# *Lecture 10: Polar codes*

*Anup Rao*

*October 28, 2019*

In this lecture, we return to the task of constructing (non-local) codes over the binary alphabet. Recall that we have seen two kinds of explicit codes — Reed-Solomon codes, which are optimal, but use a larger alphabet, and expander codes, which use the binary alphabet but do not have the optimal tradeoff between relative distance and rate. Today, we discuss *polar codes*. These are binary codes developed by Arikan that have the optimal tradeoff between rate and relative distance, but the catch is that they only work when the errors are promised to be random.

A good reference for this stuff is this text: `https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/`.

It will be convenient to use the tensor product of matrices to describe the code. Given two matrices $A, B$, their tensor product is

$$A \otimes B = \begin{bmatrix} A_{1,1} \cdot B & A_{1,2} \cdot B & \dots & A_{1,n} \cdot B \\ A_{2,1} \cdot B & A_{2,2} \cdot B & \dots & A_{2,n} \cdot B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} \cdot B & A_{m,2} \cdot B & \dots & A_{m,n} \cdot B \end{bmatrix}.$$

If $x = (x_1, x_2, \dots, x_n)$ is a column vector, and $X$ is the matrix whose columns are $x_1, \dots, x_n$, then $(A \otimes B)x$ is the vector obtained by computing $BXA^\intercal$ and rewriting this as a vector.

Define the polarizing matrix $P \in \mathbb{F}_2^{n \times n}$ as

$$P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

$t$ times. The matrix $P$ is an invertible $2^t \times 2^t$ matrix. In fact $P^2 = I$. We set $n = 2^t$. The tensor structure of $P$ allows us to compute $Px$ very quickly. Indeed, you can compute

$$\left( \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes P \right) x = (PX) \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix},$$

which gives a recursive algorithm for computing $Px$ that runs in time $O(n \log n)$.

The key property of these matrices that makes them useful for protecting against random errors is that when $Z \in \mathbb{F}_2^n$ is sampled by picking each bit to be 1 independently with probability $\epsilon$, then almost all of the bits of $PZ$ become *polarized* — except for a negligible fraction of the coordinates of $PZ$, each coordinate is close to uniform conditioned on previous coordinates, or determined by the previous coordinates.

## *An unreasonable(?) assumption*

To understand why polarization is such a useful property, let us imagine for a second that we have perfect polarization. Say that $Z \in \mathbb{F}_2^n$ is an $\epsilon$-noisy string if each bit of $Z$ is sampled independently and equal to 1 with probability $\epsilon$.

Suppose $Y = PZ$, and for every coordinate $i$, we have

$$H(Y_i|Y_{<i}) \in \{0, 1\},$$

namely, the conditional entropy of each bit is either 0 or 1. Then, since $P$ is invertible, we have

$$n \cdot h(\epsilon) = H(Z) = H(Y) = \sum_{i=1}^{n} H(Y_i|Y_{<i}) = \sum_{i \in S} H(Y_i|Y_{<i}),$$

where here $S$ is the set of coordinates where the conditional entropy is 1. We must have $|S| = n \cdot h(\epsilon)$. Intuitively, $Z$ must be determined by the coordinates of $Y$ that corresponds to $S$. Indeed, given $Y$ restricted to the coordinates in $S$, we can reconstruct all the other coordinates of $Y$, since every other coordinate is determined by the coordinates of $Y$ in $S$. Then we can recover $Z = P^{-1}Y$. So, $Y$ restricted to $S$ is an encoding of $Z$.

Now, let $C \in \mathbb{F}_2^n$ be the subspace of dimension at least $n - |S|$ such that $(Px)_i = 0$ for all $i \in S$. We shall prove that $C$ is a code. Moreover, since $P^2 = I$, this code has a simple encoding function. Just take a message $u \in \mathbb{F}_2^{h(\epsilon)\cdot n}$, and let $v \in F_2^n$ be the vector obtained by putting the bits of $u$ in the locations that corresponds to the high entropy outputs of $P$. Then set the codeword to be $Pv$. We have $P(Pv) = v$, so $Pv \in C$. Moreover, this is an injective map.

