**Introduction to Coding Theory** **CMU: Fall 2018**

# Lecture 7: Polar codes

October 2018

*Lecturer: Venkatesan Guruswami* *Scribe: Andrii Riazanov, Sai Sandeep*

## 1 Achieving Shannon capacity

In this lecture we return to the challenge of achieving Shannon capacity with explicit codes. During the last lecture we already considered the concatenated codes, introduced by Forney, which achieve the rate $1-h(p)-\varepsilon$ for $\mathrm{BSC}_p$ channel for any given $\varepsilon > 0$. However, the inner code in this approach was found by brute-force search for the code of small length which achieves Shannon capacity, which means that it doesn't have any good structure. This issue caused the decoding complexity to be exponential in $\varepsilon$, which is not nice. This leads us to the following challenge:

**Question 1.1.** *Can we achieve the capacity for $BSC_p$ (i.e. the rate $1-h(p)-\varepsilon$) using (explicit) codes with block length and encoding/decoding complexity that are* $\mathrm{poly}(1/\varepsilon)$*?*

Note that by Shannon's Theorem, the block length $n$ is of order $\Omega(\frac{1}{\varepsilon^2})$ for the codes of rate $1-h(p)-\varepsilon$ for $\mathrm{BSC}_p$. This means that we cannot use brute-force search anymore to address Question 1.1, since it takes $\exp(n)$ time.

In this lecture we consider *Polar codes*, introduced in the celebrated paper by Arikan [Ari09], where these codes were shown to achieve the capacity with $O(n\log n)$ encoding and decoding complexity for $n\to\infty$. Further, Guruswami and Xia proved in [GX15] that polar codes also have polynomially fast convergence to capacity, meaning that it suffices to take the block length $n \le \mathrm{poly}\left(\frac{1}{\varepsilon}\right)$ to achieve rate $R \ge 1-h(p)-\varepsilon$ for $\mathrm{BSC}_p$. Together with $O(n\log n)$ encoding and decoding complexity, this means that the polar codes answer the crucial Question 1.1 affirmatively. There are no other codes known to provably converge to the Shannon capacity polynomially fast, though spatially coupled LDPC codes are another candidates.

Here we only consider the context of $\mathrm{BSC}_p$, though the theory of polar codes applies more generally to Discrete Memoryless Channels.

Let's now formalize the challenge formulated in the Question 1.1 by the following theorem:

**Theorem 1.2.** *For any $p \in \left[0, \frac{1}{2}\right]$, there exists a polynomial $n_0(\cdot)$, such that for any $\varepsilon > 0$ there exist $k, n$, an encoder $Enc : \mathbb{F}_2^k \to \mathbb{F}_2^n$, and a decoder $Dec : \mathbb{F}_2^n \to \mathbb{F}_2^k$ such that*

1. *Codes are short and the rate is close to the capacity of $BSC_p$:*

$$n \le n_0\left(\frac{1}{\varepsilon}\right), \qquad k \ge (1-h(p)-\varepsilon)n;$$

2. *$Enc(\cdot)$ and $Dec(\cdot)$ run in $O(n\log n)$ time;*

*3. For all $m \in \mathbb{F}_2^k$:*

$$\mathop{\mathbf{Pr}}_{z \sim \mathrm{Bern}(p)^{\otimes n}} [Dec(Enc(m) + z) \neq m] \leq \frac{1}{n}.$$

In the above theorem we only require the decoding error probability to be small rather than exponentially small in $n$, as we had in Shannon's Theorem. It turns out that one can use concatenation codes (with the code in the above theorem as an inner code, and the Reed-Solomon code correcting $\Omega(n)$ fraction of errors as an outer code) to get an exponentially small error decoding probability , i.e. $\leq 2^{-\Omega_\varepsilon(n)}$ (an exercise).
(Note: by clever analysis, it can be shown that one can take the polynomial $n_0(x) = \theta(x^4)$ in Theorem 1.2).

We now proceed to the description of the polar codes, which will satisfy the desired conditions in Theorem 1.2.

## 2 Reduction to linear compressions

In this section we will change the context of our problem from error-correction codes to compressions of a vector of independent Bernoulli random variables.

Let $Z = (Z_1, Z_2, \ldots, Z_n) \sim Bern(p)^{\otimes n}$ be a vector of independent Bernoulli random variables. We know that the information content of $Z$ is equal to $h(p) \cdot n$ bits, where $h(\cdot)$ is a binary entropy function. Then one might want to *compress* the vector $Z$, i.e. to represent the information it carries in $\approx h(p)n$ bits. More formally, we want to find a compressor $Comp : \{0,1\}^n \to \{0,1\}^m$ such that $m \leq (h(p)+\varepsilon)n$, and for which we then can decompress the initial vector $Z$ from $Comp(Z)$ with high probability.

