Lecture 11 — February 14

Lecturer: David Tse *Scribe: Junjie J, Jiada L, Yizheng L, Vivek B*

11.1 Outline

- Achieving capacity using random codes
- Linear codes

11.2 Recap

In the last couple of lectures we showed that one cannot communicate reliably at a rate above the capacity. Therefore, we are left with the question of whether we can design codes whose rate is *arbitrarily close* to the capacity.

11.2.1 Sphere packing

We start by revisiting the geometric picture of coding shown in Figure 11.1. Consider a set of M codewords each of block length n: $x^n(1), \ldots, x^n(M)$.

Each codeword, when sent over the channel, will generate a sphere of uncertainty (a.k.a noise sphere). That is, for a particular codeword $x^n(m)$, the resulting output $Y^n(m)$ (with high probability) lies in the noise sphere of $x^n(m)$. We want to design codewords to ensure that the amount of overlap between the noise spheres is small: large overlap corresponds to large decoding error. Before we figure out how to do that, let us first give a rough upper bound on how many codewords we can choose to avoid large overlap.

Observe that for every codeword $x^n(m)$, the number of jointly typical sequences $(x^n(m), Y^n(m))$ (size of its noise sphere) is approximately $2^{nH(Y|X)}$. Coupling this with the fact that the total number of typical output sequences in the $\{0,1\}^n$-sphere is $2^{nH(Y)}$, we have

$$\begin{aligned} M &\leq \frac{\#\text{Typical output sequences in the } \{0,1\}^n\text{-sphere}}{\#\text{Typical sequences in a noise sphere}} \\ &= \frac{2^{nH(Y)}}{2^{H(Y|X)}} \\ &= 2^{nI(X,Y)}. \end{aligned} \tag{11.1}$$

This yields heuristically the same result as we proved rigorously in the last lecture, namely that we cannot communicate reliably above capacity. But the fact that this "sphere-packing" picture gives the correct upper bound suggests that it is a useful visualization of thinking about how to achieve this upper bound, namely, we need to pack the $\{0,1\}^n$-sphere using $2^{nI(X,Y)}$ noise spheres with negligible overlap.

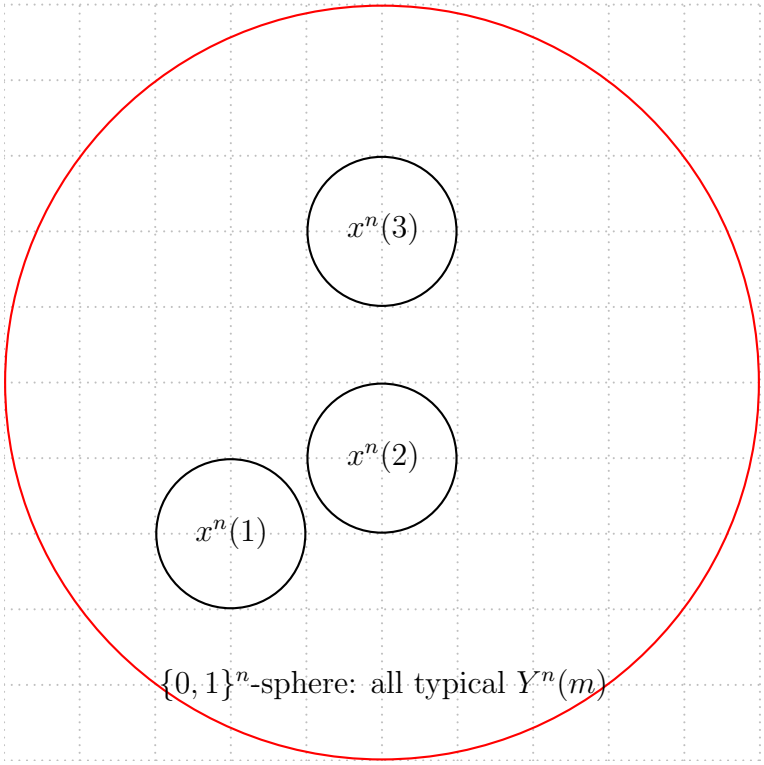

Figure 11.1

11.3 Random Codes

Instead of obtaining specific capacity-achieving codes, Shannon devised a random coding technique, which only shows the **existence** of capacity-achieving codes.

11.3.1 Coding scheme C

Unless otherwise mentioned, *input typical sequences* are defined w.r.t the capacity-achieving distribution, i.e. $p^\star(x) = \text{argmax}_{p(x)} I(X;Y)$ and *output typical sequences* are defined w.r.t the distribution $p(y) = \sum p^*(x)p(y|x)$, where $p(y|x)$ is fixed for a given channel.

Construct a random code $\mathbf{C}$ by independently picking M codewords uniformly over the set of input typical sequences. Since the coding scheme is random, we denote these codewords by random variables $X^n(1), X^n(2) \cdots X^n(M)$. Even though the coding scheme $\mathbf{C}$ is randomly chosen, both the encoder and the decoder 'know' the coding scheme (after it is chosen).

