Capacity of the Binary Erasure Channel

Electrical Engineering 126 (UC Berkeley)

Spring 2018

1 Communication

This note is about the breakthrough work of Claude Shannon in the 1940s. We begin with Shannon's famous block diagram, Figure 1.

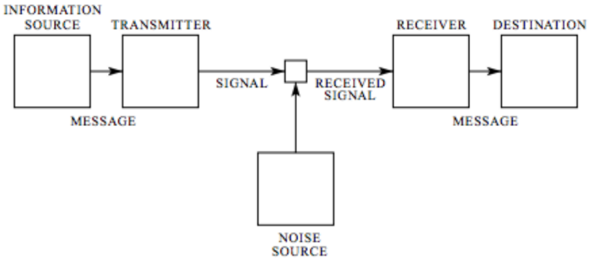


Figure 1: Shannon's block diagram of a communication system.

Suppose that you want to send a message over a noisy channel. The basic steps of a digital communication system are:

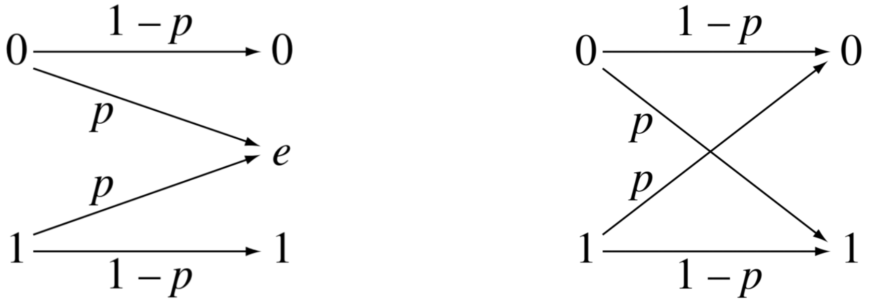
1. The source message is *compressed*.
2. *Redundancy* is added to deal with the noise in the channel.
3. The coded message is sent through the communication channel.

On the other end of the channel, the receiver reverses all of the above steps. Shannon showed that we can design the source coding in 1 and the channel coding in 2 *separately* and still transfer information over the communication channel at the optimal rate. For the rest of the note, we will focus on the channel encoding and decoding for the special case of the binary erasure channel.

2 Capacity of the Binary Erasure Channel

The two simplest models studied are the **binary symmetric channel (BSC)** and **binary erasure channel (BEC)**. The two channels are shown in Figure 2. We will focus on the BEC, which erases the input to the channel with probability $p \in (0,1)$.

We are interested in the maximum number of bits that the transmitter can send over the channel per transmission without error.



(a) BEC

(b) BSC

Figure 2: The BEC and the BSC.

Suppose that we have a message of length L and we encode to a message of length n, where L and n are positive integers. Intuitively, n must be larger than L in order to account for the erasures in the BEC. In our model of communication, we must send our encoded message over the BEC *without feedback*, which means that the receiver is not allowed to contact us in the middle of the transmission to let us know which bits were erased. Later we will consider what happens when we allow the receiver to provide feedback.

Assuming that the message goes through the channel without errors, the receiver receives a string of n bits which provides L bits of information. The ratio $R := L/n$ is therefore called the **rate** of the code: the number of bits of information about the source message per symbol that the receiver receives.

Let $\mathcal{X}$ be the **input alphabet** and $\mathcal{Y}$ be the **output alphabet**. Accompanying any code is an **encoding function** $f_n : \mathcal{X}^L \to \mathcal{X}^n$ and a **decoding function** $g_n : \mathcal{Y}^n \to \mathcal{X}^L$ which tells us how to decode the output of the channel. For the BEC, the input alphabet is binary, $\mathcal{X} := \{0, 1\}$, and the output alphabet is $\mathcal{Y} := \{0, 1, \mathbf{e}\}$, where the symbol $\mathbf{e}$ indicates that the symbol was erased by the channel.

Now, we have to account for the noise in the channel. Let $X^{(n)} := (X_1, \ldots, X_n)$ be the n bits which are fed into the channel and let $Y^{(n)} := (Y_1, \ldots, Y_n)$ be the n bits which are the output of the channel. The **maximum probability of error** of the code is

$$P_{\mathrm{e}}(n) := \max_{x \in \mathcal{X}^L} \mathbb{P}\{g_n(Y^{(n)}) \neq x \mid X^{(n)} = f_n(x)\}.$$

Take time to parse the above definition. We are considering the maximum probability that the decoding function, when applied to the output of the channel, differs from the original intended message, where the maximum is taken over all choices of the input message.

