Chapter 29

Shannon's theorem

By Sariel Har-Peled, November 1, 2015[①] Version: 0.1

> "This has been a novel about some people who were punished entirely too much for what they did. They wanted to have a good time, but they were like children playing in the street; they could see one after another of them being killed - run over, maimed, destroyed - but they continued to play anyhow. We really all were very happy for a while, sitting around not toiling but just bullshitting and playing, but it was for such a terrible brief time, and then the punishment was beyond belief; even when we could see it, we could not believe it."
>
> – – A Scanner Darkly, Philip K. Dick.

29.1. 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 29.1.1. 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 $\mathbf{Pr}[x_i = y_i] = 1 - p$ 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, its 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 29.1.2. A (k, n) ***encoding function*** $\mathtt{Enc} : \{0,1\}^k \to \{0,1\}^n$ takes as input a sequence of k bits and outputs a sequence of n bits. A (k, n) ***decoding function*** $\mathtt{Dec} : \{0,1\}^n \to \{0,1\}^k$ 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.

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[2] in 1948 states exactly how much information can be sent on such a channel.

Theorem 29.1.3 (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 $k \le n(1 - \mathbb{H}(p) - \delta)$ there exists (k, n) encoding and decoding functions such that the probability the receiver fails to obtain the correct message is at most γ for every possible k-bit input messages.

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

29.1.0.1. Intuition behind Shanon's theorem

Let assume the senders has sent a string $S = s_1 s_2 \ldots s_n$. The receiver got a string $T = t_1 t_2 \ldots t_n$, where $p = \mathbf{Pr}[t_i \neq s_i]$, for all i. In particular, let U be the Hamming distance between S and T; that is, $U = \sum_i [s_i \neq t_i]$. Under our assumptions $\mathbf{E}[U] = pn$, and U is a binomial variable. By Chernoff inequality, we know that $U \in [(1-\delta)np, (1+\delta)np]$ with high probability, where δ 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

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

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 κ we would pick would be at least

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

In particular, consider all possible strings of length k such that $2^k \le \kappa$. We map the ith string in $\{0,1\}^k$ to the center C_i of the ith 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

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

as desired.

29.1.0.2. 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.

[2] Claude Elwood Shannon (April 30, 1916 - February 24, 2001), an American electrical engineer and mathematician, has been called "the father of information theory".

29.2. Proof of Shannon's theorem

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

29.2.1. How to encode and decode efficiently

29.2.1.1. The scheme

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

$$\widehat{K} = 2^{n(1-\mathbb{H}(p)-\delta)}.$$

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.

29.2.1.2. 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 $Y_0, \ldots, Y_{\widehat{K}}$). 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

$$W_i = \sum_{r \notin S_i} \mathbf{Pr}\Big[r \text{ received when } X_i \text{ was sent}\Big].$$

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

$$\mathbf{Pr}[r \text{ received when } X_i \text{ was sent}] = w(X_i, r).$$

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

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

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

Consider the ring

$$R(r) = \Big\{ x \;\Big|\; (1 - \varepsilon)np \leq \Delta(x, r) \leq (1 + \varepsilon)np \Big\},$$

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 29.2.1. *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*

$$\tau = \mathbf{Pr}\Big[r \notin S_i \;\Big|\; r \in R(Y_i)\Big] \leq \frac{\gamma}{8},$$

where γ is the parameter of Theorem 29.1.3.

Proof: The decoder fails here, only if $R(r)$ contains some other codeword Y_j ($j \neq i$) in it. As such,

$$\tau = \mathbf{Pr}\Big[r \notin S_i \;\Big|\; r \in R(Y_i)\Big] \leq \mathbf{Pr}\Big[Y_j \in R(r), \text{ for any } j \neq i\Big] \leq \sum_{j \neq i} \mathbf{Pr}\Big[Y_j \in R(r)\Big].$$

Now, we remind the reader that the Y_js are generated by picking each bit randomly and independently, with probability $1/2$. As such, we have

$$\mathbf{Pr}\Big[Y_j \in R(r)\Big] = \sum_{m=(1-\varepsilon)np}^{(1+\varepsilon)np} \frac{\binom{n}{m}}{2^n} \leq \frac{n}{2^n}\binom{n}{\lfloor (1+\varepsilon)np \rfloor},$$

since $(1+\varepsilon)p < 1/2$ (for ε sufficiently small), and as such the last binomial coefficient in this summation is the largest. By Corollary 29.3.2 (i), we have

$$\mathbf{Pr}\Big[Y_j \in R(r)\Big] \leq \frac{n}{2^n}\binom{n}{\lfloor (1+\varepsilon)np \rfloor} \leq \frac{n}{2^n} 2^{n\mathbb{H}((1+\varepsilon)p)} = n2^{n(\mathbb{H}((1+\varepsilon)p)-1)}.$$

