CS 229r Information Theory in Computer Science Jan 29, 2019

Lecture 8: A Gentle Introduction to Polar Codes

Instructor: Madhu Sudan *Scribe: Mirac Suzgun*

1 Bookkeeping

1.1 Outline for Today

1. Overview of Polar Codes.
2. Principal Claims.
3. Encoding + Decoding.

1.2 Administrative Issues

1. Professor Sudan will not be holding his office hours today.
2. Mitali has her usual office hours at 5.00 pm this evening.
3. Problem Set 2 is due Tuesday, February 26th.

2 Review: Linear Codes

Last time, we started setting up the stage for polar codes. We wanted to perform efficient correction of errors for the Binary Symmetric Channel with parameter p, BSC (p). We know its capacity rate, and would like to get ε-close to capacity using efficient coding algorithms.

We talked about the *divide-and-encode* technique: We basically took a large block, split it into smaller chunks, and then encoded each individual small chunk separately. Working with small chunks helps us manage running time, because it will perhaps be exponential in the size of the small blocks. However, we shall note that no matter what we do, the size of our small chunks will be $\mathcal{O}(1/\varepsilon^2)$, or some polynomial in $(1/\varepsilon)$, therefore our running time is exponential in parameter (i.e. $\mathcal{O}(2^{1/\varepsilon^2})$, which doesn't get us close to capacity with feasible algorithms.

We are interested in codes with efficient algorithms that take small chunks of information – it can be as small as you want, but presumably of length at least $1/\varepsilon^2$ – and compress these small chunks. From now on, our target theorem is the following:

Theorem 1. $\forall p \in [0,1]$, $\exists$ polynomials A, B such that $\forall \varepsilon > 0$, $\exists$ code of length $n \leq A(1/\varepsilon)$ that gets ε-close to capacity with pre-processing, encoding, and encoding time $\leq B(1/\varepsilon)$.

We want these codes to be short, and we would like to decode them efficiently.

When working with a linear compression scheme, the following theorem is equivalent to the previous theorem.

Theorem 2. Suppose $p \in (0, \frac{1}{2})$. $\exists$ polynomials A, B such that $\forall \varepsilon > 0$, $\exists n \leq A(1/\varepsilon)$ and $m \leq (h(p) + \varepsilon) \cdot n$ with a linear compressor $H \in \mathbb{F}_2^{m \times n}$ and an efficient decompressor D such that

$$\Pr_{Z \sim \text{Bern}(p)^n}[D(HZ) \neq Z] \leq \frac{1}{n^{10}} \tag{1}$$

Remark The term $\frac{1}{n^{10}}$ in Theorem 2 does not have a special meaning in the equation. Changing this term will only change the polynomials A and B.

Remark Assuming that we have such a such good (i.e. linear) compression algorithm, how can we construct a good coding algorithm? This was an open question until a decade ago. In 2008, we found a code that works. In 2013, we found a proof that this code works. And now, in 2019, we are actually able to teach in classroom.

Proposition 3. $\forall p \in (0, \frac{1}{2})$. $\exists \delta > 0$ such that $\forall n$, n bits can be compressed to length $m \leq h(p) \cdot n + \mathcal{O}(n^{1-\delta})$.

Exercise 4. Try to come up with a non-linear but efficient scheme that achieves $h(p) \cdot n + \mathcal{O}(n^{1/2})$.

Note that we still expect to see some loss, but it should not grow linearly in n.

2.1 Polar Codes [Arikan, 2008]

Let us now construct these magical codes. This idea is due to Erdal Arıkan, a Turkish information theorist. Arıkan said, let me take two bits and try to show you how to compress them efficiently. Two bits! What can we do with two bits? Remember that we can only perform a linear operation.

Suppose we have two bits U, V. One simple approach would be to take their (XOR) sum:

$$(U, V) \mapsto U + V \tag{2}$$

But this is too ambitious; we definitely lost one bit of information. So, this will not work, unfortunately.

Let us try to add one more information into this. What information can we add? We need to add something which is different that $U + V$. We need to do something which is linear. There are only two (reasonable) options remaining, so we will pick one them and output V, in addition to $U + V$, that is:

$$(U, V) \mapsto (U + V, V) \tag{3}$$

Remark It is important to realize that this is a completely reversible operation. If we are given the pair $(U + V, V)$, we can easily determine the values of U and V.