We say that the rate R is **achievable** for the channel if for each positive integer n there exist encoding and decoding functions (f_n, g_n) which encode messages of length $L(n) := \lceil nR \rceil$ to messages of length n, such that $P_{\mathrm{e}}(n) \to 0$ as $n \to \infty$ (asymptotically error-free). The largest achievable rate of the channel is called the **capacity** of the channel.

The main goal is to show the following:

Theorem 1. *The capacity of the BEC with error probability p is $1 - p$.*

Proof. First, we will show that we can do no better than rate $1 - p$. Indeed, even *with feedback* (the receiver notifies the transmitter about exactly which bits were erased by the channel), the best that the transmitter can do is to resend the bits which were erased. Since the channel erases a fraction p of the input bits, the reliable rate of communication is $1-p$ bits per channel use. [1]

Next, we will show that we can achieve a rate of $R := 1-p-\epsilon$ for any $\epsilon > 0$. Shannon's insight was to leverage the SLLN to achieve capacity. How do we generate a good codebook? Flip $n2^{L(n)}$ fair coins independently, and fill in an $n \times 2^{L(n)}$ codebook accordingly (thus each of the $2^{L(n)}$ possible messages are associated with a codeword of length n).

	c_1	$\cdots$	$c_{2^{L(n)}}$
bit 1			
$\vdots$			
bit n			

Figure 3: We fill in a $n \times 2^{L(n)}$ table. The columns represent the codewords $c_1, \dots, c_{2^{L(n)}}$ (one codeword per possible message) and the rows represent the individual bits of the codewords.

Since the channel is a BEC, a fraction p of the bits transmitted will be erased (by the SLLN). Suppose that the first codeword is sent. The receiver then gets the first codeword with a fraction p of its bits erased. Assume WLOG that the first $\lfloor n(1-p) \rfloor$ symbols came through (this is fine because the encoder does not know which bits were erased so it does not affect the coding). The receiver now looks at the codebook truncated to the first $\lfloor n(1-p) \rfloor$ rows and sees if there is a unique codeword matching the bits that were received. The decoding rule is that the decoder looks for a unique match in the codebook, and if a *unique* match does not exist, an error is declared. Thus, the probability of error is the probability that there exist ≥ 2 entries in the truncated codebook which match the received bits. If the truncated codewords are denoted $c_1, \dots, c_{2^{L(n)}}$, then consider codeword c_2; we have $\mathbb{P}(c_1 = c_2) = 2^{-\lfloor n(1-p) \rfloor}$. Hence,

$$\mathbb{P}(\text{error}) = \mathbb{P}\Big(\bigcup_{i=2}^{2^{L(n)}} \{c_1 = c_i\}\Big) \leq \sum_{i=2}^{2^{L(n)}} 2^{-\lfloor n(1-p) \rfloor} = 2^{L(n)} \cdot 2^{-\lfloor n(1-p) \rfloor} \sim 2^{-n(1-p-R)}.$$

We now examine the exponent and note that as $n \to \infty$, since $R < 1-p$, our error goes to 0 exponentially fast. □

We now have a sufficient scheme to achieve capacity, but what is the drawback? Decoding in this manner requires exhaustive search over a massive codebook, so it is practically useless. Thus, one needs implementable and fast codes to achieve capacity in practice.

3 The General Channel Coding Theorem

We will not cover the general result in this course, but we include it here for those who are interested.

[1] This is known as an **oracle argument** because it proves a fundamental limit on what can be achieved by showing that you can do no better, even if you had an oracle supplying you with extra knoweldge.

The general result is stated in terms of the **mutual information** of random variables, which is defined as $I(X;Y) := H(X) + H(Y) - H(X,Y)$. Let $\mathcal{X}$ denote the source alphabet of a channel, and let $\mathcal{Y}$ denote the corresponding output alphabet. Let X be the input to the channel (one transmission), and let Y be the output of the channel. Finally, let $\mathcal{P}$ denote the set of probability distributions on the input alphabet $\mathcal{X}$. The **channel capacity** is

$$C := \max_{p \in \mathcal{P},\, X \sim p} I(X;Y). \tag{1}$$

In words, we are looking for the largest possible mutual information between the input and output random variables, where the maximization is taken over all possible input distributions. This new definition does not conflict with our earlier definition of the capacity of the channel, because of the following famous result:

Theorem 2 (Channel Coding Theorem)**.** *Any rate below the channel capacity C (as defined in (1)) is achievable. Conversely, any sequence of codes with $P_{\mathrm{e}}(n) \to 0$ as $n \to \infty$ has a rate $R \le C$. Thus, the two definitions of the channel capacity which we have given agree.*

The general result is more difficult to prove than the special case of the BEC, but the BEC example already carries most of the intuition.