To see that $C$ is a code, observe that given $x + Z$, we can recover $x$ as follows. Compute $P(x + Z) = Px + PZ$. Since $x \in C$, for every $i \in S$, we must have

$$(P(x + Z))_i = (Px)_i + (PZ)_i = (PZ)_i = Y_i.$$

So, we can recover $Z$ from received word. Once we have recovered $Z$, we can recover $x$ from the received word. Thus, we obtain a code with rate $1 - h(\epsilon)$, which is optimal.

## *Eliminating the assumption*

In fact, something very close to the assumption we made in the last section is true — almost all of the coordinates in the output have entropy that is extremely close to either 0 or 1.

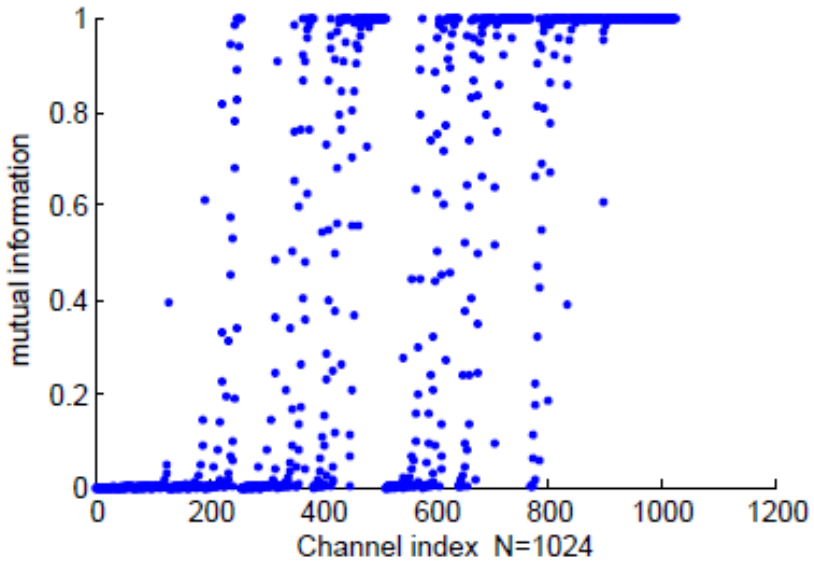


Figure 1: An example of the polarization phenomenon. The mutual information between $X$ and the coordinates of the output is shown. Most coordinates have almost no information, or close to one bit of information. This picture was taken from this paper: `http://www.ijicic.org/ijicic-13-12027.pdf`.

Given a particular $\epsilon$ and a parameter $\kappa$, let us call the set of coodinates $i$ with $\kappa \leq h_i \leq 1-\kappa$ the set of $\kappa$-middle coordinates. These are the set of coordinates where the entropy is $\kappa$-far from both 0 and 1. The key technical theorem is

**Theorem 1.** *For every $\epsilon, \delta > 0$, there is a $t_0 > 0$, such that as long as $t > t_0$, then the set of $n^{-4}$-middle coordinates is of size at most $\delta n$.*

Proving the theorem is delicate, at least at present. I was not able to find a *book-proof* for it. So, we will not cover the full proof here.

Erdős believed that God has a book containing the most elegant proofs. Whenever he saw a proof that he believed came from the book, he would call it a proof from The Book.

Let us see how to use Theorem 1 to get a code. Later we shall return to giving some intuition for why the theorem is true. Let $S$ denote the set of coordinates that are in the middle. Let $T$ denote the set of coordinates where the entropy exceeds $1-n^{-4}$, and $V$ denote the set of coordinates where the entropy is at most $n^{-4}$. We shall the define the code to be the set of vectors $x \in \mathbb{F}_2^n$ such that in $Px$, values of $x$ in $T \cup S$ is 0.

