Chapter 31

Shannon’s theorem

By Sariel Har-Peled, August 31, 2023

31.1. Coding: Shannon’s Theorem

We are interested in the problem sending messages over a noisy channel. We will assume that the channel noise is behave “nicely”.

Definition 31.1.1. The input to a binary symmetric channel with parameter $p$ is a sequence of bits $x_1, x_2, \ldots$, and the output is a sequence of bits $y_1, y_2, \ldots$, such that $\mathbb{P}[x_i = y_i] = 1 - p$ independently for each $i$.

Translation: Every bit transmitted have the same probability to be flipped by the channel. The question is how much information can we send on the channel with this level of noise. Naturally, a channel would have some capacity constraints (say, at most 4,000 bits per second can be sent on the channel), and the question is how to send the largest amount of information, so that the receiver can recover the original information sent.

Now, it's important to realize that handling noise is unavoidable in the real world. Furthermore, there are tradeoffs between channel capacity and noise levels (i.e., we might be able to send considerably more bits on the channel but the probability of flipping [i.e., $p$] might be much larger). In designing a communication protocol over this channel, we need to figure out where is the optimal choice as far as the amount of information sent.

Definition 31.1.2. A $(k, n)$ encoding function $\text{Enc} : \{0, 1\}^k \rightarrow \{0, 1\}^n$ takes as input a sequence of $k$ bits and outputs a sequence of $n$ bits. A $(k, n)$ decoding function $\text{Dec} : \{0, 1\}^n \rightarrow \{0, 1\}^k$ takes as input a sequence of $n$ bits and outputs a sequence of $k$ bits.

Thus, the sender would use the encoding function to send its message, and the receiver would use the transmitted string (with the noise in it), to recover the original message. Thus, the sender starts with a message with $k$ bits, it blow it up to $n$ bits, using the encoding function (to get some robustness to noise), it send it over the (noisy) channel to the receiver. The receiver takes the given (noisy) message with $n$ bits, and use the decoding function to recover the original $k$ bits of the message.

\[^{\text{a}}\text{This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.}\]
Naturally, we would like $k$ to be as large as possible (for a fixed $n$), so that we can send as much information as possible on the channel.

The following celebrated result of Shannon\footnote{Claude Elwood Shannon (April 30, 1916 - February 24, 2001), an American electrical engineer and mathematician, has been called “the father of information theory”} in 1948 states exactly how much information can be sent on such a channel.

**Theorem 31.1.3 (Shannon’s theorem).** For a binary symmetric channel with parameter $p < 1/2$ and for any constants $\delta, \gamma > 0$, where $n$ is sufficiently large, the following holds:

(i) For an $k \leq n(1 - \mathbb{H}(p) - \delta)$ there exists $(k, n)$ encoding and decoding functions such that the probability the receiver fails to obtain the correct message is at most $\gamma$ for every possible $k$-bit input messages.

(ii) There are no $(k, n)$ encoding and decoding functions with $k \geq n(1 - \mathbb{H}(p) + \delta)$ such that the probability of decoding correctly is at least $\gamma$ for a $k$-bit input message chosen uniformly at random.

**31.1.0.1. Intuition behind Shannon’s theorem**

Let assume the senders has sent a string $S = s_1s_2\ldots s_n$. The receiver got a string $T = t_1t_2\ldots t_n$, where $p = \mathbb{P}[t_i \neq s_i]$, for all $i$. In particular, let $U$ be the Hamming distance between $S$ and $T$; that is, $U = \sum[s_i \neq t_i]$. Under our assumptions $\mathbb{E}[U] = pn$, and $U$ is a binomial variable. By Chernoff inequality, we know that $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is some tiny constant. So lets assume this indeed happens. This means that $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$. This ring has

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2^{n H((1 + \delta)p)} \leq \alpha = 2^{n H((1 + \delta)p)}.$$

Let us pick as many rings as possible in the hypercube so that they are disjoint: $R_1, \ldots, R_\kappa$. If somehow magically, every word in the hypercube would be covered, then we could use all the possible $2^n$ codewords, then the number of rings $\kappa$ we would pick would be at least

$$\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2^{n H((1 + \delta)p)}} \approx 2^{n(1 - H((1 + \delta)p))}.$$

In particular, consider all possible strings of length $k$ such that $2^k \leq \kappa$. We map the $i$th string in $\{0,1\}^k$ to the center $C_i$ of the $i$th ring $R_i$. Assuming that when we send $C_i$, the receiver gets a string in $R_i$, then the decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to. Now, observe that

$$k = \lfloor \log \kappa \rfloor = n(1 - H((1 + \delta)p)) \approx n(1 - H(p)),$$

as desired.