As such, we have

$$\begin{aligned}
\tau = \mathbf{Pr}\Big[r \notin S_i \;\Big|\; r \in R(Y_i)\Big] &\leq \sum_{j \neq i} \mathbf{Pr}\Big[Y_j \in R(r)\Big] \\
&\leq \widehat{K}\, \mathbf{Pr}\Big[Y_1 \in R(r)\Big] \leq 2^{k+1} n 2^{n(\mathbb{H}((1+\varepsilon)p)-1)} \\
&\leq n2^{n(1-\mathbb{H}(p)-\delta)+1} n 2^{n(\mathbb{H}((1+\varepsilon)p)-1)} \leq n2^{n\left(\mathbb{H}((1+\varepsilon)p)-\mathbb{H}(p)-\delta\right)+1}
\end{aligned}$$

since $k \leq n(1-\mathbb{H}(p)-\delta)$. Now, we choose ε to be a small enough constant, so that the quantity $\mathbb{H}((1+\varepsilon)p) - \mathbb{H}(p) - \delta$ is equal to some (absolute) negative (constant), say $-\beta$, where $\beta > 0$. Then, $\tau \leq n2^{-\beta n+1}$, and choosing n large enough, we can make τ smaller than $\gamma/2$, as desired. As such, we just proved that

$$\tau = \mathbf{Pr}\Big[r \notin S_i \;\Big|\; r \in R(Y_i)\Big] \leq \frac{\gamma}{2}.$$

■

Lemma 29.2.2. *We have, that $\sum_{r \notin R(Y_i)} w(Y_i, r) \leq \gamma/8$, where γ is the parameter of Theorem 29.1.3.*

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 εpn bits where flipped. But this quantity can be bounded using the Chernoff inequality. Let $Z = \Delta(Y_i, r)$, and observe that $\mathbf{E}[Z] = pn$, and it is the sum of n independent indicator variables. As such

$$\sum_{r \notin R(Y_i)} w(Y_i, r) = \mathbf{Pr}[|Z - \mathbf{E}[Z]| > \varepsilon pn] \leq 2\exp\left(-\frac{\varepsilon^2}{4} pn\right) < \frac{\gamma}{4},$$

since ε is a constant, and for n sufficiently large. ■

Lemma 29.2.3. *For any i, we have $\mu = \mathbf{E}[W_i] \leq \gamma/4$, where γ is the parameter of Theorem 29.1.3.*

Proof: By linearity of expectations, we have

$$\begin{aligned}\mu = \mathbf{E}[W_i] &= \mathbf{E}\left[\sum_r \overline{S_{i,r}} w(Y_i, r)\right] = \sum_r \mathbf{E}\left[\overline{S_{i,r}} w(Y_i, r)\right] \\ &= \sum_r \mathbf{E}\left[\overline{S_{i,r}}\right] w(Y_i, r) = \sum_r \mathbf{Pr}[x \notin S_i]\, w(Y_i, r),\end{aligned}$$

since $\overline{S_{i,r}}$ is an indicator variable. Setting, $\tau = \mathbf{Pr}\left[r \notin S_i \;\middle|\; r \in R(Y_i)\right]$ and since $\sum_r w(Y_i, r) = 1$, we get

$$\begin{aligned}\mu &= \sum_{r \in R(Y_i)} \mathbf{Pr}[x \notin S_i]\, w(Y_i, r) + \sum_{r \notin R(Y_i)} \mathbf{Pr}[x \notin S_i]\, w(Y_i, r) \\ &= \sum_{r \in R(Y_i)} \mathbf{Pr}\left[x \notin S_i \;\middle|\; r \in R(Y_i)\right] w(Y_i, r) + \sum_{r \notin R(Y_i)} \mathbf{Pr}[x \notin S_i]\, w(Y_i, r) \\ &\leq \sum_{r \in R(Y_i)} \tau \cdot w(Y_i, r) + \sum_{r \notin R(Y_i)} w(Y_i, r) \leq \tau + \sum_{r \notin R(Y_i)} w(Y_i, r) \leq \frac{\gamma}{4} + \frac{\gamma}{4} = \frac{\gamma}{2}.\end{aligned}$$