The Hamming weight of $Z$ is $\leq (p+\varepsilon)n$ with high probability, meaning that $Z$ lies in the Hamming ball $B$ centered at 0 of radius $(p+\varepsilon)n$ with high probability. The size of the ball $B$ is $\mathrm{Vol}(n, (p+\varepsilon)n) \approx 2^{(h(p)+\varepsilon)n}$. Then one way to build a compressor is just to enumerate all strings inside the ball $B$ using $\{0,1\}^{(h(p)+\varepsilon)n}$ strings, and when $Z$ lies inside the ball, the compressor maps $Z$ directly to the enumerated string in $\{0,1\}^{(h(p)+\varepsilon)n}$. Otherwise, when $Z$ is outside the ball $B$ (the probability of this event is negligible), the compressor just gives up (maps to an arbitrary string). The decompressor just maps the string from $\{0,1\}^{(h(p)+\varepsilon)n}$ back to the vector from the ball $B$ which this strings enumerates. In these settings, we obtain

$$\mathop{\mathbf{Pr}}_{Z} [Decom(Comp(Z)) \neq Z] = \mathop{\mathbf{Pr}}_{Z} [Z \notin B] \to 0.$$

This is called *lossless compression*, meaning that most of the time we will get $Z$ back after compressing and decompressing it. So the above example shows that the compression in general is an easy task.

What we in fact need for the construction of the polar codes is a *linear compression*. That is, a linear map $H : \mathbb{F}_2^n \to \mathbb{F}_2^k$ which has the same properties of compression, discussed above. Now, instead of directly looking for the codes which satisfy the conditions of Theorem 1.2, we will provide a linear compression scheme with similar conditions:

1. $n \leq n_0\left(\frac{1}{\varepsilon}\right)$;

2. Compression and decompression algorithms run in $O(n \log n)$;

3. $\mathbf{Pr}_{Z\sim \mathrm{Bern}(p)^{\otimes n}}\left[Decomp(HZ) \neq Z\right] \leq \frac{1}{n}$.

The following Lemma shows that such a construction is indeed enough to prove Theorem 1.2

**Lemma 2.1.** *If $H$ is a linear compression, then $C = \{c \,|\, Hc = 0\}$ (the code for which $H$ is a parity check matrix) is a capacity achieving code for $BSC_p$, with "good" encoding/decoding complexity, and decoding error probability equal to the decompression error probability of $H$.*

*Proof.* Since $H \in \mathbb{F}_2^{m\times n}$ is a parity check matrix for $C$, the rate satisfies

$$R(C) \geq 1 - \frac{m}{n} \geq 1 - h(p) - \varepsilon.$$

Next, the generator matrix $G$ can easily be found from the equation $H \cdot G = 0$ and condition $\operatorname{rank}(G) = n - \operatorname{rank}(H)$ using linear algebra tools. So the encoding complexity for $C$ is just a complexity of multiplying $G$ by a vector.

Let's analyze the decoding for $C$. Denote $R = Enc(m) + Z$ – the received message from $\mathrm{BSC}_p$ channel. Then

$$H \cdot R = H \cdot (Enc(m) + Z) = \underbrace{H \cdot Enc(m)}_{=0} + H \cdot Z = H \cdot Z.$$

When we run the decompression on $HR = HZ$, we receive $Z$ with probability $1 - \tau$, where $\tau$ is the decompression error probability for the compression scheme $H$. Therefore, $Enc(m) = R - Decomp(HR)$ with probability $1 - \tau$, meaning that the decoding error probability for $C$ is $\tau$. Note that we also described the decoding algorithm above, since $m = Enc^{-1}(R - Decomp(HR)) = G^* \cdot (R - Decomp(HR))$ in the case of correct decoding, where $G^*$ is a left inverse for $G$. Thus the decoding complexity is equal to the complexity of compression (multiplying by $H$), decompression, and multiplying by $G^*$. □

Considering the above connection between linear codes and linear compressions, we now can formulate the analogue of Theorem 1.2 for linear compressions:

**Theorem 2.2.** *For any $p \in \left[0, \frac{1}{2}\right]$, there exists a polynomial $n_0(\cdot)$, such that for any $\varepsilon > 0$ there exist $k, n$, and a linear map $H : \mathbb{F}_2^n \to \mathbb{F}_2^m$ such that*

1. $n \leq n_0\left(\frac{1}{\varepsilon}\right)$, $\qquad m \leq (h(p) + \varepsilon)n$;

2. *Compression and decompression run in $O(n \log n)$ time;*

3. *For all $m \in \mathbb{F}_2^k$:*

$$\mathop{\mathbf{Pr}}_{z\sim \mathrm{Bern}(p)^{\otimes n}}\left[Decomp(HZ) \neq Z\right] \leq \frac{1}{n}.$$

From now on, we will move on to the construction of the linear compression scheme satisfying Theorem 2.2, from which the construction of the code for Theorem 1.2 will follow by Lemma 2.1.