**31.1.0.2. What is wrong with the above?**

The problem is that we can not find such a large set of disjoint rings. The reason is that when you pack rings (or balls) you are going to have wasted spaces around. To overcome this, we would allow rings to overlap somewhat. That makes things considerably more involved. The details follow.
31.2. Proof of Shannon’s theorem

The proof is not hard, but requires some care, and we will break it into parts.

31.2.1. How to encode and decode efficiently

31.2.1.1. The scheme

Our scheme would be simple. Pick \( k \leq n(1 - \mathbb{H}(p) - \delta) \). For any number \( i = 0, \ldots, \widetilde{K} = 2^{k+1} - 1 \), randomly generate a binary string \( Y_i \) made out of \( n \) bits, each one chosen independently and uniformly. Let \( Y_0, \ldots, Y_{\widetilde{K}} \) denote these code words. Here, we have

\[
\widetilde{K} = 2^{n(1-\mathbb{H}(p)-\delta)}.
\]

For each of these codewords we will compute the probability that if we send this codeword, the receiver would fail. Let \( X_0, \ldots, X_{K} \), where \( K = 2^k - 1 \), be the \( K \) codewords with the lowest probability to fail. We assign these words to the \( 2^k \) messages we need to encode in an arbitrary fashion.

The decoding of a message \( w \) is done by going over all the codewords, and finding all the codewords that are in (Hamming) distance in the range \([p(1-\varepsilon)n, p(1+\varepsilon)n]\) from \( w \). If there is only a single word \( X_i \) with this property, we return \( i \) as the decoded word. Otherwise, if there are no such words or there is more than one word, the decoder stops and report an error.

31.2.1.2. The proof

Intuition. Let \( S_i \) be all the binary strings (of length \( n \)) such that if the receiver gets this word, it would decipher it to be \( i \) (here are still using the extended codeword \( Y_0, \ldots, Y_{\widetilde{K}} \)). Note, that if we remove some codewords from consideration, the set \( S_i \) just increases in size. Let \( W_i \) be the probability that \( X_i \) was sent, but it was not deciphered correctly. Formally, let \( r \) denote the received word. We have that

\[
W_i = \sum_{r \notin S_i} \mathbb{P}[r \text{ received when } X_i \text{ was sent}].
\]

To bound this quantity, let \( \Delta(x, y) \) denote the Hamming distance between the binary strings \( x \) and \( y \). Clearly, if \( x \) was sent the probability that \( y \) was received is

\[
w(x, y) = p^{\Delta(x, y)}(1 - p)^{n - \Delta(x, y)}.
\]

As such, we have

\[
\mathbb{P}[r \text{ received when } X_i \text{ was sent}] = w(X_i, r).
\]

Let \( S_{i,r} \) be an indicator variable which is 1 if \( r \notin S_i \). We have that

\[
W_i = \sum_{r \notin S_i} \mathbb{P}[r \text{ received when } X_i \text{ was sent}] = \sum_{r \notin S_i} w(X_i, r) = \sum_r S_{i,r} w(X_i, r).
\]

The value of \( W_i \) is a random variable of our choice of \( Y_0, \ldots, Y_{\widetilde{K}} \). As such, its natural to ask what is the expected value of \( W_i \).

