CS 229r Information Theory in Computer Science Feb 21, 2019

Lecture 8

Instructor: Madhu Sudan *Scribe: Dan Stefan Eniceicu*

1 Overview

1.1 Outline

1. Polar Coding
2. Overview
3. Principal Claims
4. Encoding, etc.

1.2 Administrative Things

1. No office hours today
2. Mitali has usual office hours
3. Pset 2 due Tuesday
4. When you ask a question, please state your name so that Madhu gets to know you

2 Review

Our goal is to perform efficient correction of errors for the binary symmetric channel BSC(p). We know its capacity; we want to get ε close to capacity with efficient algorithms, and one thing we have seen is that we can take a large block, split it into smaller chunks and work with each one separately. Working with small blocks helps with running time because it will be maybe exponential in the size of the small blocks. But no matter what, this length will be $\sim O(1/\varepsilon^2)$ or some polynomial in $1/\varepsilon$, which when exponentiated results in $\sim O(2^{1/\varepsilon^2})$. It turns out we are only interested in blocks of small length (around $1/\varepsilon^2$). The target theorem from now on is:

Theorem 1. *$\forall p \in [0,1]$, there exist polynomials A and B such that $\forall \varepsilon > 0$ there exists a code of length $n \leq A(1/\varepsilon)$ that gets ε close to capacity, with preprocessing, encoding, decoding time all $\leq B(1/\varepsilon)$.*

Reminder: These codes need to be short and we want to decode them very efficiently. This theorem is our target. Last time we said that even though we were talking about error correction, we should actually be talking about compression with a linear mechanism, so we are interested in a linear compression scheme. The following is a theorem which is equivalent to the other one when working with a linear compression scheme:

Theorem 2. *$\forall p \in (0, 1/2)$, there exist polynomials A, B such that $\forall \varepsilon > 0$, there exist $n \leq A(1/\varepsilon)$, $m \leq (h(p) + \varepsilon)n$, a matrix $H \in \boldsymbol{F}_2^{m \times n}$, and a decompressor D such that*

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

Notes:

1. The term $1/n^{10}$ is just a constant. Changing this constant will likely change A and B.
2. Linearity over $\mathbf{F}_2$ is xor sum between bit subsets.
3. Theorem 2 implies Theorem 1, so we will prove Theorem 2.
4. In 2008 (spoiler: https://arxiv.org/abs/0807.3917) a code satisfying this was found and in 2013 it was proved that said code works. Now we can teach it. :)

We can attempt to work with an alternative claim:

Claim 3. *$\forall p \in (0, 1/2)$ there exists $\delta > 0$ such that $m \leq h(p) + n + O(n^{1-\delta})$.*

We basically want subpolynomial behavior in n.

Exercise 4. *Try to come up with a nonlinear (efficient) scheme that achieves for example $h(p)n + O(n^{0.51})$.*

This shows us we should expect to see some loss (which can be sublinear in n). This is our target.

3 Polar Codes

3.1 Compressing 2 bits

Arikan's remarkable idea was to look at 2 bits and attempt to compress them linearly. So we have two bits, u, v which we can try to compress. What can we do with them? One idea is to take their xor sum, $u+v$. This leads to information loss. It is too ambitious, so we need one more bit. Let's then map $(u, v) \to (u + v, v)$. This is invertible, so obviously not a good compression scheme. What Arikan noticed however was that

$$H(u, v) = H(u + v, v)$$

due to invertibility, but at the same time, we have

$$H(u + v) > H(u), H(v).$$

To prove this last claim, let $u, v \sim \text{Bern}(p)$ i.i.d. Then, $u + v \sim \text{Bern}(p')$. For $0 < p < 1/2$, the claim is equivalent to $p < p' < 1/2$. We can show explicitly that $p' = 2p(1-p)$, so $p < p' < 1/2$. A more slick proof is to consider $u, v \in \{-1, 1\}$, with 1 corresponding to probability p and -1 corresponding to probability $1 - p$. We now look at their product, uv.