## 3 Information Theory background

We need to cover some basic information theoretic notions and their properties for our further constructions.

The *entropy* of the random variable $X$, denoted by $H(X)$, is defined as

$$H(X) = \sum_{x \in \mathrm{supp}(X)} \mathbf{Pr}[X = x] \log_2 \frac{1}{\mathbf{Pr}[X = x]}.$$

For any r.v. $X$ it holds $0 \leq H(X) \leq \log|\mathrm{supp}(X)|$, and the equality $H(X) = \log|\mathrm{supp}(X)|$ is satisfied only for $X$ uniformly distributed over $\mathrm{supp}(X)$.
Note that $H(Bern(p)) = p \log \frac{1}{p} + (1-p) \log \frac{1}{1-p} = h(p)$, where $h(p)$ is the binary entropy function.

For several random variables $X_1, X_2, \ldots, X_n$, the *joint entropy* $H(X_1, X_2, \ldots, X_n)$ is defined as just the entropy of $(X_1, X_2, \ldots, X_n)$ treated as a single random variable, i.e.

$$\begin{aligned} H(X_1,&X_2, \ldots, X_n) \\ &= \sum_{x_1 \in \mathrm{supp}(X_1)} \cdots \sum_{x_n \in \mathrm{supp}(X_n)} \mathbf{Pr}[X_1 = x_1, \ldots, X_n = x_n] \log_2 \frac{1}{\mathbf{Pr}[X_1 = x_1, \ldots, X_n = x_n]}. \end{aligned}$$

For independent $X_1, X_2, \ldots, X_n$ the joint entropy is additive: $H(X_1, X_2, \ldots, X_n) = \sum_{i=1}^n H(X_i)$. In general, the joint entropy is subadditive: $H(X_1, X_2, \ldots, X_n) \leq \sum_{i=1}^n H(X_i)$ for any random variables $X_1, X_2, \ldots, X_n$.

Next, the *conditional entropy* $H(X|Y)$ for random variables $X$ and $Y$ is defined as

$$H(X|Y) = \sum_{y \in \mathrm{supp}(Y)} \mathbf{Pr}[Y = y] H(X|Y = y) = \mathop{\mathbf{E}}_{y \sim Y}[H(X|Y = y)].$$

The *chain rule* for the conditional entropy is:

$$H(X, Y) = H(Y) + H(X|Y) = H(X) + H(Y|X),$$

which can be easily obtained using $\mathbf{Pr}[X = x, Y = y] = \mathbf{Pr}[Y = y]\,\mathbf{Pr}[X = x|Y = y]$. The chain rule for random variables $X_1, X_2, \ldots, X_n$ then can be written as

$$H(X_1, X_2, \ldots, X_n) = H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) + \cdots + H(X_n|X_1, X_2, \ldots, X_{n-1}).$$

Another property of the above notions is that conditioning can only reduce the entropy: $H(X) \geq H(X|Y)$. This is because the knowledge about $Y$ cannot possibly reduce your knowledge about $X$.

The final fact we need is that the random variable with low entropy is essentially determined, meaning that most of the time it takes one single value. One then can show that the pair of variables with low conditional entropy are predictable from each other. Let's formalize these statement in the following lemma (with very loose bounds):

**Lemma 3.1.** *Let $\alpha \geq 0$.*

1. *If $H(X) \leq \alpha$, then there exists $x$ such that $\mathbf{Pr}[X \neq x] \leq \alpha$.*

2. *If $H(X|Y) \leq \alpha$, then given that $Y = y$ you can predict the value of $X$ with high probability. In other words, denote*

$$A(y) = \operatorname*{argmax}_x \mathbf{Pr}[X = x | Y = y].$$

*Then $\mathbf{Pr}_{X,Y}[X \neq A(Y)] \leq \alpha$.*

## 4 Polarizing transformations

We now return to designing a good linear compression scheme. Let's see what can we imply from the existence of some "good" (i.e., satisfying the conditions of Theorem 2.2) linear compressor $A \in \mathbb{F}_2^{m \times n}$.

