CS 229r Essential Coding Theory April 1, 2020

Lecture 17

Instructor: Madhu Sudan *Scribe: Amal Mattoo, Tristan Yang*

1 Today's Topics

- Polarization Theorem: basically a statement about the behavior of random variables. We will state it and defer the proof.
- Analysis of code: again deferring main theorem.
- Encoding and decoding: explaining algorithms and why they are efficient (if they weren't achieving Shannon capacity efficiently we wouldn't be interested)

2 Review

2.1 Linear Compressor

Recall that given some input $Z_i \overset{i.i.d}{\sim} \text{Bern}(p)$, our goal is to construct a linear compressor $H : \mathbb{F}_2^n \to \mathbb{F}_2^m$ for satisfying:

- $m \leq (H(p) + \varepsilon) \cdot n$.
- There exists a decoder $D : (\mathbb{F}_2 \cup \{?\})^m \to \mathbb{F}_2^n$ with runtime $\text{poly}(1/\varepsilon)$ such that $D(H(Z)) = D(Z \cdot H) = Z$ with high probability over Z.

That is, a "tall and skinny" matrix with not too many columns, roughly the optimal number $H(p)n$.

2.2 Polarization Method

Last time we noted that the two-bit transformation on independent bits $(U, V) \mapsto (U \oplus V, V)$ "moves entropy" from the second bit to the first bit: it is easier to guess the second bit given the first in the image than in the original. Precisely, it is an invertible transformation so total entropy is preserved, but one of the bits $(u \oplus v)$ is more entropic than what we had, so $v|u$ is less entropic, i.e.

$$H(U) \leq H(U \oplus V)$$
$$H(V) \geq H(V|U \oplus V).$$

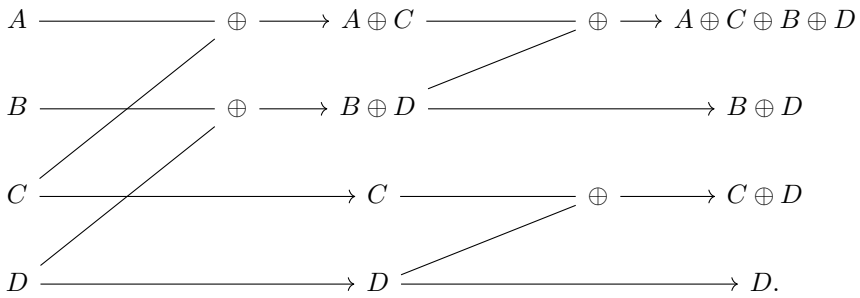
Note that this conditioning is only in the analysis stage, not part of the computation.

While nothing too interesting happens in the 2 bit case, in the multiple bit cases we get the polarization: some bits will be extremely entropic and some will be extremely unentropic. If we know which is which, we can compress the bits by dropping the low entropy bits. Given the rest of the bits, we should be able to recover the dropped bits with high probability. Now let's flesh this out.

The *polarization map* $P_{2n} : \mathbb{F}_2^{2n} \to \mathbb{F}_2^{2n}$ is defined by

$$(U, V) \mapsto (P_n(U \oplus V), P_n(V)).$$

This is illustrated for the case $2n = 4$ by the following diagram:



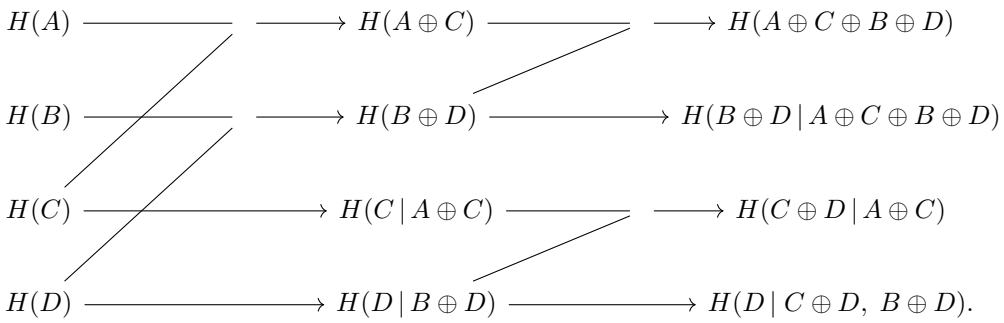
Exercise 1. *Explicitly describe the matrix P_{2n} giving the polarization map (i.e.such that $P_{2n}(Z) = Z \cdot P_{2n}$).*

