U|Y_1,Y_2,V) \\ &= H(U) - \Pr\{Y_1 = Y_2 = e\} \\ &= 1 - p^2 \\ &\geq C. \end{aligned} \tag{12.5}$$

This is **equivalent to the rate achieved by passing input U through the BEC(p^2) channel**.
Now lets evaluate the second term of equation (12.4),

$$\begin{aligned} I(V;Y_1,Y_2) &= H(V) - H(V|Y_1,Y_2) \\ &= H(V) - \Pr\{Y_1 = e \cup Y_2 = e\} \\ &= 1 - 2p + p^2 = (1-p)^2 \\ &\leq C. \end{aligned} \tag{12.6}$$

From the above equation, we see that the term $I(V;Y_1,Y_2)$ **is equivalent to the rate achieved by passing input V through a BEC($\tilde{p}$) channel**, where $\tilde{p} = 1 - (1-p)^2$.

The channel corresponding to the input V squeezes the extra bits equal to the gap between the capacity $2C$ and the rate of the repetition channel($n = 2$).
In light of the above discussions, we conclude that:

1. From U's point of view, it is passed through a BEC(p^2) channel, which has **more** capacity than BEC(p).

2. However, from V's point of view, it is passed through a BEC($1-(1-p)^2$) channel, which has **less** capacity than BEC(p).

This is diagrammatically represented in Figure 12.5.

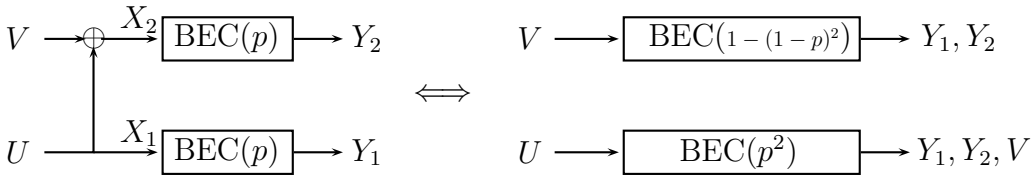

Figure 12.5: Diagrammatic interpretation of equivalence between coding scheme in Figure 12.4, and two separate binary channels for U and V.

Sanity Check for equivalence: From equations (12.4), (12.5) and (12.6), we have

$$\begin{aligned} I(U,V;Y_1,Y_2) &= I(V;Y_1,Y_2) + I(U;Y_1,Y_2|V) \\ &= 1-p^2+(1-p)^2 \\ &= 2(1-p), \end{aligned}$$

which gives us back equation (12.3).

Extending the conclusions in section 12.3.3, two uses of a general symmetric channel P are equivalent to a P^+ and P^- channel, as shown in Figure 12.6.

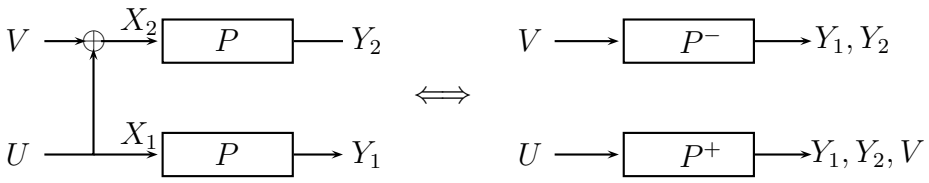

Figure 12.6

12.3.4 Conclusion

Thus, we have in some sense divided two symmetric channels P into two separate channels. The first channel P^- is less 'reliable' than P, whereas the second channel P^+ is more 'reliable'. In the next lecture, we'll cascade this argument to obtain polar codes.