Each coordinate of $T$ contributes at least $h^{-1}(1-1/n^4) \geq 1-1/n$ to the entropy. So, we get $(1-1/n)|T| \leq h(\epsilon)n$, proving that $|T| \leq h(\epsilon)n(1+2/n) \leq h(\epsilon)n+2$. So we get,

$$|V| = n - |T| + |S| \geq n - h(\epsilon)n - 2 - \delta n \geq n - h(\epsilon)n - 2\delta n.$$

In other words, the rate of the code is $(1-h(\epsilon)-2\delta)$. Moreover, the code can recover from errors, exactly as above. This is because during the decoding algorithm, we will need to guess the values of coordinates whose entropy is at most $1/n^4$. In each step, the decoder just

guesses the more likely value. The probability of having a decoding failure is at most

$$\begin{aligned}&\mathop{\mathbb{E}}_{Y}\left[\sum_{i\in V} h^{-1}(H(Y_i|Y_{<i}))\right]\\&=\sum_{i\in V}\mathop{\mathbb{E}}_{Y}\left[h^{-1}(H(Y_i|Y_{<i}))\right].\end{aligned}$$

Now, two bad things can happen. The first is that $H(Y_i|Y_{<i}) > 1/n^2$. This happens with probability at most $1/n^2$ for each $i$, since the expected value of this quantity is at most $1/n^4$. The second bad thing is that $Y_i$ does not take on the most likely value. But if the conditional entropy is at most $1/n^2$, this happens with probability at most $1/n^2$, since $h(\alpha) \geq \alpha$. In total, the probability of decoding failure is at most $n(1/n^2 + 1/n^2) = 2/n^2$.

## *Some intuition for Theorem 1*

Let us return to trying to understand why the polarization should occur. Throughout this discussion, let $\epsilon$, the noise rate, be an arbitrary constant in between 0 and 1. Given a particular choice of $t$, let $h_i$ be defined to be $H(Y_i|Y_{<i})$. We always have $h_1 + h_2 + \ldots + h_n = h(\epsilon) \cdot n$.

Here is an observation:

**Lemma 2.** $\lim_{t\to\infty} \sum_{i=1}^{n} h_i(1-h_i)/n = 0$.

In words, the lemma says that the average coordinate has conditional entropy close to 0 or 1.

Let us sketch the proof. Consider a particular value of $t$, and let us see what happens when $t$ is increased by 1. Recall that $Z$ is an $\epsilon$-noisy string, and we can compute the output in step $t+1$ as

$$P \cdot \begin{bmatrix} Z & Z' \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} Y & Y' \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},$$

where $Y$ and $Y'$ are independent identically distributed strings that are the outputs after $t$ polarization steps.

Now, what happens to the average $\sum_{i=1}^{n} h_i(1-h_i)/n$? Each term $h_i$ that corresponds to $H(Y_i|Y_{<i})$ is replaced by two terms. One of them corresponds to $f_i = H(Y_i + Y'_i|Y_{<i}, Y'_{<i})$, and the second corresponds to $g_i = H(Y'_i|Y_{<i}, Y'_{<i}, Y_i + Y'_i)$. Since the map is invertible, we have $f_i + g_i = 2h_i$. On the other hand, it turns out that $f_i$ is very close to $2h_i$. So we see that

$$\begin{aligned} h_i(1-h_i) - \frac{f_i(1-f_i)+g_i(1-g_i)}{2} &= \frac{f_i^2+g_i^2}{2} - h_i^2 \\ &= \frac{(f_i+g_i)^2+(f_i-g_i)^2}{4} - h_i^2 \\ &= \frac{(2h_i)^2+(f_i-g_i)^2}{4} - h_i^2 \\ &= \frac{(f_i-g_i)^2}{4}. \end{aligned}$$

You can argue that $f_i > g_i$, so the average is always strictly decreasing. In fact, it is decreasing very quickly.