CS 229r Information Theory in Computer Science Feb 19, 2019

Lecture 7

Instructor: Madhu Sudan *Scribes: Jane Ahn, Tristan Yang*

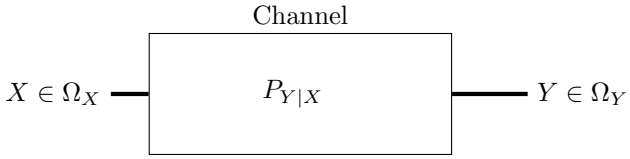
Outline

1. Converse Coding Theorems
2. Efficiency in Coding
3. Linear Coding and Linear Compression

1 Converse Coding Theorems

1.1 Review of Channel Coding

Recall from last week: a general channel takes as input some $X \in \Omega_X$ and outputs some $Y \in \Omega_Y$. Its behavior is specified by $P_{Y|X}$. We encode a message $m \in \{0,1\}^k$ with an encoding function $E_n : \{0,1\}^k \to \Omega_X^n$ and recover the decoded message $\hat{m}$ with a decoding function $D_n : \Omega_Y^n \to \{0,1\}^k$, illustrated below:



We define the rate $R = k/n$. The *Capacity* of the channel is defined as:

$$\sup_R \lim_{\varepsilon \to 0} \lim_{n \to \infty} \{\text{communication of } Rn \text{ bits is possible with } \varepsilon\text{-error during } n \text{ uses of channel}\}.$$

Previously, we proved that by simply picking random i.i.d $E_n(m)_i \sim P_X$ over (m, i) we can achieve:

$$R \geq \sup_{P_X} \{I(X;Y)\}$$

This means that $\sup_{P_X}\{I(X;Y)\}$ is a lower bound for the capacity. We now aim to prove that it is also an upper bound, and thus, equal to the capacity.

1.2 Capacity Upper Bound

Let $C_0 = \sup_{P_X}\{I(X;Y)\}$. We have the following theorem:

Theorem 1. *For the Binary Symmetric Channel BSC(p), for all $\varepsilon > 0$ there exists $\delta > 0$ such that if rate $R > C_0 + \varepsilon$,*

$$\Pr[\textit{decoding error}] \geq 1 - \exp(-\delta n)$$

However, instead of proving this we will focus on general channels and prove that the decoding error cannot be $o(1)$:

Theorem 2. *For all $\varepsilon > 0$ there exists $\delta > 0$ such that if rate $R > C_0 + \varepsilon$ then*

$$\Pr[\textit{decoding error}] \geq \delta$$

Proof. Consider the complete encoding/decoding process for message m:

Note that this is a markov chain (i.e. $\hat{m}|Y^n \perp m, X^n$).
Let $\delta = \Pr[m \neq \hat{m}]$. We want to show that $\delta > 0$. Consider $H(m|\hat{m})$. We have that

$$nR = H(m) = H(m|\hat{m}) + I(\hat{m}; m) \tag{1}$$

The first equality $nR = H(m)$ comes from the fact that we're considering a uniformly random message from $\{0,1\}^{nR}$. By the data processing inequality:

$$I(\hat{m}; m) \leq I(Y^n; m) \leq I(Y^n; X^n) = \sum_{i=1}^{n} I(Y_i; X^n, Y_1 \dots Y_{i-1})$$

Now note that

$$(X^n, Y_1 \dots Y_{i-1}) \to X_i \to Y_i$$

is again a markov chain, so

$$I(\hat{m}; m) \leq \sum_{i=1}^{n} I(Y_i; X^n, Y_1 \dots Y_{i-1}) \leq \sum_{i=1}^{n} I(Y_i; X_i) \leq nC_0. \tag{2}$$

(The last inequality comes from the fact that $I(Y_i; X_i) \leq C_0$ no matter the distribution of X_i.)
To deal with the other term in (1), we note that Fano's inequality implies that if $\Pr[m \neq \hat{m}]$ is small, then $H(m|\hat{m})$ is small:

$$H(m|\hat{m}) \leq H(\mathbf{1}_{m \neq \hat{m}}) + \Pr[m \neq \hat{m}] \log(|\{0,1\}^{nR}|) \leq 1 + \delta nR. \tag{3}$$