11.3.2 Decoder

For the sake of exposition, will shall **fix** our message m for the entire discussion below. Suppose we send the random codeword $X^n(m)$ through the channel and receive an output sequence $Y^n(m)$. The output $Y^n(m)$ will fall into one of the three cases:

- Case 1: there is a unique $\tilde{m} = m$ for which $(X^n(\tilde{m}), Y^n(m))$ is jointly typical. In this case, the decoder returns m.

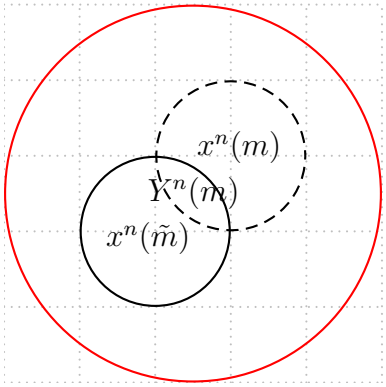

Figure 11.2: Coding schema

- Case 2: there is another $\tilde{m} \neq m$ for which $(X^n(\tilde{m}), Y^n(m))$ is jointly typical. Here, the decoder declares an error.
- Case 3: there exists no $\tilde{m}$ for which $(X^n(\tilde{m}), Y^n(m))$ is jointly typical. Here too, the decoder declares an error.

11.3.3 Probability of error $\mathbb{E}\big[P_e(\mathbf{C})\big]$

Case 3 occurs only when the output $Y^n(m)$ does not belong to the set of 'output typical sequence' and from *AEP,* the probability of this event vanishes to zero as $n \to \infty$. Therefore, we have to bound the probability of case 2 in order to bound $\mathbb{E}\big[P_e(\mathbf{C})\big]$.

It is easy to see that case 2 occurs if and only if $Y^n(m)$ lies in an overlapping region of two (or more) noise spheres (Figure 11.2); computing the probability of such an event for a particular coding scheme $\mathbf{C} = c$ is hard. However, we can compute the average probability of this event over all the coding schemes.

Before proceeding any further, lets take a step back and analyze the random variables $Y^n(m)$ and $X^n(\tilde{m})$ for $\tilde{m} \neq m$.

1. From the construction of the codewords, $X^n(m)$ is uniform over the set of input typical sequences. Since the output $Y^n(m)$ depends only on the input $X^n(m)$ (and the channel), it is uniform over the set of output typical sequences.
2. Since $X^n(m)$ and $X^n(\tilde{m})$ are picked independently, and $Y^n(m)$ depends only the input $X^n(m)$ (and the channel), $Y^n(m)$ and $X^n(\tilde{m})$ are independent.
3. From the construction of the codewords, $X^n(\tilde{m})$ is also uniform over the set of input typical sequences.

Using the above three points, for $\tilde{m} \neq m$, the probability that $\big(X^n(\tilde{m}), Y^n(m)\big)$ is a jointly typical sequence is

$$\frac{\text{number of jointly typical sequences}}{\text{product of input and output typical sequences}} \approx \frac{2^{nH(X,Y)}}{2^{nH(X)}2^{nH(Y)}} = 2^{-nI(X,Y)}. \tag{11.2}$$

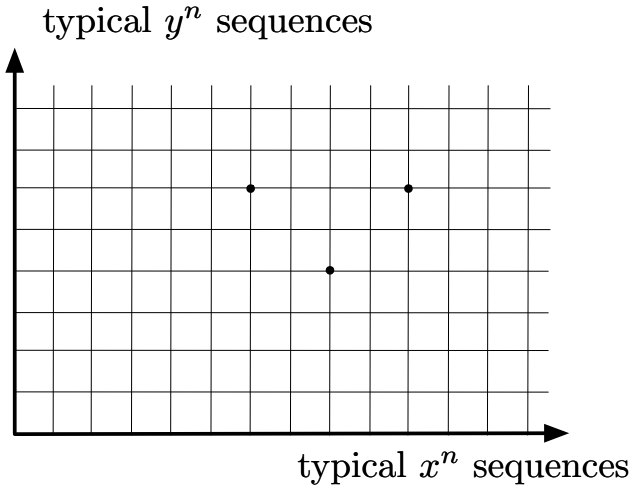

Figure 11.3: Only a subset of the intersections (black dots) are jointly typical

Refer to Figure 11.3 for a diagrammatic explanation of the above derivation.

Equation (11.2) gives us the probability that $X^n(\tilde{m})$ is jointly typical with $Y^n(m)$ for any message $\tilde{m} \neq m$. Therefore, using union bound we have

$$\begin{aligned} Pr(\exists \tilde{m} \neq m, \text{ st. } X^n(\tilde{m}) \text{ is jointly typical with } Y^n(m)) &\leq (M-1)2^{-nI(X;Y)} \\ &= 2^{nR}2^{-nI(X;Y)} \\ &\overset{n\to\infty}{\longrightarrow} 0 \end{aligned} \tag{11.3}$$

for $R < I(X:Y) =$ Capacity. Since the above analysis does not depend on the choice of the input message m, these arguments directly imply that the **average** probability of error across all random codes converges to 0, i.e.,

$$\mathbb{E}\big[P_e(\mathbf{C})\big] \overset{n\to\infty}{\longrightarrow} 0.$$

Since $P_e(\mathbf{C})$ for all codes cannot be strictly above the average, there exists a code C^* such that

$$P_e(C^*) \leq E_{\mathcal{C}}[P_e(\mathbf{C})] \to 0.$$

Therefore, we have shown the existence of a capacity-achieving code without specifying the code itself!