Exercise 5. *By looking at $E[u]$ and $E[uv]$, complete the slick proof to show that $1/2 > p' > p$.*

Now, we have

$$H(u, v) = H(u + v, v) \text{ and } H(u + v) > H(u), H(v).$$

By the chain rule,

$$H(u, v) = H(u) + H(v),$$

since u and v are independent, and

$$H(u + v, v) = H(u + v) + H(v|u + v).$$

Thus,

$$H(v|u + v) < H(v).$$

3.2 The method

The idea behind the algorithm is to attempt to squeeze conditional entropies of as many bits as possible to 0. This is a good goal to attempt simply because we are able to infer with high probability the bits whose conditional entropy is very small, based on the values of the other bits that the conditioning is taken over. The bits with conditional entropy close to 0 are therefore not necessary anymore. Thus, the goal is to get every bit to have conditional entropy very close to either 0 or 1 (the ones close to 1 are the ones we send, and the ones close to 0 are the ones we suppress). The algorithm is remarkably simple and effective: We simply repeat the process with 2 bits!

More precisely, let $Z_1, ..., Z_n \sim \text{Bern}(p)$ i.i.d. These represent the bits we want to compress. We assume for simplicity that n is a power of 2. The algorithm works just as well if n is not a power of 2, but for our purposes, we will let $n = 2^t$ for some positive integer t.

We pair the n bits arbitrarily, to obtain $n/2$ (ordered) pairs of bits. For each of these pairs (u, v), we let $(u, v) \to (u + v, v)$, and group the $n/2$ elements of the form $u + v$ together (we call this the top group) and group the $n/2$ elements of the form v together (we call this the bottom group). Now, all the elements of the top group are i.i.d. Bern(p') random variables with $p < p' < 1/2$, so $H(u + v) = h(p') > h(p)$, and the elements of the bottom group will satisfy

$$H(v|u + v) < H(v) = h(p).$$

Since the elements of the top group, apart from $u + v$, are independent of v, we can instead condition on all bits in the top group, leading to $H(v|\text{top group bits}) < h(p)$. We can think of this step as leading to an increase in entropy for the top group bits and a decrease in conditional entropy for the bottom group bits.

We repeat this step with each group obtained at a previous step, until we are left with singleton bits, i.e. there are n groups left, each with one bit, $\{w_1\}$,...,$\{w_n\}$.

Note: Vertical placement does not determine ordering among conditional entropies.

This method allows w_1 to have entropy very close to 1, and w_n to have conditional entropy very close to 0, satisfying our goal. We need to make sure that a small number of bits will end up in the middle, i.e. with conditional entropy not close to 0 or 1. Before analyzing this possibility, let's see the result if it did not happen. Let $w_{<j} = \{w_1, \ldots, w_{j-1}\}$.

Claim 6. *Suppose $H(w_j|w_{<j}) \in \{0, 1\}$ for all j. Then we have solved the problem, at least of encoding.*

Why does this give us a compression scheme? We know

$$\sum_{j=1}^{n} H(w_j|w_{<j}) = H(w_1, \ldots, w_n) = H(z_1, \ldots, z_n) = nh(p).$$

Let $S = \{j \in \{1, \ldots, n\} : H(w_j|w_{<j}) = 1\} = nh(p)$. Then, the claim implies that $|S| = nh(p)$, which is what we want up to sublinear terms in n.

Let $W_S = \{w_{i_1}, \ldots, w_{i_{|S|}}\}$, where $S = \{i_1, \ldots, i_{|S|}\}$. To compress the vector Z, we simply output W_S.

Note: The supposition that $H(w_j|w_{<j}) \in \{0, 1\}$ for all j is not true. However, its analysis is illustrative of the idea behind the compression scheme.

3.3 Analysis

The issues we still have to worry about:

1. $H(w_j|w_{<j}) \in \{0,1\}$ is problematic.
2. How to recover $z_1, \ldots, z_n$ from W_S.
3. What is S?

