Lecture 8 — February 2

Lecturer: David Tse *Scribe: Mark N, Varun V., Russel L, Richard W, Vivek B*

8.1 Outline

- Communication over noisy channels
- Channel capacity

8.1.1 Readings

- Shannon Part II - Sec. XI - XIV

8.2 Communication over noisy channels

8.2.1 Introduction

In previous lectures, our communication model consisted of three elements: a source, an encoder to compress the data into a codeword, and a decoder, which 'inverts' the codeword to obtain the original data:

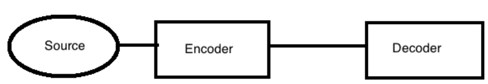


Figure 8.1: Source coding.

Our principal concerns in this framework were the compression rate and the efficiency of various coding schemes. We established lower bounds on the compression rate and designed codes that could achieve these bounds such as Huffman code and arithmetic code. These codes removes all redundancy in the source, getting down to the fundamental limit of the its entropy rate.

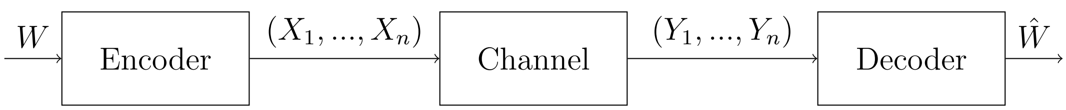
However, until now our model has ignored an important characteristic of practical communication systems: the noisy nature of the communication **channel**. To deal with this, we need to do channel encoding and decoding to introduce redundancy to combat the random noise in the channel. The entire system, including source and channel coding, is shown below.

Figure 8.2: End-to-end system.

In this lecture, we consider the challenge of communicating over a noisy channel, and in the next few lectures, we will characterize the fundamental limit of communication and how to efficient achieve it.

8.2.2 Model

Focusing only on the channel coding part of the overall system, we will deal with the situation below:



Communicating over a noisy-channel is a more challenging scenario than source coding because it has to deal with two sources of randomness, the source data and the channel noise. Indeed, modeling the channel probabilistically was one of the most important contributions of Shannon, replacing a previous deterministic model of the channel.

However, in our analysis, it will be possible to simplify the model of uncertainty in the input data. As we have seen in previous lectures, the bit sequences produced by an optimal compressor have a **uniform distribution**. (otherwise we could apply more compression to reduce the redundancy of the bits even further) Therefore, from the perspective of the channel encoder, we can assume that the input bits are uniformly distributed; the only remaining question is to transmit these bits with as reliably and as fast as possible.

Now a bit more details. The encoder receives message bits from the compressor. We receive a sequence of bits $(B_1, B_2, ..., B_k)$ that is uniformly distributed on $\{0,1\}^k$. We enumerate over all possible messages and assign each sequence a number W:

$$(B_1, B_2, ..., B_k) \to W$$

where $W \in \{1, ..., 2^k\}$. Note that W is a random variable.

The encoder takes in the message W and outputs a sequence of symbols $X_1, X_2, ..., X_n$, with $X_i \in \{0,1\}$. These symbols are sent through the channel, which outputs a sequence of symbols $Y_1, Y_2, ..., Y_n$. These symbols are read by the decoder, which outputs a decoded message $\hat{W}$.

There are two important metrics to consider when analyzing this system.

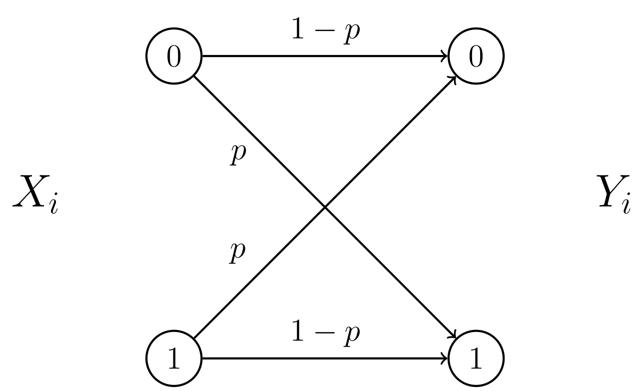
- **Probability of error** $p_e = \mathbb{P}\left(\hat{W} \neq W\right)$.
- **Rate** $R = \frac{k}{n}$.

Let's look at some sample channels:

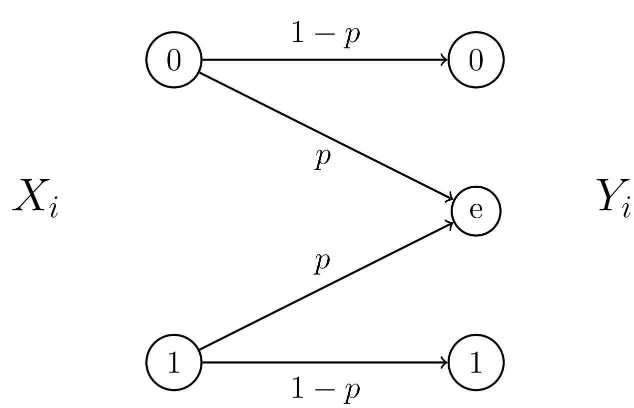
8.2.3 Binary Symmetric Channel (BSC)

Given $Z_i \sim \text{Ber}(p)$.

$$Y_i = X_i \oplus Z_i$$



8.2.4 Binary Erasure Channel (BEC)



Both of these channels operate under the **memoryless** assumption: that is,:

$$p(y_1, ..., y_n|x_1, ..., x_n) = p(y_1|x_1)p(y_2|x_2)...p(y_n|x_n)$$

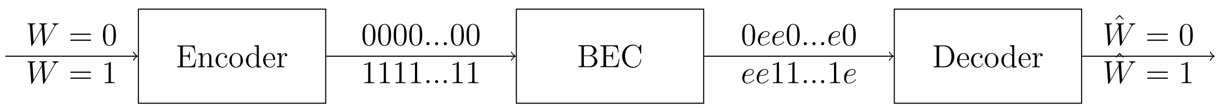
This says that conditional on X_i, Y_i is independent of all X_j, $j \neq i$.