Arıkan noticed that this process does not compress yet, but it starts to differentiate the entropies.

Lemma 5. If U and V are i.i.d. random variables distributed according to Bern(p), where $p \in (0, \frac{1}{2})$, then

$$H(U + V) > H(U), H(V) \tag{4}$$

Suppose $U, V \overset{\text{i.i.d.}}{\sim} \text{Bern}(p)$, with $p \in (0, \frac{1}{2})$. Then, $U + V \sim \text{Bern}(p')$. In fact, if $0 < p < \frac{1}{2}$, then $0 < p < p' < \frac{1}{2}$. We can show that $p' = (2p)(1 - p)$. However, let us consider a simpler case, where $U, V \in \{-1, +1\}$.

$$U, V = \begin{cases} +1 & \text{with probability } 1 - p \\ -1 & \text{with probability } p \end{cases} \tag{5}$$

Now, consider the product of these two random variables, that is $U \cdot V$.

UV	**0**	**1**
0	$(1-p)^2$	$(1-p)p$
1	$p(1-p)$	p^2

Exercise 6. Analyzing $\mathbb{E}[U]$ and $\mathbb{E}[UV]$, show that $\frac{1}{2} > p' > p$.

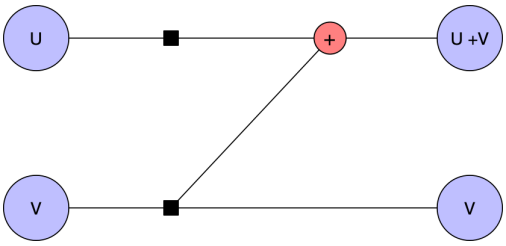

Figure 1: Local Polarization

Conditioning tells us $H(U,V) = H(U) + H(V)$. Therefore, we can write $H(U+V,V)$ as follows:

$$H(U+V,V) = H(U+V) + H(V \mid U+V) \tag{6}$$

If $U,V \overset{\text{i.i.d.}}{\sim} \text{Bern}(p)$, then $H(U) = H(V) = h(p)$. On the other hand, $U+V \sim \text{Bern}(p')$, where $p' > p$ and $H(V \mid U+V) = H(U+V,V) - H(U+V) = 2h(p) - h(p') < h(p)$.

2.2 The Main Idea

We would like to squeeze our conditional entries as much as possible to 0. We want every bit of the message to have an entropy rate close to 1 or 0. As we will see shortly, the bits with conditional entropy close to 0 will no longer be necessary, and at the end, we will send the bits whose conditional entropies are close to 1.

Let us start with n independent $\text{Bern}(p)$ bits. This is the message that we would like to compress. For the time being, let us assume that n is a power of 2, that is $n = 2^t$, for some $t \in \mathbb{N}$, however the algorithm works just fine even without this assumption.

Let us pair these n bits arbitrarily. We then get $(n/2)$ pairs of bits. For each of these ordered pairs of the form (U,V), we map (U,V) to $U+V$ and V, separately. We then group all the elements of the type $U+V$ and V together, while respecting their order. Now, all the elements in the first group are i.i.d. Bernoulli random variables with parameter p', where $0 < p < p' < \frac{1}{2}$. Therefore, $H(U+V) = h(p') > h(p)$. Similarly, all the elements in the second group satisfy the following condition: $H(V \mid U+V) < H(V) = h(p)$.

It should be clear to our astute readers that this process increases the conditional entropy of one group while decreasing that of the other. We therefore repeat this process until we have only single bits $W_1, W_2, \cdots, W_n$. Under this scheme, the conditional entropy of first singleton W_1 is very close to 1, whereas the conditional entropy of the last singleton W_n is very close to 0. We now would like to make sure that the singletons in the middle, whose entropies are in between 0 and 1, are as few as possible.

Claim 7. Suppose $\forall j \; H(W_0 \mid W_{<j}) \in \{0,1\}$, where $W_{<j}$ denotes the set of $W_1, W_2, \cdots, W_{j-1}$. Then, we claim that we solved the encoding problem.

Why does this scheme solve the encoding problem?