S depends on p and n (not on z_1,...,z_n). We address point 3, by choosing not to answer this question. This is a question of preprocessing time, which we will skip in the next few lectures (assume infinite preprocessing time). There is indeed a polynomial time deterministic algorithm to find S.

How do we resolve the other issues? We first need to define the limits we are interested in precisely. The main claim we will prove is

Theorem 7. *Fix $p \in (0,1)$. Then, there exist $\gamma = \gamma(t)$, $\delta = \delta(t)$, $\tau = \tau(t)$ such that*

$$Pr_{j,j\sim[n]}\big[\textit{after } t \textit{ steps, } H(w_j|w_{<j}) \in (\tau, 1-\gamma)\big] < \delta.$$

Then, at the very least $\gamma(t), \delta(t), \tau(t) \to 0$ as $t \to \infty$.

We would actually prefer a stronger claim, such as

Claim 8. *$\tau(t) < (1/2^{11})^t$ and $\delta(t), \gamma(t) < 0.999^t$), so each of these gets exponentially suppressed.*

This claim is what we need to prove Theorem 2. The algorithm is not optimal, but this analysis is enough for our current goal. Note that at each step, if we look at the average value of $H(w_j|w_{<j})$ is constant at different columns. Also it is impossible for $\delta < 2^{-t}$.

Which subset of $\{w_1, \ldots, w_n\}$ do we want to send? It turns out we want to send all bits with conditional entropy $H(w_j|w_{<j})$ between τ and 1. We let

$$S = \{j : H(w_j|w_{<j}) \geq \tau\}$$

and

$$T = \{j : H(w_j|w_{<j}) \geq 1-\gamma\}.$$

Then,

$$|S| = |T| + |\{j : H(w_j|w_{<j} \in (\tau, 1-\gamma)\}| = |T| + \delta n.$$

Note that $(1-\gamma)|T| \leq H(W_T) \leq nh(p)$. Thus,

$$|S| \leq \frac{h(p)n}{1-\gamma} + \delta n \approx nh(p) + (\gamma + \delta)n.$$

How good the length of the compression is is determined by γ, δ, which we want to be polynomial in n, $\gamma(t), \delta(t) < n^{-\alpha}$, so $\gamma, \delta < (2^{-\alpha})^t \approx (1-\alpha)^t$.

We know what to do with the bits with conditional entropy between $(1-\gamma, 1)$, $(\tau, 1-\gamma)$. What do we do with the rest, ie the bits in $W_{\overline{S}}$? We started out with Z, and our compression scheme was to map

$$Z \to W \to W_S.$$

Exercise 9. *The above operation is linear. What is the matrix (acting on the dimension n vector space over $\boldsymbol{F}_2$) which maps Z to W?*

For decompression, we now start with W_S,

$$W_S \to \hat{W} \to \hat{Z}.$$

We proved in homework 1, problem 5a, that if $H(W_{\overline{S}}|W_S) \leq$ something small, then we can guess $W_{\overline{S}}$. A precise bound is

$$H(W_{\overline{S}}|W_S) \leq |\overline{S}| \cdot \tau \leq n\tau.$$

We proved that $H(W_{\overline{S}}|W_S) \leq \beta$, then we can guess correctly with probability $1 - \beta$. We want to compress correctly to $1/n^{10}$, so we want $\beta \leq 1/n^{10}$, and $n\tau = \beta$, so $\tau \leq 1/n^{11} = (1/2^{11t})$. We will actually prove that $\gamma, \tau \leq c^{-t}$, $\delta < \beta^t$. More precisely, we will prove

Theorem 10. *Fix $p \in (0, 1/2)$. Then, $\forall 1 < c < \infty$, there exists $\beta < 1$, such that $\gamma(t), \tau(t) \leq c^{-t}$, $\delta(t) \leq \beta^t$.*

This leads to a very efficient encoding algorithm, with encoding time at most $O(n \log n)$, provided we know S, which follows from preprocessing.

In the next lecture we will prove this theorem and discuss decoding.