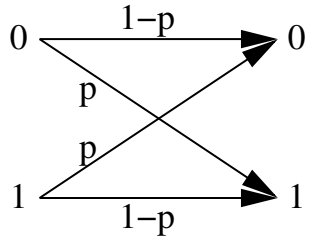
Shannon's Noisy Coding Theorem

Lecturer: Michel Goemans

1 Channel Coding

Suppose that we have some information that we want to transmit over a noisy channel. Nowadays, this happens all the time: when you're talking on a cell phone, and there is interference from radio waves from other devices; when you're playing a CD (on a good CD player, CD's are remarkably scratch-resistant); when you're downloading stuff from the Internet. You'd like to make sure that the information gets through intact, even though the line is noisy. How can we do this? Here we will discuss a theoretical result on how much information can be transmitted over a noisy channel. The proof given below that it can be done relies on an algorithm which is hopeless to run in practice because it would take way too long. In the next few classes, we'll talk about specific error correcting codes. First, we will discuss Hamming codes, which were the first error-correcting codes to be discovered. Next, we'll discuss BCH codes, which are one of the first families of codes discovered that worked reasonably well in practice.

To study the theory of communication mathematically, you first need a mathematical model of a communication channel. We will focus on one specific channel: the binary symmetric channel. This is a channel where you input a bit, 0 or 1, and with probability $1 - p$ is passes through the channel intact, and with probability p it gets flipped to the other parity. That is,



This is called a binary channel because the input and output are both bits, and symmetric because the probability of an error is the same for an input of 0 and an input of 1. We could define a more general type of channel, the memoryless channel, and give Shannon's theorem for this type of channel, and we won't do this.

How can you transmit information reliably, when you have a noisy channel? There's an obvious way to encode for transmission to ensure that a gets through, which has probably been known for centuries: you just send several copies of the message. In this scenario, what that means is sending many copies of each bit. Suppose that you replaced each 1 in the original message with five 1's, and each 0 with five 0's. Suppose the probability of error, p, is relatively small. To make an error in decoding the redundant message, at least three copies of each bit would have to be flipped, so the probability of error in a bit decreases from p to around $\binom{5}{3}p^3 = 10p^3$.

The problem with this redundant encoding is that it is inefficient. The longer the string of 0's or 1's you replace each original bit is, the smaller the probability of error, but the more bits you have to send. For a given p, to get a probability of error ϵ in decoding arbitrarily small, you need to send $k(\epsilon)$ copies of each bit, and $k(\epsilon)$ tends to infinity as ϵ tends to 0. Until 1948, most scientists believed this was true; in order to make the probability of error closer to 0, the amount of communication would have to go up indefinitely. In a groundbreaking paper in 1948, Claude Shannon showed that this was not true.

What Shannon showed was that every channel has a *capacity* C. The channel capacity is generally expressed as bits per channel use—a channel use for the binary symmetric channel is sending one bit through the channel. In other words, it is a ratio of the number of bits of the message to the number of bits sent through the channel. If you want to send data at a rate less than C, then you can reduce the error rate to zero by encoding blocks of $(C-\epsilon)n$ bits into a codeword consisting of n bits, and sending it through the channel. You can reduce the error to zero and make the rate arbitrarily close to C by choosing n large enough. If you want to send data at a rate larger than the capacity, your error rate will be close to 1. That is, no matter what your encoding, the message decoded by the receiver will differ from the message encoded by the sender with high probability, and as the length of the message increases, this probability goes to 1.

Here is the notation for the encoding/decoding we will be using. The sender takes a block of Rn bits (the message m), encodes it into some n bits (the codeword c) and send it through the channel. The receiver gets the output of the channel (the received word $\tilde{c}$ — this is still n bits) and decodes it into a putative message $\tilde{m}$ (Rn bits). We will say that this process is successful when $m = \tilde{m}$, and we would like to make the failure rate small; that is, we want $\mathbb{P}(m \neq \tilde{m}) \leq \epsilon$ no matter what m is. Here R is the ratio of the number of bits in the message to the number of bits we encoded, or the *rate* of the code. Note that for any binary channel, we must have $R \leq 1$ (otherwise we would be encoding $Rn > n$ bits into n bits, which is impossible).

2 Binary symmetric channels

We won't state Shannon's theorem formally in its full generality, but focus on the binary symmetric channel. In this case, Shannon's theorem says precisely what the capacity is. It is $1 - H(p)$ where $H(p)$ is the entropy of one bit of our source, i.e., $H(p) = -p\log_2 p - (1-p)\log_2(1-p)$.