When $A$ has a full rank, we can substack some matrix $B \in \mathbb{F}_2^{(n-m)\times n}$ to the bottom of $A$ to obtain a square full-rank matrix $P \in \mathbb{F}_2^{n \times n}$. Note that this means that $P$ is an invertible matrix. Now let $Z \sim Bern(p)^{\otimes n}$, so $H(Z) = \sum H(Z_i) = H(Bern(p))n = h(p)n$. Since the matrix $P$ is invertible (one-to-one map), the vector $W = PZ$ carries the same amount of information as $Z$, and so $H(PZ) = H(Z) = h(p)n$. Let's now write the chain rule for the vector $W = PZ$:

$$H(W) = H(W_1) + H(W_2|W_1) + H(W_3|W_1, W_2) + \cdots + H(W_n|W_1, W_2, \ldots, W_{n-1}) = h(p)n.$$

Recall now that $A$ is a linear compressor, which means that we can predict $Z$ with high probability from the vector $AZ$, and then we clearly can predict the vector $BZ$ with high probability. But $AZ$ just forms the first $m$ coordinates of $W$, while the vector $BZ$ is the last $(n-m)$ coordinates of $W$. Thus it follows that

$$H(W_j|W_{<j}) \leq H(W_j|\underbrace{W_1, W_2, \ldots, W_m}_{AZ}) \to 0 \qquad \text{for } j > m.$$

Recall now that $m \approx h(p)n$ for $A \in \mathbb{F}_2^{m\times n}$ being a "good" linear compressor, and that $H(W_i|W_{<i}) \leq 1$ because $|\mathrm{supp}(W_i)| = |\{0,1\}| = 2$. Then

$$H(W) = \sum_{i=1}^{m \approx h(p)n} H(W_i|W_{<i}) + \sum_{j=m+1}^{n} \underbrace{H(W_j|W_{<j})}_{\to 0} = h(p)n,$$

so the sum of $\approx h(p)n$ values, which are $\leq 1$, is almost equal to $h(p)n$. Therefore, we can conclude

$$H(W_l|W_{<l}) \to 1 \qquad \text{for } l = 1, 2, \ldots, m.$$

The above matrix $P$ has a property that it maps a vector $Z \sim Bern(p)^{\otimes n}$, for which all the conditional entropies $H(Z_j|Z_{<j})$ are equal to $h(p)$ (since $Z_j$ are independent), to a vector $W$ which has all such conditional entropies $H(W_j|W_{<j})$ close to either 0 or 1. We call the matrices with such property *polarizing*.

Above we constructed a polarizing matrix $P$ from a linear compression $A$. It turns out the there is also a reverse construction: if we find some polarizing matrix, we can build a linear compression. Below we formalize the definition of polarizing matrices, and we will further show how exactly one can construct a linear compressor out of such a matrix.

**Definition 4.1.** *The family of invertible transformations $P_n : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is called polarizing if for any $\gamma \in (0, 1)$ and for $Z \sim Bern(p)^{\otimes n}$ it holds*

$$\lim_{n\to\infty} \frac{|\{i \mid H\left((PZ)_i|(PZ)_{<i}\right) \in (\gamma, 1-\gamma)\}|}{n} = 0.$$

In fact, it will be more convenient for us to use finite definition of polarizing transformations:

**Definition 4.2.** *The invertible transformation $P \in \mathbb{F}_2^{n\times n}$ is called $(\varepsilon, \tau)$-polarizing for $Z \sim Bern(p)^{\otimes n}$ if for $W = PZ$ and*

$$S_\tau = \{i \mid H(W_i|W_{<i}) \geq \tau\}$$

*it holds $|S_\tau| \leq (h(p) + \varepsilon)n$.*

One should think of taking $\varepsilon$ small and $\tau = o\left(\frac{1}{n}\right)$ in the above definition. It is easy to show by averaging that $|S_\tau| \geq h(p)n - \tau n$, so the $(\varepsilon, \tau)$-polarizing matrices are those for which the set $|S_\tau|$ is (almost) the smallest that it can be.

## 5 Compression and Decompression from Polarizing matrix

Given a $(\varepsilon, \tau)$-polarizing matrix $P$, we describe how one can get a compressing matrix $H$, and decompressing algorithms. First, define the notion of upredictable set of an $(\varepsilon, \tau)$- polarizing matrix $P$.