8.3 Channel capacity

8.3.1 Motivation

The goal of communication will be to optimize the **tradeoff between R and p_e**. More specifically, we want to minimize the probability of error for a given data rate R. How does this optimal tradeoff look like. We start by looking at the tradeoff achieved by a specific family of codes.

Example 1 (Repetition Codes)**.** Consider a single-bit message $W \in \{0, 1\}$, being sent over the binary erasure channel with erasure probability p. The encoder will send a string of n symbols that are all 0's if W is 0, and all 1's if W is 1.



The decoder sees a mixture of 0's and e's or a mixture of 1's and e's. It looks for the first bit that's not erased, and outputs that bit. The rate and probability of error are:

- $p_e = \frac{1}{2}p^n$, in the case when all the bits get erased and a random guess makes a mistake.
- $R = \frac{1}{n}$ bits per channel symnol, because we need to send n channel symbols to encode a message of length $k = 1$.

The tradeoff curve looks like this:

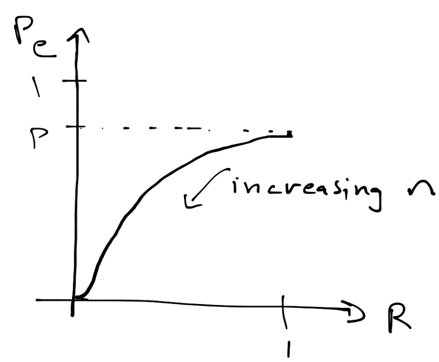


Figure 8.3: The tradeoff curve for a repetition code over a BEC.

Note that to achieve smaller and smaller error probability, the data rate has to go smaller and smaller to zero. Before Shannon, people thought that this is true for any communication schemes, not just repetition codes.

8.3.2 The amazing thing

Shannon however showed that the optimal tradeoff curve (for a BEC with erasure probability p) looks like this:

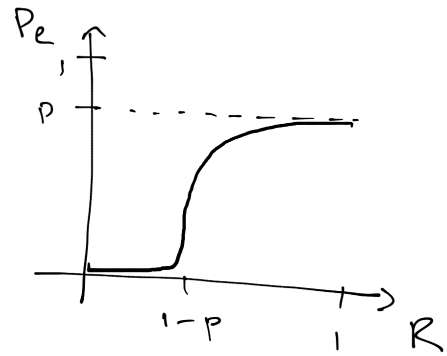


Figure 8.4: Shannon's optimal tradeoff curve for a BEC.

This is a very surprising result! Communication is possible with arbitrarily small probability of error at positive rates, all the way up to $C = 1-p$. C is naturally called the *capacity* of the channel. ! No feedback is required to achieve this result, and a channel with feedback can do no better than this! The above curve displays **critical** behavior - it is very small for rates less than $1-p$, but blows up for rates even slightly above $1-p$. It turns out this kind of optimal tradeoff holds for all memoryless channels. Only the capacity is different, depending on the channel statistics.

8.3.3 Channel Capacity: General Memoryless Channel

The channel capacity of a general memoryless channel with transition probability $p(y|x)$ is:

$$C = \max_{p(x)} I(X;Y)$$

Note that the capacity of a channel is only a function of $p(y|x)$; the input distribution is optimized to "match" the channel.

Example 2 (Capacity of BSC)**.** A BSC is modeled by the relation $Y = X \oplus Z$. Recall:

$$I(X;Y) = H(Y) - H(Y|X)$$

Therefore $H(Y|X) = H(Z)$ because the uncertainty in Y given X is equal to the uncertainty in Z. This is a constant, and so :

$$C = \max_{p(x)} H(Y) - H(Z)$$

is maximized with $H(Y)$ is 1 (which is true when $p(x)$ is uniform). Hence the channel capacity is $1 - H(Z)$, and if $Z \sim \text{Ber}(p)$, then

$$C = 1 - H(Z) = 1 - H(p) = 1 - p\log\frac{1}{p} - (1-p)\log\frac{1}{1-p}$$

For a Binary Erasure Channel, $C = 1-p$. Intuitively, given n symbols going through the channel, only roughly $(1-p)n$ coded symbols don't get erased, and hence you cannot decode more than $(1-p)n$ bits. It is however not clear how one can design a code to achieve this.

You might expect, then, that a Binary Symmetric Channel should have capacity $(1-p)$ as well! But Shannon made the following argument: what if you were transmitting over a BSC with flip probability $p = 0.5$? Under the previous argument, one might think that the capacity should then be 0.5 bits as well, but this is clearly false because a BSC that flips each symbol with probability 0.5 should transmit no information. The output symbol is completely independent of the input symbol. The key is that with a BSC, we know that $\sim 1/2$ the coded symbols made it through the channel unchanged, but we don't know **which** ones. This is in contrast to the erasure channel, where the decoder knows exactly which symbols were erased.

8.4 Next steps

Shannon showed that low p_e codes exist for rates up to the channel capacity. However, he did not show how to explicitly construct such codes - instead, he used a random coding argument. Even more importantly, he never showed how to efficiently encode and decode. Instead, It is only in the past 10 years that we have found explicit binary codes that achieve the channel capacity and can be encoded and decoded efficiently.