Definition 1. A (k,n)-*encoding function* is a function $\text{Enc} : \{0,1\}^k \to \{0,1\}^n$. A (k,n)-*decoding function* is a function $\text{Dec} : \{0,1\}^n \to \{0,1\}^k$. Call a pair of (k,n) encoding and decoding functions an *encoding scheme*. The *rate* of the encoding scheme is precisely k/n.

The rate of an encoding scheme is a measure of how efficiently it uses the channel, with larger being better. For a fixed n, and a fixed rate $0 < R \leq 1$ (so $k = Rn$; formally, you would round up), what we want to consider is the *best* (Rn,n)-encoding scheme. By this, we mean the encoding scheme that minimizes the probability of decoding incorrectly, for any input message: $\max_{m\in\{0,1\}^{Rn}} \mathbb{P}(\tilde{m} \neq m)$ is as small as possible. As such, define λ_n^* to be this error probability of the *best* (Rn,n)-encoding scheme:

$$\lambda^*(R,n) := \min_{\phi \text{ an } (Rn,n)\text{-encoding scheme}} \max_{m\in\{0,1\}^{Rn}} \mathbb{P}(\tilde{m} \neq m).$$

Now we're ready to formally state Shannon's theorem.

Theorem 1. *Consider the binary symmetric channel with parameter p.*

1. *For any $R > 1 - H(p)$, $\lambda^*(R, n) \to 1$ as $n \to \infty$. (In other words, no matter how clever we are with the coding, the probability of decoding correctly goes to zero.)*
2. *For any $R < 1 - H(p)$, $\lambda^*(R, n) \to 0$ as $n \to \infty$. (In other words, there exist good encoding schemes with this choice of R, and we can make the probability of decoding incorrectly as small as we like, as long as we allow n to be large enough.)*

This states that the capacity of the binary symmetric channel is precisely $1 - H(p)$.

We will need the notion of the *Hamming distance*. The Hamming distance $d_H(s, t)$ between two binary strings s and t of the same length is the number of places where the two strings differ. For example, $d_H(10110100, 00111100) = 2$ since these two strings differ in the first bit and the fifth bit.

Before we go into the technical details of the proof, let us give the intuition behind the proof of the first part of the theorem (without any ϵ or δ). The idea is that we will send one of $M = 2^{Rn}$ codewords. If one of these codewords c is put into the channel, the output $\tilde{c}$ will very likely be in a very thin "ring" around the codeword of radius about pn (measuring distance using the Hamming distance). In order that with high probability we decode the output to the correct codeword, for each of these rings, we need to decode most of the codewords in this ring to the center of the ring, c. There are approximately $2^{H(p)n}$ words in each of these rings (remember the analysis of Shannon's noiseless theorem: there are $2^{H(p)n}$ *typical* messages of length n with two symbols, one occuring with probability p, the other with probability $1 - p$), and there are 2^n words altogether, so to make sure that the rings are mostly disjoint, we need that we have at most $2^{(1-H(p))n}$ codewords.

Proof of Theorem 1, part 1. Fix some $\epsilon > 0$. Our goal is to show that for any $R < 1 - H(p)$, that $\lambda^*(R, n) > 1 - \epsilon$ for n sufficiently large. Or equivalently, that if $\lambda^*(R, n) \leq 1 - \epsilon$ for all n, then $R \leq 1 - H(p)$. We will do this by giving an upper bound (depending on n) on the number of possible codewords M in any encoding scheme which decodes every codeword correctly with probability at least ϵ. Relating M to R by $M = 2^{Rn}$, this will give the required bound.

So for the moment, fix n, and fix some encoding scheme with codes of length n and which decodes each code correctly with probability at least ϵ. Let M denote the number of codewords in this code. Now, consider any codeword c. When we send it through the channel, the output is likely going to be a word near c, with the closer messages the more likely. Each possible output $\tilde{c}$ could be decoded to some message $\tilde{m}$, and we would like to make sure that when we put the codeword c corresponding to m into the channel, the probability that the output of the channel decodes to m is at least ϵ. That is, we want

$$\sum_{\tilde{c} \text{ decodes to } m} \mathbb{P}(\tilde{c}|c) \geq \epsilon.$$