**Definition 5.1** (Unpredictable Set)**.** *Let $P$ be an $(\varepsilon, \tau)$- polarizing matrix. Let $Z$ is distributed as $Bern(p)^{\otimes n}$, and $W = PZ$. The set $S = \{i|H(W_i|W_{<i}) \geq \tau\}$ is called an unpredictable set of the matrix $P$.*

From the definition of polarizing matrix, we can deduce that the cardinality of $S$ is at most $(h(p) + \varepsilon)n$. Given a $(\varepsilon, \tau)$-polarizing matrix $P$ with unpredictable set $S$, consider the rows $i$ such that $i \in S$. We can remove the rest of the rows from $P$ to obtain the matrix $H : |S| \times n$. Turns out this $H$ can be used as a good compressing matrix. Intuitively, most of the "information" of $PZ$ is stored in the entries of $S$.

We now describe the compression and decompression algorithms given $P$ and $S$.

**Algorithm 1** Polar Compressor $C(P, S, Z)$

```
1: $H = P_S$
2: $W \leftarrow HZ$
3: **return** $W$
```

**Algorithm 2** Successive Cancellation Decompressor $SCD(P, S, W)$

```
1: for  $i = 1\,\mathrm{to}\,n$ do
2:     if  $i \in S$ then
3:         $\tilde{W}_i \leftarrow W_i$
4:     else
5:         $\tilde{W}_i \leftarrow \arg\max_{b\in\mathbb{F}_2} \left\{\mathbf{Pr}\left[W_i = b | W_{<i} = \tilde{W}_{<i}\right]\right\}$
6:     end if
7: end for
8: $\tilde{Z} \leftarrow P^{-1}\tilde{W}$
9: return $\tilde{Z}$
```

**Lemma 5.2.** *The error probability of the above compression scheme is at most $n\tau$, i.e.*

$$\mathop{\mathbf{Pr}}_{Z\sim Bern(p)^{\otimes n}} [SCD(P, S, C(P, S, Z)) \neq Z] \leq n\tau$$

*Proof.* We can rewrite the error term as follows :

$$\mathop{\mathbf{Pr}}_{Z\sim Bern(p)^{\otimes n}} [SCD(P, S, C(P, S, Z)) \neq Z] \tag{1}$$

$$= \mathop{\mathbf{Pr}}_{Z\sim Bern(p)^{\otimes n}} [SCD(P, S, W = HZ) \neq Z] \tag{2}$$

$$= \mathop{\mathbf{Pr}}_{Z\sim Bern(p)^{\otimes n}} \left[\tilde{W} \neq W\right] \tag{3}$$

where in the last step, we have used the fact that $P$ is an invertible matrix. Using union bound, we can rewrite it further as following

$$\sum_i \mathop{\mathbf{Pr}}_{Z\sim Bern(p)^{\otimes n}} \left[\left(\tilde{W}_i \neq W_i\right) \cap \left(\tilde{W}_{<i} = W_{<i}\right)\right] \tag{4}$$

$$\leq \sum_{i\notin S} \mathop{\mathbf{Pr}}_{Z\sim Bern(p)^{\otimes n}} \left[\tilde{W}_i \neq W_i | \tilde{W}_{<i} = W_{<i}\right] \tag{5}$$

$$\leq \sum_{i\notin S} \tau \leq n\tau \tag{6}$$

The last step is obtained from using the fact that $H(W_i|W_{<i}) \leq \tau$, and applying Lemma 3.1. $\square$

Before proceeding further, we note a couple of points.

- First, the above algorithm is a generic scheme to compress and decompress and we have not mentioned how to make it efficient yet. After we introduce the actual polarizing matrix in the next section, we redo the above scheme efficiently.

- Second, recall that in Lemma 2.1 the encoding and decoding complexity for the capacity-achieving code, which we obtain from the compressor $H$, depend also on the complexity of multiplying by the generator matrix $G$ and its inverse $G^*$. Using the fact that $P$ is invertible and that $H = P_S$, one can show that in fact $G = (P^{-1})_{\overline{S}}$ and $G^* = P_{\overline{S}}$, where $\overline{S} = \{1, 2, \ldots, n\} \setminus S$. This means that multiplication by $G$ and $G^*$ is not harder than just multiplication by $P$ and $P^{-1}$.

- Finally, recall that the error probability of our compression scheme directly corresponds to the error probability of our code for the BSC channel in Lemma 2.1. So, in order to obtain inverse polynomial error, we set $\tau$ to $\frac{1}{n^c}$ later. From the definition of polarizing matrix, it should be clear that $\varepsilon$ corresponds to the gap to capacity of the code.

## 6 A Polarizing Transform

In this section, we present a recursive construction of a polarizing transform.