Applying the bounds from (2) and (3) to (1) allows us to conclude:

$$\begin{aligned} nR \leq 1 + \delta nR + nC_0 &\implies (1-\delta)nR \leq 1 + nC_0 \\ &\implies \delta nR \geq n(R - C_0) - 1 \geq \varepsilon n - 1 \\ &\implies \delta \geq \varepsilon/R - 1/n \end{aligned}$$

□

2 Efficiency in Coding

We've shown that a random encoding reaches the optimal bound, but from an algorithmic efficiency standpoint this is pretty bad. In practice, we want to consider the following:

1. Complexity of designing E_n and decoder (preprocessing)
2. Encoding time/space complexity
3. Decoding time/space complexity

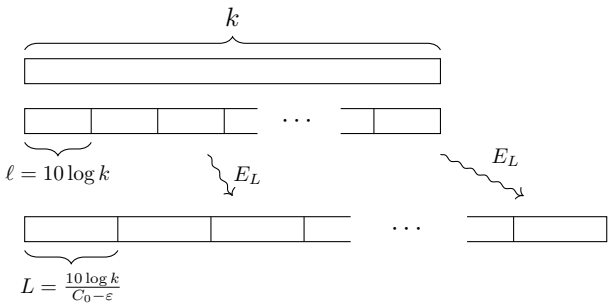
Analysis for random encoder:

1. Space complexity and randomized time complexity to construct E_n is of order 2^{Rn} since there are 2^{Rn} possible messages.

2. Encoding process has 2^{Rn} space complexity to store the lookups. The time complexity is polynomial in n.

3. Decoding process also has 2^{Rn} space complexity. The time complexity is deterministic.

To create a better algorithm, we can leverage the fact that the probability of decoding error in the above case was exponentially low, since we only require that the error approaches 0. We focus only on the case of the binary symmetric channel from now on.
The idea is to divide k-bit sequence into chunks of length e.g. $l = 10 \log k$ and then to apply Shannon's methodology independently to each chunk (encodes block to length $L = l/(C_0 - \varepsilon)$). Now the preprocessing cost and space (including randomness), as well as the encoding and decoding time/space complexities are of order $\exp(L) = \text{poly}(k)$.



We can use the union bound on the error probability

$$\Pr[\exists \text{ block which was decoded incorrectly}] \leq k \Pr[\text{fixed block is decoded incorrectly}].$$

Since the latter probability is exponentially small in k, this will go to 0. In practice, breaking up messages into chunks is used all the time e.g. in CDs.

There are still some issues with the above solution:

- The running time of decoder is at least 1/error prob.
- Each block has to be big enough so that a bit flip is "detectable" to check for errors. Let $\varepsilon = C_0 - R$. We get that the length of each block must at least $1/\varepsilon^2$. So even to achieve 10% of capacity, we would need blocks of length 100 which has running time on the order of 2^{100}.

The first issue was resolved by "Concatenated codes" by Forney '66. The rough idea is that instead of taking the union bound over separate blocks, we use extra redundent encodings ("outer codes") of a 2^{δ} fraction of the blocks to help correct errors. Thus instead of worrying about a single corruption, we worry about corruption of a δ fraction and can use Chernoff bounds.
The second problem persisted until 2008, and was only proved in 2013. The solution uses *Polar Codes*, which will be the focus of the next few lectures.

3 Linear Coding

In *Linear coding*, the encoding map is linear over $\mathbb{F}_2$:

$$E_n(m) = Gm$$

where G is $n \times k$ matrix (m has length k). To get a random linear encoding function we can simply pick G at random.

Claim 3. *A random linear encoding achieves capacity over the binary symmetric channel with parameter p.* In this case, two different messages still have independent encodings, which it turns out is sufficient.

Exercise 4. *Prove the above claim.*

Proof. Problem in the third problem set. □

Linear encoding has several benefits:

1. It only requires polynomial space.
2. It is likely to be injective: *For all* m, Pr[incorrect decoding] is small.
3. Error detection is easy. Given $x \in \mathbb{F}_2^n$, we can easily find out if there exists m such that $x = Gm$.