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,

$$\tau = \mathbf{Pr}\left[r \notin S_i \;\middle|\; r \in R(Y_i)\right] \leq \mathbf{Pr}\left[Y_j \in R(r), \text{ for any } j \neq i\right] \leq \sum_{j \neq i} \mathbf{Pr}\left[Y_j \in R(r)\right].$$

Now, we remind the reader that the Y_js are generated by picking each bit randomly and independently, with probability $1/2$. As such, we have

$$\mathbf{Pr}\left[Y_j \in R(r)\right] = \sum_{m=(1-\varepsilon)np}^{(1+\varepsilon)np} \frac{\binom{n}{m}}{2^n} \leq \frac{n}{2^n}\binom{n}{\lfloor (1+\varepsilon)np \rfloor},$$

since $(1+\varepsilon)p < 1/2$ (for ε sufficiently small), and as such the last binomial coefficient in this summation is the largest. By Corollary 29.3.2 (i), we have

$$\mathbf{Pr}\left[Y_j \in R(r)\right] \leq \frac{n}{2^n}\binom{n}{\lfloor (1+\varepsilon)np \rfloor} \leq \frac{n}{2^n} 2^{n\mathbb{H}((1+\varepsilon)p)} = n 2^{n(\mathbb{H}((1+\varepsilon)p)-1)}.$$

As such, we have

$$\begin{aligned}\tau = \mathbf{Pr}\left[r \notin S_i \;\middle|\; r \in R(Y_i)\right] &\leq \sum_{j \neq i} \mathbf{Pr}\left[Y_j \in R(r)\right]. \leq \widehat{K}\, \mathbf{Pr}[Y_1 \in R(r)] \leq 2^{k+1} n 2^{n(\mathbb{H}((1+\varepsilon)p)-1)} \\ &\leq n 2^{n(1-\mathbb{H}(p)-\delta)+1+n(\mathbb{H}((1+\varepsilon)p)-1)} \leq n 2^{n\left(\mathbb{H}((1+\varepsilon)p)-\mathbb{H}(p)-\delta\right)+1}\end{aligned}$$

since $k \leq n(1 - \mathbb{H}(p) - \delta)$. Now, we choose ε to be a small enough constant, so that the quantity $\mathbb{H}((1+\varepsilon)p) - \mathbb{H}(p) - \delta$ is negative (constant). Then, choosing n large enough, we can make τ smaller than $\gamma/2$, as desired. As such, we just proved that

$$\tau = \mathbf{Pr}\left[r \notin S_i \;\middle|\; r \in R(Y_i)\right] \leq \frac{\gamma}{2}.$$

■

In the following, we need the following trivial (but surprisingly deep) observation.

Observation 29.2.4. *For a random variable X, if $\mathbf{E}[X] \leq \psi$, then there exists an event in the probability space, that assigns X a value $\leq \mu$.*

This holds, since $\mathbf{E}[X]$ 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 29.2.5. *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 γ.*

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

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

by Lemma 29.2.3 (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 29.2.4, there exist a choice of Y_is, such that

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

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

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

Which is a contradiction. As such, there are 2^k codewords with failure probability smaller than γ. 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 γ. ∎

Lemma 29.2.5 concludes the proof of the constructive part of Shannon's theorem.

29.2.2. Lower bound on the message size

We omit the proof of this part.

29.3. From previous lectures

Lemma 29.3.1. *Suppose that nq is integer in the range $[0, n]$. Then* $\dfrac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{nq} \leq 2^{n\mathbb{H}(q)}$.

Lemma 29.3.1 can be extended to handle non-integer values of q. This is straightforward, and we omit the easy details.

Corollary 29.3.2. *We have:* *(i)* $q \in [0, 1/2] \Rightarrow \binom{n}{\lfloor nq \rfloor} \leq 2^{n\mathbb{H}(q)}$. *(ii)* $q \in [1/2, 1]$ $\binom{n}{\lceil nq \rceil} \leq 2^{n\mathbb{H}(q)}$. *(iii)* $q \in [1/2, 1] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{\lfloor nq \rfloor}$. *(iv)* $q \in [0, 1/2] \Rightarrow \frac{2^{n\mathbb{H}(q)}}{n+1} \leq \binom{n}{\lceil nq \rceil}$.

Theorem 29.3.3. *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 $\lfloor \lg m \rfloor - 1 = \lfloor \mathbb{H}(X) \rfloor - 1$ independent and unbiased bits.*

29.4. Bibliographical Notes

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

Bibliography

[MU05] M. Mitzenmacher and U. Upfal. *Probability and Computing – randomized algorithms and probabilistic analysis*. Cambridge, 2005.