Let us note that:

$$H(W_1, W_2, \cdots, W_n) = H(Z_1, Z_2, \cdots, Z_n) \tag{7}$$
$$= n \cdot h(p) \tag{8}$$

In fact,

$$H(W_1, W_2, \cdots, W_n) = \sum_j H(W_j \mid W_{<j}) \tag{9}$$
$$= \#\{j \mid H(W_j \mid W_{<j}) = 1\} \tag{10}$$
$$= n \cdot h(p) \tag{11}$$

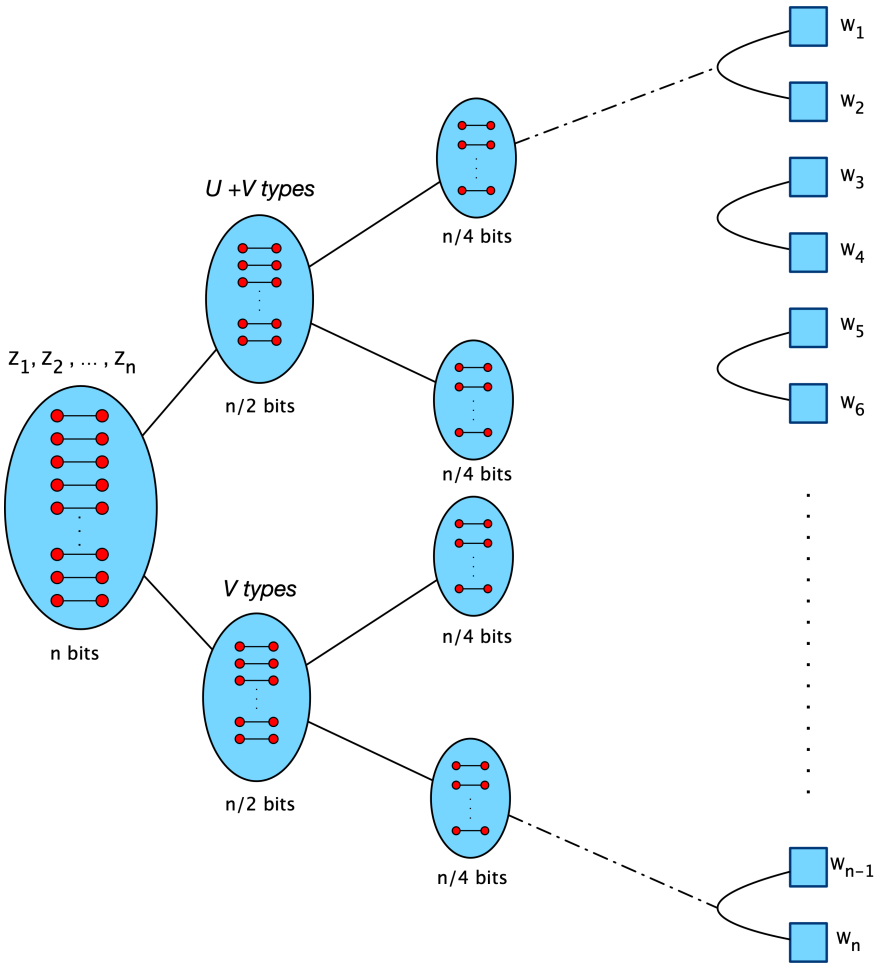

Figure 2: Polarization Process

Suppose $S = \{j \mid H(W_j \mid W_{<j}) = 1\}$, then $W_S \triangleq (W_{i_1, W_{i_2}, \cdots, W_{i_{|S|}}})$, with $S = \{i_1, i_2, \cdots, i_{|S|}\}$.
Therefore, we simply output W_S to compress the message Z.
However, there are still some issues we need to address:

1. $H(W_j \mid W_{<j}) \in \{0, 1\}$ is problematic.
2. How do we recover $Z_1, Z_2, \cdots, Z_n$ from W_S?
3. What is the set S?[1]

Let us focus on the first problem.

Theorem 8. Fix p. $\exists \gamma = \gamma(t)$, $\delta = \delta(t)$, and $\tau = \tau(t)$ such that

$$\Pr_{j \sim [n]}[\text{after } t \text{ steps } H(W_j \mid W_{<j}) \in (\tau, 1 - \gamma)] < \delta. \tag{12}$$

In other words, $\delta(t), \gamma(t), \tau(t) \to 0$, as $t \to \infty$.