Consider the case when $n = 2$ where the goal is to polarize a pair of two independent $Bern(p)$ variables $Z_1$ and $Z_2$. Consider the transform $P_2$ that maps $Z = \{Z_1, Z_2\}$ to $W = \{W_1 = Z_1 + Z_2, W_2 = Z_2\}$ depicted below.

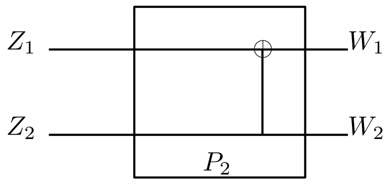

Figure 1: Polarizing transform for $n = 2$

Note that this transform corresponds to the matrix $P_2$.

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Using the independence of $Z_1$ and $Z_2$, we get $H(Z_1, Z_2) = H(Z_1) + H(Z_2) = 2h(p)$. Since the transform $P_2$ is invertible, $H(W_1, W_2) = H(Z_1, Z_2)$. $W_1$ is a Bernoulli random variable with $\mathbf{Pr}[W_1 = 1] = 2p(1-p)$. Thus, $H(W_1) = h(2p(1-p))$. Using chain rule, we can find the conditional entropy $H(W_2|W_1) = H(W_1, W_2) - H(W_1) = 2h(p) - h(2p(1-p))$. Since $H$ is an increasing function in $(0, \frac{1}{2})$, $H(W_1) = h(2p(1-p)) > h(p)$, and using the above summation, $H(W_2|W_1) = 2h(p) - H(W_1) < h(p)$.

Thus, we have achieved a mild polarization of the conditional entropy. In order to achieve this for higher $n$, while also pushing (most of) the conditional entropy values to either 0 or 1, we apply the same construction recursively.

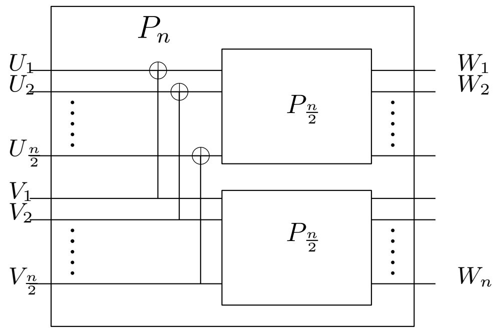

Figure 2: Recursive construction of a polarizing transform

Let $U = (Z_1, Z_2, .., Z_{\frac{n}{2}})$ and $V = (Z_{\frac{n}{2}+1}, Z_{\frac{n}{2}+2}, .., Z_n)$ denote the first and second half vectors of $Z$. Henceforth, we restrict ourselves to $n$ being a power of 2. We can define $P_n$ formally as below:

$$\begin{aligned} P_n(Z) =& (P_{\frac{n}{2}}(U+V), P_{\frac{n}{2}}(V)) \\ P_1 =& (1) \end{aligned}$$

We can also write $P_n$ as $P_2$ tensor product with itself $\log(n)$ times: $P_n = P_2^{\otimes \log(n)}$. It is not hard to see that indeed $P_n$ is an invertible matrix for every $n$. But more importantly, it turns out that this simple recursive construction achieves strong polarization!

**Theorem 6.1** (Polynomially strong polarization)**.** *For every $\varepsilon > 0$ and constant $c > 0$, there exists $n_0 = poly(\frac{1}{\varepsilon})$ such that for all $n \geq n_0$, $P_n$ is $(\varepsilon, \frac{1}{n^c})$-polarizing.*

We will prove this Theorem in the next lecture.

## 7 Compression and Decompression Algorithms

In this section, we show how the compressing and decompressing algorithms of Section 5 can be efficiently implemented for $P_n$. We assume that we are also given the unpredictable set $S$ along with the polarizing matrix $P = P_n$.

### 7.1 Compression

In order to compute $PZ$, we can first compute $U + V$ and recursively compute $P_{\frac{n}{2}}(U+V)$, and $P_{\frac{n}{2}}(V)$, and then merge them to get $P_n(Z)$. Once we compute $PZ$, we can eliminate some entries to obtain $HZ$ using $S$.

**Algorithm 3** Polar Compressor $C(P_n, Z)$

```
1: if $n = 1$ then
2:     return $Z$
3: end if
4: $U = (Z_1, .., Z_{\frac{n}{2}}), V = (Z_{\frac{n}{2}+1}, .., Z_n)$
5: $W_1 \leftarrow C(P_{\frac{n}{2}}, U + V)$
6: $W_2 \leftarrow C(P_{\frac{n}{2}}, V)$
7: return $(W_1, W_2)$                                        ▷ Returns $PZ$
```

The running time of algorithm follows the recursion $T(n) = 2T(n/2) + O(n)$, and thus is $O(n \log n)$.