Consider the ring

\[
R(r) = \left\{ x \mid (1 - \varepsilon)np \leq \Delta(x, r) \leq (1 + \varepsilon)np \right\},
\]

where \( \varepsilon > 0 \) is a small enough constant. Suppose, that the code word \( Y_i \) was sent, and \( r \) was received. The decoder return \( i \) if \( Y_i \) is the only codeword that falls inside \( R(r) \).
Lemma 31.2.1. Given that $Y_i$ was sent, and $r$ was received and furthermore $r \in R(Y_i)$, then the probability of the decoder failing, is
\[
\tau = \mathbb{P}[r \notin S_i \mid r \in R(Y_i)] \leq \frac{\gamma}{8},
\]
where $\gamma$ is the parameter of Theorem 31.1.3.

Proof: The decoder fails here, only if $R(r)$ contains some other codeword $Y_j (j \neq i)$ in it. As such,
\[
\tau = \mathbb{P}[r \notin S_i \mid r \in R(Y_i)] \leq \mathbb{P}[Y_j \in R(r), \text{ for any } j \neq i] \leq \sum_{j \neq i} \mathbb{P}[Y_j \in R(r)].
\]

Now, we remind the reader that the $Y_j$s are generated by picking each bit randomly and independently, with probability $1/2$. As such, we have
\[
\mathbb{P}[Y_j \in R(r)] = \sum_{m=\lfloor (1+\epsilon)n \rfloor}^{(1+\epsilon)n} \frac{n^m}{2^m} \leq \frac{n}{2^n} \binom{n}{\lfloor (1+\epsilon)n \rfloor},
\]
since $(1+\epsilon)p < 1/2$ (for $\epsilon$ sufficiently small), and as such the last binomial coefficient in this summation is the largest. By Corollary 31.3.2 (i), we have
\[
\mathbb{P}[Y_j \in R(r)] \leq \frac{n}{2^n} \binom{n}{\lfloor (1+\epsilon)n \rfloor} \leq \frac{n}{2^n} 2^{n\mathcal{H}(1+\epsilon)p} = n^{2^{n\mathcal{H}(1+\epsilon)p-1}}.
\]

As such, we have
\[
\tau = \mathbb{P}[r \notin S_i \mid r \in R(Y_i)] \leq \sum_{j \neq i} \mathbb{P}[Y_j \in R(r)]
\]
\[
\leq K \mathbb{P}[Y_1 \in R(r)] \leq 2^{k+1} n^{2^{n\mathcal{H}(1+\epsilon)p-1}}
\]
\[
\leq n^{2^{n(1-\mathcal{H}(p)-\delta)+1}} n^{2^{n\mathcal{H}(1+\epsilon)p-1}} \leq n^{2^{n(\mathcal{H}(1+\epsilon)p-\mathcal{H}(p)-\delta)+1}}
\]
since $k \leq n(1-\mathcal{H}(p)-\delta)$. Now, we choose $\epsilon$ to be a small enough constant, so that the quantity $\mathcal{H}(1+\epsilon)p - \mathcal{H}(p) - \delta$ is equal to some (absolute) negative (constant), say $-\beta$, where $\beta > 0$. Then, $\tau \leq n^{2^{-\beta n+1}}$. and choosing $n$ large enough, we can make $\tau$ smaller than $\gamma/2$, as desired. As such, we just proved that
\[
\tau = \mathbb{P}[r \notin S_i \mid r \in R(Y_i)] \leq \frac{\gamma}{2}.
\]

Lemma 31.2.2. We have, that $\sum_{r \notin R(Y_i)} w(Y_i, r) \leq \gamma/8$, where $\gamma$ is the parameter of Theorem 31.1.3.

Proof: This quantity, is the probability of sending $Y_i$ when every bit is flipped with probability $p$, and receiving a string $r$ such that more than $\epsilon pn$ bits where flipped. But this quantity can be bounded using the Chernoff inequality. Let $Z = \Delta(Y_i, r)$, and observe that $\mathbb{E}[Z] = pn$, and it is the sum of $n$ independent indicator variables. As such
\[
\sum_{r \notin R(Y_i)} w(Y_i, r) = \mathbb{P}[|Z - \mathbb{E}[Z]| > \epsilon pn] \leq 2 \exp\left(-\frac{\epsilon^2}{4} pn\right) < \frac{\gamma}{4},
\]
since $\epsilon$ is a constant, and for $n$ sufficiently large.