The probability that on channel input c, we get output $\tilde{c}$ is

$$\mathbb{P}(\tilde{c}|c) = p^{d_H(c,\tilde{c})}(1-p)^{n-d_H(c,\tilde{c})}.$$

By the law of large numbers (or Chebyshev's inequality), there is some γ so that with probability $1 - \epsilon/2$,

$$(p - \gamma)n \leq d_H(c, \tilde{c}) \leq (p + \gamma)n,$$

where γ goes to 0 as n goes to ∞. Call this set of words the *γ-typical outputs* on input c; they form a "ring" around c. Because with probability $1-\epsilon/2$ the output is typical, and we need to decode the output correctly with probability ϵ, we need to decode a γ-typical output correctly with probability at least $\epsilon/2$ (otherwise even if every output that is not γ-typical was magically decoded correctly, there would be no chance of decoding correctly with probability ϵ). Now, a γ-typical output with highest probability is one which has Hamming distance between $(p-\gamma)n$ and $(p+\gamma)n$ from the codeword, say it is at distance $(p-\alpha)n$ for $-\gamma \le \alpha \le \gamma$. The probability of getting any particular one of these outputs is

$$\begin{aligned} p^{(p-\alpha)n}(1-p)^{(1-p-\alpha)n} &= 2^{((p-\alpha)\log p+(1-p+\alpha)\log(1-p))n} \\ &= 2^{(p\log p+(1-p)\log(1-p))n+\delta' n} \\ &= 2^{(-H(p)+\delta')n}, \end{aligned}$$

where δ' also goes to 0 when δ goes to 0 (and hence α goes to 0). So to make sure the probability of decoding this output is at least ϵ, we need

$$\frac{\epsilon/2}{2^{(-H(p)+\delta')n}}$$

words that decode to the message m. This same argument applies to each $m \in \{0,1\}^{Rn}$. Since no word can decode to more than one message, and there are 2^n words altogether, by the pigeonhole principle we must have

$$M(\epsilon/2)2^{(H(p)-\delta')n} \le 2^n$$

The ϵ term is tiny compared with the δ', so we can absorb them into δ', and we have

$$M \le 2^{n(1-H(p)+\delta'')},$$

for some $\delta'' > 0$ which goes to 0 as $n \to \infty$. Now since $M = 2^{Rn}$, we obtain that $R \le 1-H(p)+\delta''$. But since $\delta'' \to 0$ as $n \to \infty$, we deduce that $R \le 1-H(p)$, and we're done. □

Now for the second part of the theorem. The proof will be *nonconstructive*; it will only show that a good encoding exists, but because it uses the probabilistic method, won't give us an explicit encoding scheme.

Proof of Theorem 1, part 2. The proof will proceed in two main steps:

1. Pick codewords uniformly at random, but twice as many as we will eventually need. It is shown that such a random code is "good on average", in the sense that upon sending a random codeword, it is very likely to be decoded correctly. This is not the end of the story however, since we need to show that every possible codeword can be correctly decoded, with a high probability.

2. After a derandomization step that picks the collection of codewords that is best on average, the "worst half" of the codewords are thrown away, yielding a collection of codewords where every code has a large probability of being correctly decoded.

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_M\}$ be some arbitrary collection of distinct codewords (we'll call this a *codebook*). If we send c_i through the channel, the output is a random code $\tilde{c}_i$. As already discussed, the Hamming distance between c_i and $\tilde{c}_i$ is very likely to be close to np. Choose $\gamma > 0$ as small as possible such that $\mathbb{P}(|d_H(c_i, \tilde{c}_i - np| > \gamma n) < \epsilon/2$, and define

$$\text{Ring}(c_i) = \{c : |d_H(c_i, c) - np| \leq \gamma n\}.$$

So $\tilde{c}_i \in \text{Ring}(c_i)$ with probability at least $1-\epsilon/2$. Now let $E_i(\mathcal{C})$ denote the event that $\tilde{c}_i \in \text{Ring}(c_i)$, but $\tilde{c}_i \notin \text{Ring}(c_j)$ for any $j \neq i$. In this case, upon receiving $\tilde{c}_i$ we can be very confident about decoding it to c_i. If we receive a code that is not in a unique ring, we won't even try to decode it, we just give up. So what we eventually want is a codebook where $\mathbb{P}(E_i(\mathcal{C}))$ is large for every i. Let $\lambda_i(\mathcal{C}) = 1 - \mathbb{P}(E_i(\mathcal{C}))$, the probability that this event does not occur; we want these to all be small.

Now let's choose $\mathcal{C}$ randomly. Let $M = 2 \cdot 2^{Rn}$ (this is twice as big as we want in the end), and let X_i be a uniformly chosen binary string in $\{0,1\}^n$. Let $\mathcal{C} = \{X_1, \ldots, X_M\}$. Suppose now we pick a codeword i uniformly at random from $\{1, 2, \ldots, M\}$, and send it through the channel. What is the probability that we cannot decode it, i.e., that the event $E_i(\mathcal{C})$ does not occur?

For a fixed $\mathcal{C}'$ and j, we know that the probability is $\lambda_j(\mathcal{C}')$. What we are asking for is the *average* value of this probability, over our random choice of $\mathcal{C}$ and i. Let's prove that this average is small.