The above shows that the multiplication by $P$ takes $O(n \log n)$ time. In fact, it is not hard to show that $P$ is self-inverse, i.e. $P = P^{-1}$, which means that the multiplication by $P^{-1}$ can also be done in $O(n \log n)$ time. Combining this with the comment at the end of section 5, we obtain that the complexity of multiplication by $G$ and $G^*$ in Lemma 2.1 takes $O(n \log n)$ time. This completely justifies our claim that the Theorem 1.2 will follow from Theorem 2.2 by Lemma 2.1, for this particular linear compressor (since in general matrix-vector multiplication cannot be done in $O(n \log n)$ time).

## 7.2 Decompression

In order to apply successive cancellation decoding algorithm to our polarizing matrix $P_n$, it suffices if we can efficiently compute the probability $\mathbf{Pr}\left[W_i = 1 | W_{<i} = a\right]$ for a given $a$ and $i$. Since we don't have any structural understanding about $P_n$ other than the recursive construction, it seems a good idea to try to recursively compute the probabilities.

There is a small catch - the inputs no longer need to be $Bern(p)$. For example, consider the probability $\mathbf{Pr}\left[W_2 = 0 | W_1 = 1\right]$, and assume that $n > 2$. We can calculate this by recursively calling the same probability on $P_{\frac{n}{2}}$ matrix whose input is $U + V$. However, $U + V$ is not distributed as $Bern(p)^{\otimes \frac{n}{2}}$. But the probability distribution of $U + V$ is infact $Bern(q)^{\otimes \frac{n}{2}}$, for $q = 2p(1-p)$.

Consider the conditional probability $\mathbf{Pr}\left[W_i = 1 | W_{<i} = a\right]$ when $i > \frac{n}{2}$. In order to condition on the knowledge of $(W_1, .., W_{\frac{n}{2}})$, we can instead condition on the knowledge of $U + V$, since $P_{\frac{n}{2}}$ is an invertible transform. However, once we condition on a particular value being assigned to $U + V$, the probability distribution of $V$ changes. Like in previous case, it turns out the new probability distribution is an independent Bernoulli distribution as well.

This suggests that we should really be solving a more general problem, where the input $Z$ is distributed as $Bern(p_1) \times Bern(p_2) \times .. \times Bern(p_n)$. The recursive computation of conditional probabilities can be combined with the SCD algorithm in a neat way presented below. We assume the input is formatted to be in $\{0, 1, ?\}^n$ by adding $'?'$ characters in positions $i$ such that $i \notin S$.

Let $q = b^+(p_1, p_2)$ be defined such that $Bern(q)$ is the distribution of $X + Y$ when $X$ and $Y$ are independent Bernoulli random variables distributed as $Bern(p_1)$ and $Bern(p_2)$. Similarly, let $q = b^|(p_1, p_2, b)$ be defined such that $Bern(q)$ is the distribution of $Y$ conditioned on $X + Y$ equals $b$, when $X$ and $Y$ are independent Bernoulli random variables distributed as $Bern(p_1)$ and $Bern(p_2)$. We assume that we can compute $b^+$ and $b^|$ in constant time.

The running time of the algorithm can be shown to be $O(n \log n)$ using the same argument as Algorithm 3. We argue about the correctness in the lemma below.

**Lemma 7.1.** *If $P_n$ is a $(\varepsilon, \tau)$-polarizing matrix, the error probability of the above algorithm is at most $n\tau$.*

*Proof.* Observe that we are implicitly implementing Algorithm 2 in the algorithm. The crucial step of SCD algorithm, line 5 is implemented here in the $n = 1$ case. However, we are computing the $\rho$ values recursively, and then using the $\rho$ values as surrogates for conditional probabilities $\mathbf{Pr}\left[W_i = 1 | W_{<i} = \tilde{W}_{<i}\right]$. Applying Lemma 5.2, to argue about the correctness of the algorithm, it suffices if we prove that $\rho_i = \mathbf{Pr}\left[W_i = 1 | W_{<i} = \tilde{W}_{<i}\right]$.