Sketch. In the case $2n = 4$ the matrix is as below:

$$\begin{bmatrix} A & B & C & D \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} A+C+B+D & B+D & C+D & D \end{bmatrix}$$

In general P_{2n} can be expressed as a binary lower triangular matrix, and by induction we can see that the upper left and lower right quadrants are P_n, while the upper right quadrant is all 0's and the lower left quadrant is 1's in the first column and last row. □

How should we think about the entropies? Globally, letting the inputs be Z and the outputs be W, we are interested in $H(W_i|W_{<i})$ for each i, since this corresponds to recovering each bit of the output given the previous bits. [1] Locally, we condition as in the two-bit case:



The idea is that the local and global pictures are the same, though equating them in general is tedious. But in this case, we see that we can rewrite the conditioning in the last two rows by conditioning on additional independent variables and applying invertible transformations:

$$H(C \oplus D \,|\, A \oplus C, B \oplus D) = H(C \oplus D \,|\, B \oplus D,\ A \oplus C \oplus B \oplus D)$$
$$H(D \,|\, C \oplus D,\ B \oplus D) = H(D \,|\, C \oplus D,\ B \oplus D,\ A \oplus C \oplus B \oplus D).$$

which is indeed what we wanted from the global picture. In general, we thus see that for $W = P_{2n}(Z)$ we can equivalently consider the conditional entropies $H(W_i|W_{<i})$.

[1] A question was raised about why we need to condition in this order of the W_i's and what happens if we take a permutation. But we will see that this analysis achieves Shannon capacity, so regardless of how other permutations behave, we are witnessing the firepower of this fully armed and operational code.

3 Polarization

Now, we set up the machinery necessary to analyze polarization.

Definition 2. *A invertible map $Z \mapsto W$ is an $(\varepsilon, \tau, \delta)$-polarization if*

$$\Pr_{i \in [2n]}[H(W_i|W_{<i}) \in (\tau, 1-\delta)] \leq \varepsilon$$

The idea is that $H(W_i|W_{<i}) \in [0,1]$, but the polarization condition says that it will be in $(\tau, 1-\delta)$ with probability boudned by ε. For our purposes τ and δ are symmetric, and we want to send $\varepsilon, \tau, \delta \to 0$, showing that the entropy is close to 0 or 1. Next week, we will show the following:

Theorem 3 (Polarization Theorem)**.** *For all c, there exists $\alpha > 0$ such that P_{2n} is a $(n^{-\alpha}, n^{-c}, n^{-c})$-polarization.*

Note that the above implies that $n = (1/\varepsilon)^{1/\alpha} = \text{poly}(1/\varepsilon)$. Think of τ and δ as being very small.

It turns out this is enough to get us a good compressor, (and once we show that encoding and decoding are efficiently computable, we will have achieved Shannon capacity):

Lemma 4. *If $P : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is an $(\varepsilon, \tau, \delta)$-polarization, then there exists some set $S \subset [n]$ with*

$$m = |S| \leq (H(p) + (\varepsilon + \delta)) \cdot n$$

such that for $H = P|_S$ there exists some decoder D such that

$$\Pr_Z[D(HZ) \neq Z] \leq n \cdot \tau.$$

Proof. Define the sets

$$\begin{aligned} T &= \{i : H(W_i|W_{<i}) \geq 1-\delta\} \\ B &= \{i : H(W_i|W_{<i}) \leq \tau\} \\ M &= \{i : H(W|W_{<i}) \in (\tau, 1-\delta)\} \end{aligned}$$

(T is the "Top" most entropic bits, B is the "Bottom" and M is the "Middle"). We have that

$$|T| + |B| + |M| = n$$

$$|M| \leq \varepsilon n$$

$$|T| \leq \frac{H(p) \cdot n}{1-\delta} \leq H(p)n + \delta n$$

To see why the last identity is true, note that

$$\begin{aligned} H(W) = n \cdot H(p) &= \sum_i H(W_i|W_{<i}) \\ &\geq \sum_{i \in T} H(W_i|W_{<i}) \\ &\geq (1-\delta)|T|. \end{aligned}$$

from the fact that our operations are invertible starting from n bits of entropy $H(p)$, and applying the chain rule. Then we crudely bound above by $H(p)n + \delta n$.