Proposition 5. *For all full rank $G \in \mathbb{F}_2^{n \times k}$, there exists full rank $H \in \mathbb{F}_2^{m \times n}$ such that $HG = 0$ where $m = n - k$.*

Exercise 6. *Prove the above proposition.*

Proof. Viewed as a linear map $G \colon \mathbb{F}_2^k \to \mathbb{F}_2^n$, the full rank condition translates to the fact that G is injective. Consider the short exact sequence:

$$0 \to \mathbb{F}_2^k \xrightarrow{G} \mathbb{F}_2^n \to \mathbb{F}_2^n / \operatorname{im}(G) \to 0.$$

The dimension of the quotient is $n - k = m$. Therefore we may take an ordered basis for $\mathbb{F}_2^n / \operatorname{im}(G)$ and realize the quotient map as a map

$$H : \mathbb{F}_2^n \twoheadrightarrow \mathbb{F}_2^m,$$

which is an $m \times n$ matrix. Because H is surjective, it has full rank. Moreover, since the kernel of H is the image of G, we have $HG = 0$. □

Exercise 7. *For $x \in \mathbb{F}_2^n$, show that $Hx = 0$ iff there exists m such that $x = Gm$.*

Proof. In our construction, we defined H so that the kernel of H is equal to the image of G. (In fact, this holds for any H satisfying the requirements for H. If $HG = 0$ then $\ker(H) \supset \operatorname{im}(G)$, but the full rank conditions ensure that $\dim \ker(H) = n - m = k = \dim \operatorname{im}(G)$. This shows that $\ker(H) = \operatorname{im}(G)$.) Because $Hx = 0$ means that x is in the kernel of H and existence of m such that $x = Gm$ is x being in the image of G, the two are equivalent. □

Thus the point 3 above is equivalent to finding out if $Hx = 0$. It turns out a good way of constructing a good G is to construct a good H. This means that *linear compression* $\implies$ *linear coding.*

3.1 Efficient Linear Compression for $\mathrm{Bern}(p)^n$

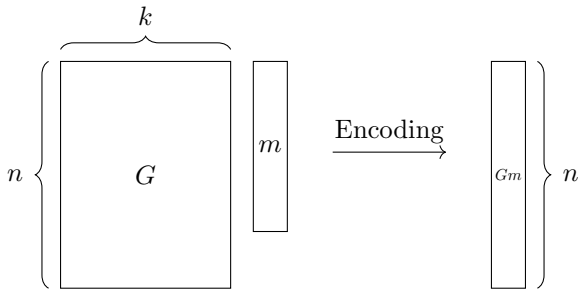
Definition 8. *An **efficient linear compression** for $\mathrm{Bern}(p)^n$ consists of a pair of linear maps $H \in \mathbb{F}_2^{m \times n}$ and $D \in F_2^{n \times m}$ with $m \leq (H(p) + \varepsilon)n$. The efficient compression process maps Z to HZ and decompression maps HZ to $D(HZ)$. In addition, we want*

$$\Pr_{Z \sim \mathrm{Bern}(p)^n}[D(HZ) \neq Z] \leq \delta$$

for some δ.

Proposition 9. *Linear coding over the binary symmetric channel reduces to linear compression for $\mathrm{Bern}(p)^n$.*

Proof. We let G be such that $HG = 0$, and encode m as Gm, as follows:



We receive $Gm + Z$ where $Z \sim \text{Bern}(p)^n$. Recovering m is the same as recovering Z, which we can do by multiplying by H:

$$D(H(Gm + Z)) = D(HZ)$$

This is equal to Z with probability $1 - \delta$. □

The challenge now is to compress n Bern(p) bits to $(H(p) + \varepsilon)n$ bits with decoding time polynomial in n/ε. This is equivalent to encoding to $n_0 = \text{poly}(1/\varepsilon)$ bits with decoding time $\text{poly}(1/\varepsilon)$.