[1]Today, we will ignore the pre-processing time, therefore we will not concern ourselves with the question of how to get S – we assume, for the time being, that we can determine the set S in exponential time.

Let us actually prove a stronger proposition:

$$\tau(t) < \underbrace{\left(\frac{1}{2^{11}}\right)^t}_{\text{fast exp. decay}} \quad \text{and } \delta(t), \gamma(t) \leq \underbrace{(0.999)^t}_{\text{gentle decay}} \tag{13}$$

Suppose we have this theorem. How would we compress it?
It turns out that we should send all the bits with condition entropy $H(W_j \mid W_{<j})$ between τ and 1.[2]
Now, let us define $S \triangleq \{j \mid H(W_j \mid W_{<j}) > \tau)$. Then,

$$|S| = |\{j \mid H(W_j \mid W_{<j}) \geq 1-\delta\}| + |\{j \mid H(W_j \mid W_{<j}) \in (\tau, 1-\delta)\}| \tag{14}$$

For the sake of simplicity, let $T = \{j \mid H(W_j \mid W_{<j}) \geq 1-\delta\}$, then

$$n \cdot h(p) \geq H(W_T) \geq |T|(1-\delta) \tag{15}$$

Therefore,

$$|S| = |\{j \mid H(W_j \mid W_{<j}) \geq 1-\delta\}| + |\{j \mid H(W_j \mid W_{<j}) \in (\tau, 1-\delta)\}| \tag{16}$$

$$\leq \frac{n \cdot h(p)}{1-\gamma} + n \cdot \delta \tag{17}$$

$$\geq n \cdot h(p) + n \cdot (\gamma + \delta) \tag{18}$$

The length of the compression is determined by γ and δ, so we want $n\gamma, n\delta < n^{1-\delta} \iff \gamma, \delta < n^{-\delta}$.
Recall that we started with Z and mapped everything to W, followed by W_S. What should we do with the rest of the bits in $W_{\overline{S}}$?

$$Z \to W \to W_S \qquad \text{(Compression)} \tag{19}$$

$$\hat{Z} \leftarrow \hat{W} \leftarrow W_S \qquad \text{(Decompression)} \tag{20}$$

Exercise 9. Show the explicit matrix that maps $Z \to W$.

If $H(W_{\overline{S}} \mid W_S) = \beta$ is very small, we can guess $W_{\overline{S}}$, given W_S.
Remark Recall that we have shown in the first problem set that the conditional entropy of Y is small given X if and only if Y is predictable given X.

We can say $H(W_{\overline{S}} \mid W_S) \leq |\overline{S}| \cdot \tau \leq n\tau$.
Want our β to be:

$$\beta \leq \frac{1}{n^{10}} \Rightarrow n^{\tau} \leq \frac{1}{n^{10}} \Rightarrow \tau \leq \frac{1}{n^{11}} = \frac{1}{(2^t)^{11}} = \left(\frac{1}{2^{11}}\right)^t \tag{21}$$

Theorem 10. Fix $p \in (0, \frac{1}{2})$. $\forall C \in \mathbb{N}$, $\exists \beta < 1$ such that $\exists \gamma = \gamma(t)$, $\delta = \delta(t)$, and $\tau = \tau(t)$ with $\gamma, \tau \leq C^{-t}$ and $\delta < \beta^t$ satisfying

$$\Pr_{j \sim [n]}[\text{after } t \text{ steps } H(W_j \mid W_{<j}) \in (\tau, 1-\gamma)] < \delta. \tag{22}$$

This leads to a very efficient algorithm, with encoding time at most $\mathcal{O}(n \log n)$, given that we know the set S. Next time, we will prove this new theorem and discuss decoding.

References

[Arikan, 2008] Arikan, E. (2008). Channel polarization: A method for constructing capacity-achieving codes. In *2008 IEEE International Symposium on Information Theory*, pages 1173–1177. IEEE.

[2]Send $(\gamma + \delta)$ fraction. Since δ is going to zero, we can surely afford it.