The claim/belief/hope is that we can erase the bits with low entropy, leaving only T and M. Let $S = T \cup M$, so $|S| \leq (H(p) + \varepsilon + \delta)n$. Let $\overline{S}$ be the complement of S and let $W_{<i,\overline{S}}$ denote the bits of W in indices $< i$ that are in $\overline{S}$. Then we have

$$\begin{aligned} H(W_{\overline{S}}|W_S) &= \sum_{i \in \overline{S}} H(W_i|W_S, W_{<i,\overline{S}}) \\ &\leq \sum_{i=\overline{S}} H(W_i|W_{<i}) \\ &\leq \tau|\overline{S}| \\ &\leq \tau n \end{aligned}$$

and $H(W_{\overline{S}}|W_S) \leq n \cdot \tau$ This following exercise implies that there exists some (inefficient) algorithm $\tilde{D}$ such that

$$\Pr[\tilde{D}(W_S) \neq W_{\overline{S}}] \leq n \cdot \tau$$

Exercise 5. *Let X, Y be jointly distributed. Prove that there exists a predictor $\hat{X} = \hat{X}(Y)$ such that*

$$\Pr_{(X,Y)}[\hat{X} \neq X] \leq H(X|Y).$$

Note: Since there are no requirements on efficiency, $\hat{X}$ will just be the Maximum Likelihood Estimator. This is also known as the converse to Fano's inequality.

Sketch. We have that

$$\begin{aligned} \Pr[\hat{X} \neq X|Y = y] &= 1 - \max_x \Pr[X = x|Y = y] \\ &\leq -\log(\max_x \Pr[X = x|Y = y]) \\ &\leq \mathop{\mathbb{E}}_x[-\log \Pr[X = x|Y = y]] \\ &= H(X|Y = y). \end{aligned}$$

Averaging w.r.t. Y gives the desired result. □

If $H(X|Y)$ is small, then this is a good bound. Applying it with $Y = W_S$ and $X = W_{\overline{S}}$ finishes the lemma; since P is invertible we can recover Z from W. □

At this point what we have left to do is

1. Encode and decode given S
2. Proof of Theorem
3. Compute S

We will do the first now, the second next lecture, and the third never. Computing S requires exponential time, but note that doing so is a preprocessing step that only needs to be done once.

4 Encoding and Decoding

Over the course of this section, we will prove the following.

Theorem 6. *If the set S is given, there are polynomial time encoding and decoding algorithms E and D for this linear compressor such that $D(E(Z)) = Z$ with high probability.*

4.1 Encoding

Our construction of P_{2n} already gives an efficient algorithm for encoding. The runtime is given by the recurrence relation

$$T(2n) = 2T(n) + O(n)$$

which implies $T(n) = O(n \log n)$.

4.2 Recursive Decoding Algorithm

We wish to similarly define a recursive decoding algorithm $D : \hat{W} \mapsto Z$, where $\hat{W}$ is a vector of $\{0, 1, ?\}^{2n}$ with ? denoting bits erased by the compression.

But there is an obstacle: the second half of W will contain most of the erased bits (because those bits are conditioned on more variables they will tend to be lower entropy and so will be erased) so we cannot recursively decode them without considering the first half of W.

Another issue is that, conditioned on the top bits, the bottom bits are independent of each other but no longer identically distributed. As we saw last lecture, given that p is small, if $U \oplus V = 1$ then (U, V) is $(0, 1)$ or $(1, 0)$ and both are equally likely; but if $U \oplus V = 0$ then (U, V) is $(0, 0)$ or $(1, 1)$ and the former is much more likely, so the bias of V is highly dependent on $U \oplus V$. Recall that this method can be used to explicitly compute the bias in each case.