Lemma 31.2.3. For any $i$, we have $\mu = \mathbb{E}[W_i] \leq \gamma/4$, where $\gamma$ is the parameter of Theorem 31.1.3.
Proof: By linearity of expectations, we have
\[
\mu = \mathbb{E}[W_i] = \mathbb{E}\left[\sum_r S_{i,r} w(Y_i, r)\right] = \sum_r \mathbb{E}[S_{i,r} w(Y_i, r)]
\]
\[
= \sum_r \mathbb{E}[S_{i,r}] w(Y_i, r) = \sum_r P[x \notin S_i] w(Y_i, r),
\]
since \(S_{i,r}\) is an indicator variable. Setting, \(\tau = \mathbb{P}[r \notin S_i \mid r \in R(Y_i)]\) and since \(\sum_r w(Y_i, r) = 1\), we get
\[
\mu = \sum_{r \in R(Y_i)} \mathbb{P}[x \notin S_i] w(Y_i, r) + \sum_{r \notin R(Y_i)} \mathbb{P}[x \notin S_i] w(Y_i, r)
\]
\[
= \sum_{r \in R(Y_i)} \mathbb{P}[x \notin S_i \mid r \in R(Y_i)] w(Y_i, r) + \sum_{r \notin R(Y_i)} \mathbb{P}[x \notin S_i] w(Y_i, r)
\]
\[
\leq \sum_{r \in R(Y_i)} \tau \cdot w(Y_i, r) + \sum_{r \notin R(Y_i)} w(Y_i, r) \leq \tau + \sum_{r \notin R(Y_i)} w(Y_i, r) \leq \frac{\gamma}{4} + \frac{\gamma}{4} = \frac{\gamma}{2}.
\]

Now, the receiver got \(r\) (when we sent \(Y_i\)), and it would miss encode it only if (i) \(r\) is outside of \(R(Y_i)\), or \(R(r)\) contains some other codeword \(Y_j\) (\(j \neq i\)) in it. As such,
\[
\tau = \mathbb{P}[r \notin S_i \mid r \in R(Y_i)] \leq \mathbb{P}[Y_j \in R(r), \text{ for any } j \neq i] \leq \sum_{j \neq i} \mathbb{P}[Y_j \in R(r)].
\]

Now, we remind the reader that the \(Y_j\)’s are generated by picking each bit randomly and independently, with probability 1/2. As such, we have
\[
\mathbb{P}[Y_j \in R(r)] = \sum_{m=(1+\varepsilon)n}^{(1+\varepsilon)n} \binom{n}{m} \leq \frac{n}{2^n} \binom{n}{n(1+\varepsilon)n_p},
\]
since \((1+\varepsilon)n_p < 1/2\) (for \(\varepsilon\) sufficiently small), and as such the last binomial coefficient in this summation is the largest. By Corollary 31.3.2 (i), we have
\[
\mathbb{P}[Y_j \in R(r)] \leq \frac{n}{2^n} \binom{n}{n(1+\varepsilon)n_p} \leq \frac{n}{2^n} 2^{n \mathbb{H}((1+\varepsilon)n_p)} = n 2^{n \mathbb{H}((1+\varepsilon)n_p) - 1}.
\]

As such, we have
\[
\tau = \mathbb{P}[r \notin S_i \mid r \in R(Y_i)] \leq \sum_{j \neq i} \mathbb{P}[Y_j \in R(r)] \leq K \mathbb{P}[Y_1 \in R(r)] \leq 2^{k+1} n 2^{n \mathbb{H}((1+\varepsilon)n_p) - 1}
\]
\[
\leq 2^{n (1 - \mathbb{H}(p) - \delta) + 1 + n \mathbb{H}((1+\varepsilon)n_p) - 1} \leq 2^{n \mathbb{H}((1+\varepsilon)n_p) - \mathbb{H}(p) - \delta + 1}
\]
since \(k \leq n (1 - \mathbb{H}(p) - \delta)\). Now, we choose \(\varepsilon\) to be a small enough constant, so that the quantity \(\mathbb{H}((1+\varepsilon)n_p) - \mathbb{H}(p) - \delta\) is negative (constant). Then, choosing \(n\) large enough, we can make \(\tau\) smaller than \(\gamma/2\), as desired. As such, we just proved that
\[
\tau = \mathbb{P}[r \notin S_i \mid r \in R(Y_i)] \leq \frac{\gamma}{2}.
\]

In the following, we need the following trivial (but surprisingly deep) observation.
**Observation 31.2.4.** For a random variable $X$, if $\mathbb{E}[X] \leq \psi$, then there exists an event in the probability space, that assigns $X$ a value $\leq \mu$.