11.3.4 Alternative argument

The probability in (11.3) is computed by averaging over both the randomness in $Y^n(m)$ and all the other codewords $X^n(\tilde{m})$'s, $\tilde{m} \neq m$. But in fact, one can make a slightly stronger statement: *conditional* on a particular choice of $X^n(\tilde{m}) = x^n(\tilde{m})$ for all $i \neq m$, the probability of $Y^n(m)$ being jointly typical with some $x^n(\tilde{m})$ is also upper bounded by $(M-1)2^{-nI(X,Y)}$. This is because the size of the noise sphere, i.e. the number of output sequences jointly typical with $x^n(\tilde{m})$, is $2^{nH(Y|X)}$. Therefore, the total number of output sequences in the union of all the noise spheres corresponding to all the other codewords (i.e. $m \neq \tilde{m}$) is

at most $(M-1)2^{nH(Y|X)}$. Since $Y^n(m)$ is uniform in the set of typical output sequences, the probability of $Y^n(m)$ falling in some noise sphere is precisely $(M-1)2^{nH(Y|X)}/2^{nH(Y)}$. Hence, if $(M-1)2^{nH(Y|X)} \ll 2^{nH(Y)}$, this probability is small. This condition almost exactly matched our heuristic upper bound argument by comparing the total volume of the noise spheres with the total volume of the output sequence sphere. The key idea that enables the matching is to randomize the choice of $X^n(m)$, so that $Y^n(m)$ is uniformly distributed among thte typical output sequences.

11.3.5 Computational efficiency

Even though the random coding scheme achieves capacity, it is computationally inefficient for the following reasons:

- Storage: For achieving capacity we need to use large block lengths i.e, $n \gg 1$. The above coding scheme needs to store all the codewords since there is no structure in the codebook. Thus, the space and time complexity of this (brute) encoding scheme is exponential in n. For example, if $R = 1/2$ and $n = 1000$, we have $M = 2^{nR} = 2^{500} \approx 10^{150}$ codewords!
- Decoding: The time complexity of the decoder is also exponential in n, as one needs to test for joint typicality with the output sequence for all the codewords.

11.4 Linear codes

The space and time complexity for encoding random codes is large because the codewords are not 'structured' w.r.t the messages. We overcome this problem using *random* ***linear*** *codes*, where a code is defined by a generator matrix G; for a message $u^k \in \{0,1\}^k$, the corresponding codeword is

$$x^n = Gu^k.$$

Here, the generator matrix G is a random matrix in the space $\{0,1\}^{n\times k}$. Since the encoder needs to store only the generator matrix G, the space and time complexity for encoding is $nk \ll 2^n$.

Now that linear codes exhibit efficient encoding scheme, the next logical questions are:

1. Do random linear codes achieve capacity?
2. What is the time complexity of the corresponding decoder?

Special case: BEC(p)

In order to get a better understanding of random linear codes, let us first characterize it on the BEC(p) channel. For BEC(p), on the average np entries of x^n are erased and decoder only observes $n(1-p)$ entries of x^n. In the language of linear algebra, this is equivalent of

having $n(1-p)$ equations and k variables which immediately gives us the following bound:

$$\begin{aligned} k &\leq n(1-p) \\ \implies R = \frac{k}{n} &\leq 1-p = \text{ Capacity.} \end{aligned}$$

The above bound shouldn't come as a surprise, because we've already shown that no code can achieve a rate greater than the capacity. Conversely, it can be shown that for any $\epsilon > 0$, a random $(1+\epsilon)k \times k$ square matrix has full column rank with high probability (for large k), which further implies that random linear codes achieve capacity for a BEC(p). Inadvertently, the time complexity of the decoder for BEC(p) happens to be polynomial in n because inverting a general n by n matrix requires $O(n^3)$ operations. However, as shown later, decoding is not polynomial for a general channel.

General channel

For general binary symmetric channels, one can prove that linear random codes achieve capacity using a similar argument as the proof that general random codes achieve capacity. However, there is no known efficient decoding algorithm for a general channel; the optimal decoder (MLE) looks for the codeword which has minimum hamming distance from the received message y^n, i.e.,

$$\hat{u}^k = \min_{u \in \{0,1\}^k} ||y^n - Gu^k||_H. \tag{11.4}$$

For a general symmetric channel and a general matrix G, the above decoder has exponential time complexity.

11.4.1 What's next?

In this lecture, we have shown that a general linear code can encode in polynomial time but **cannot** decode in polynomial time. Hence, the natural question is - can we restrict our class of linear codes to codes which can encode and decode in polynomial time? The answer is yes!

In the upcoming lectures, we will design polar codes (a sub-class of linear codes), which achieve the capacity for a general channel along with encoding and decoding space/time complexities of order $O(n \log n)$. Stay tuned!