Thus, our recursive decoding function will need the additional data of the biases of each of the bits conditioned on the values of the previous bits that we have decoded. That is, $p_i = W_i | W_{<i} = w_{<i}$. The decoding function will map

$$D : (\hat{W}; p_1, \dots p_n) \mapsto Z$$

outputting the maximum likelihood $Z \sim \text{Bern}(p_1) \times \cdots \times \text{Bern}(p_n)$ satisfying $P_{2n}(Z) = \hat{W}$ on the non-erased coordinates.

Now we define how the algorithm works — it does the "obvious" thing given the data we have.

Algorithm 1 Decoding Algorithm

1: Compute $q_1, \dots, q_{n/2}$ such that $U_i + V_i \sim \text{Bern}(q_i)$, where $Z = (U, V)$, using p_i and $p_{\frac{n}{2}+i}$ as the biases of U_i and V_i.
2: Decode $A = D(\hat{W}_{\text{top}}; q_1, \dots, q_{n/2})$.
3: Compute $r_1, \dots, r_{n/2}$ such that $V_i \,|\, (U_i + V_i = A_i) \sim \text{Bern}(r_i)$, using p_i and $p_{\frac{n}{2}+i}$ as the biases of U_i and V_i and A_i as computed above.
4: Decode $B = D(\hat{W}_{\text{bot}}; r_1, \dots, r_{n/2})$.
5: **return** $(A - B, B)$.

To clarify, we divide $\hat{W}$ in half into $\hat{W}_{\text{top}}$ and $\hat{W}_{\text{bot}}$. We compute the biases q_i of each bit of $\hat{W}_{\text{top}}$ (step i.). Then we recursively decode $\hat{W}_{\text{top}}$ using these biases in our input, recording the output as A (step ii.). Then we compute the biases r_i of each bit of $\hat{W}_{\text{bot}}$ conditioned on the values of $\hat{W}_{\text{top}}$ computed above and recorded in A (step iii). Then we recursively decode $\hat{W}_{\text{bot}}$ using these biases in our input, recording the output as B (step iv.). Note that decoding the top half first (and finding their biases) deals with the issue of the bits of $\hat{W}_{\text{bot}} | \hat{W}_{\text{top}}$ being independent but *not* identically distributed. Finally, we output $(A - B, B)$, noting that this does indeed invert the encoding function.

Finding the biases in steps (i) and (iii) take linear time, since finding the bias q_i of $U_i + V_i$ or the bias r_i of $V_i | (U_i + V_i = A_i)$ is a constant time arithmetic operation for each i, as outlined above. Thus, the whole recursive algorithm runs in $O(n \log n)$ time as desired.

Finally, we need to show that this decoding is accurate. In particular:

Claim 7. *The error probability of D is at most $\sum_{i \in \overline{S}} H(W_i | W_{<i})$.*

and by polarization, this quantity is small. We will see why the claim holds shortly.

4.3 Base Case: Decoding Bits

Although computing the biases solved the issue of our recursion, the "flow of information" is not clear — i.e., how we are using the data of the biases. Of course, this will manifest in the base case $n = 1$:

$$D(\hat{W}_1) = \begin{cases} 0 & \hat{W}_1 = 0 \\ 1 & \hat{W}_1 = 1 \\ 0 & \hat{W}_1 = ? \text{ and } p_1 < \frac{1}{2} \\ 1 & \hat{W}_1 = ? \text{ and } p_1 \geq \frac{1}{2} \end{cases}$$

That is, if we are given $\hat{W}_1$, we output it without worrying about the bias (which anyway will usually be close to $\frac{1}{2}$). If we are not given $\hat{W}_1$, we go with the more likely option.

Since this maximizes the likelihood, the probability of the first error occurring at i is $H(W_i|W_{<i})$ from our proof of the Lemma (see exercise 5). probability of D is $\leq \sum_{i \in \overline{S}} H(W_i|W_{<i})$ from our proof of the lemma (see Exercise **??**) Taking the union bound over all $i \in \overline{S}$, i.e. the indices not sent directly, yields Claim **??**.