This holds, since $\mathbb{E}[X]$ is just the average of $X$ over the probability space. As such, there must be an event in the universe where the value of $X$ does not exceed its average value.

The above observation is one of the main tools in a powerful technique to proving various claims in mathematics, known as the *probabilistic method*.

**Lemma 31.2.5.** For the codewords $X_0, \ldots, X_K$, the probability of failure in recovering them when sending them over the noisy channel is at most $\gamma$.

**Proof:** We just proved that when using $Y_0, \ldots, Y_{\hat{K}}$, the expected probability of failure when sending $Y_i$, is $\mathbb{E}[W_i] \leq \gamma_2$, where $\hat{K} = 2^{k+1} - 1$. As such, the expected total probability of failure is

$$\mathbb{E}\left[ \sum_{i=0}^{\hat{K}} W_i \right] = \sum_{i=0}^{\hat{K}} \mathbb{E}[W_i] \leq \frac{\gamma}{2} 2^{k+1} = \gamma 2^k,$$

by **Lemma 31.2.3** (here we are using the facts that all the random variables we have are symmetric and behave in the same way). As such, by **Observation 31.2.4**, there exist a choice of $Y_i$s, such that

$$\sum_{i=0}^{\hat{K}} W_i \leq 2^k \gamma.$$

Now, we use a similar argument used in proving Markov’s inequality. Indeed, the $W_i$ are always positive, and it cannot be that $2^k$ of them have value larger than $\gamma$, because in the summation, we will get that

$$\sum_{i=0}^{\hat{K}} W_i > 2^k \gamma.$$

Which is a contradiction. As such, there are $2^k$ codewords with failure probability smaller than $\gamma$. We set our $2^k$ codeword to be these words. Since we picked only a subset of the codewords for our code, the probability of failure for each codeword shrinks, and is at most $\gamma$.

**Lemma 31.2.5** concludes the proof of the constructive part of Shannon’s theorem.

### 31.2.2. Lower bound on the message size

We omit the proof of this part.

### 31.3. From previous lectures

**Lemma 31.3.1.** Suppose that $nq$ is integer in the range $[0, n]$. Then $\frac{2^{n\mathbb{E}(q)}}{n+1} \leq \binom{n}{nq} \leq 2^{n\mathbb{E}(q)}$.

**Lemma 31.3.1** can be extended to handle non-integer values of $q$. This is straightforward, and we omit the easy details.
Corollary 31.3.2. We have:

(i) \( q \in [0, 1/2] \Rightarrow \left( \frac{n}{\lfloor nq \rfloor} \right) \leq 2^{nH(q)} \).

(ii) \( q \in [1/2, 1] \Rightarrow \left( \frac{n}{\lceil nq \rceil} \right) \leq 2^{nH(q)} \).

(iii) \( q \in [1/2, 1] \Rightarrow \frac{2^{nH(q)}}{n+1} \leq \left( \frac{n}{\lfloor nq \rfloor} \right) \).

(iv) \( q \in [0, 1/2] \Rightarrow \frac{2^{nH(q)}}{n+1} \leq \left( \frac{n}{\lceil nq \rceil} \right) \).

Theorem 31.3.3. Suppose that the value of a random variable \( X \) is chosen uniformly at random from the integers \( \{0, \ldots, m-1\} \). Then there is an extraction function for \( X \) that outputs on average at least \( \lfloor \lg m \rfloor - 1 = \lceil H(X) \rceil - 1 \) independent and unbiased bits.

31.4. Bibliographical Notes

The presentation here follows [MU05, Sec. 9.1-Sec 9.3].

Bibliography