Lemma 1. *For $\mathcal{C}$, i chosen randomly as described,*

$$\mathbb{E}(\lambda_i(\mathcal{C})) \leq \epsilon, \qquad \textit{if } n \textit{ is large enough.}$$

(Note: the expectation is taken over both the random choice of $\mathcal{C}$ and the random choice of i.)

Proof. With probability at least $1 - \epsilon/2$, $\tilde{c}_i \in \text{Ring}(c_i)$. The main work is to show that the probability that $\tilde{c}_i \in \text{Ring}(c_j)$ for some $j \neq i$ is small. First, fix some arbitrary $j \neq i$ (we will apply a union bound at the end).

Fix some arbitrary $\tilde{c} \in \text{Ring}(c_i)$. The probability that $\tilde{c} \in \text{Ring}(c_j)$ is the same as the probability that $c_j \in \text{Ring}(\tilde{c})$. Because c_j is chosen uniformly at random, this is just the "volume" of $\text{Ring}(\tilde{c})$ divided by the volume of $\{0,1\}^n$. We have seen already in the proof of the first part that the volume of a ring is at most $2^{(H(p)+\delta')n}$, where $\delta' \to 0$ as $n \to \infty$. Thus

$$\mathbb{P}(\tilde{c} \in \text{Ring}(c_j)) = \mathbb{P}(c_j \in \text{Ring}(\tilde{c})) \leq 2^{(H(p)+\delta'-1)n}.$$

This is true for any $\tilde{c}$ in $\text{Ring}(c_i)$, and so we can deduce that

$$\mathbb{P}(\tilde{c}_i \in \text{Ring}(c_j) | \tilde{c}_i \in \text{Ring}(c_i)) \leq 2^{(H(p)+\delta'-1)n}.$$

Now take a union bound over all $M-1$ possible choices of j, to deduce that

$$\mathbb{P}(\exists j \neq i : \tilde{c}_i \in \text{Ring}(c_j) | \tilde{c}_i \in \text{Ring}(c_i)) \leq M 2^{(H(p)+\delta'-1)n}.$$

Thus

$$\mathbb{P}(E_i(\mathcal{C}) \text{ does not occur}) \leq M 2^{(H(p)+\delta'-1)n} + \epsilon/2 = 2^{1+(R+H(p)+\delta'-1)n} + \epsilon/2.$$

But $\delta' \to 0$ as $n \to \infty$; since $R < 1 - H(p)$, it follows that $R - H(p) + \delta' + 1 < 0$ for n large enough. The first term then goes to zero with n, and so for n large enough,

$$\mathbb{P}(E_i(\mathcal{C}) \text{ does not occur}) < \epsilon.$$

This is precisely what we wanted. □

Choose n large enough so that the conclusion of the lemma holds. Let us rewrite this conclusion, $\mathbb{E}(\lambda_i(\mathcal{C}))$ written out explicitly as an average:

$$\frac{1}{\#\text{possible codebooks}} \sum_{\text{possible codebooks } \mathcal{C}} \frac{1}{M} \sum_{i=1}^{M} \lambda_i(\mathcal{C}) \leq \epsilon.$$

There must be some choice of codebook $\mathcal{C}^*$ which does better than average, i.e., for which

$$\frac{1}{M} \sum_{i=1}^{M} \lambda_i(\mathcal{C}^*) \leq \epsilon.$$

So we have found a codebook $\mathcal{C}^* = \{c_1, c_2, \ldots, c_M\}$ that is "good on average"; if we send a random codeword through the channel, we're very likely to be able to decode it. But this doesn't mean that this is true for every codeword; it's possible that $\lambda_j(\mathcal{C}^*)$ is still large for some choices of j.

So let $\mathcal{C}'$ be the codebook consisting of just the best half of the codewords in $\mathcal{C}^*$: those with the smallest values of $\lambda_i(\mathcal{C}^*)$. Then $\lambda_i(\mathcal{C}^*) \leq 2\epsilon$ for all $c_i \in \mathcal{C}'$; for if not, then at least half the codewords in $\mathcal{C}^*$ have $\lambda_i(\mathcal{C}^*) > 2\epsilon$, and we would have

$$\frac{1}{M} \sum_{i=1}^{M} \lambda_i(\mathcal{C}^*) > \frac{1}{M} \cdot (M/2)2\epsilon = \epsilon,$$

a contradiction.

So $\mathcal{C}'$ provides a good code (where the probability of an error is at most 2ϵ, where we can choose ϵ as small as we like) with 2^{Rn} codewords, and so it can be used as an (Rn, n)-encoding scheme. $\square$