We prove this statement by induction on $n$. When $n = 1$, we simply return $\rho_1 = p_1$, which corresponds to probability that $W_1 = 1$. When $n > 1$, consider two cases separately :

1. $i \leq \frac{n}{2}$, the first recursive $P_{\frac{n}{2}}$. Here, the input to $P^{\frac{n}{2}}$ distributed as $Bern(q_1) \times .. \times$

**Algorithm 4** Polar Decompressor $SCD(P_n, W)$

```
1: $(\tilde{W}, \rho) \leftarrow RPD(W, n, (p, p, .., p))$
2: $\tilde{Z} \leftarrow {P_n}^{-1}\tilde{W}$    ▷ Inverting by $P_n$ is same as multiplication by $P_n$
3: return $\tilde{Z}$
4: procedure Recursive Polar Decoder $RPD(W, n, (p_1, .., p_n))$
5:     if $n = 1$ then
6:         if $W_1 \in \mathbb{F}_2$ then
7:             return $(W_1, p_1)$
8:         else
9:             if $p_1 \geq 0.5$ then
10:                return $(1, p_1)$
11:            else
12:                return $(0, p_1)$
13:            end if
14:        end if
15:    else
16:        $W^{(1)} \leftarrow (W_1, .., W_{\frac{n}{2}}), W^{(2)} \leftarrow (W_{\frac{n}{2}+1}, .., W_n)$
17:        for $i = 1 \text{ to } \frac{n}{2}$ do
18:            $q_i \leftarrow b^+(p_i, p_{\frac{n}{2}+i})$    ▷ Computing input probabilities for first half
19:        end for
20:        $(\tilde{W}^1, \rho^{(1)}) \leftarrow RPD(W^{(1)}, \frac{n}{2}, (q_1, .., q_{\frac{n}{2}}))$
21:        $\tilde{X} \leftarrow (P_{\frac{n}{2}})^{-1}(\tilde{W}^1)$
22:        for $i = 1 \text{ to } \frac{n}{2}$ do
23:            $r_i \leftarrow b^|(p_i, p_{\frac{n}{2}+i}, \tilde{X}_i)$    ▷ Computing input probabilities for second half
24:        end for
25:        $(\tilde{W}^2, \rho^{(2)}) \leftarrow RPD(W^{(2)}, \frac{n}{2}, (r_1, .., r_{\frac{n}{2}}))$
26:        $\tilde{W} \leftarrow (\tilde{W}^1, \tilde{W}^2), \rho \leftarrow (\rho^{(1)}, \rho^{(2)})$
27:        return $(\tilde{W}, \rho)$
28:    end if
29: end procedure
```

$Bern(q_{\frac{n}{2}})$.

$$\underset{Z\sim Bern(p_1)\times..\times Bern(p_n)}{\mathbf{Pr}}\left[W_i = 1 | W_{<i} = \tilde{W}_{<i}\right] \tag{7}$$
$$= \underset{U+V\sim Bern(q_1)\times..\times Bern(q_{\frac{n}{2}})}{\mathbf{Pr}}\left[W_i = 1 | W_{<i} = \tilde{W}_{<i}\right] \tag{8}$$
$$=\rho_i^{(1)} \text{ (by inductive claim)} \tag{9}$$
$$=\rho_i \tag{10}$$

2. When $i > \frac{n}{2}$, we have

$$\underset{Z\sim Bern(p_1)\times..\times Bern(p_n)}{\mathbf{Pr}}\left[W_i = 1 | W_{<i} = \tilde{W}_{<i}\right] \tag{11}$$
$$= \underset{Z\sim Bern(p_1)\times..\times Bern(p_n)}{\mathbf{Pr}}\left[W_i = 1 | W^1 = \tilde{W}^1, (W_{\frac{n}{2}+1}, .., W_i) = (\tilde{W}_{\frac{n}{2}+1}, .., \tilde{W}_i)\right] \tag{12}$$
$$= \underset{Z\sim Bern(p_1)\times..\times Bern(p_n)}{\mathbf{Pr}}\left[W_i = 1 | U + V = \tilde{X}, (W_{\frac{n}{2}+1}, .., W_i) = (\tilde{W}_{\frac{n}{2}+1}, .., \tilde{W}_i\right] \tag{13}$$
$$= \underset{V\sim Bern(r_1)\times..\times Bern(r_{\frac{n}{2}})}{\mathbf{Pr}}\left[W_i = 1 | (W_{\frac{n}{2}+1}, .., W_i) = (\tilde{W}_{\frac{n}{2}+1}, .., \tilde{W}_i\right] \tag{14}$$
$$=\rho^{(2)}_{i-\frac{n}{2}} \text{ (by inductive claim)} \tag{15}$$
$$=\rho_i \tag{16}$$

where in, $W^1$ stands for $(W_1, W_2, .., W_{\frac{n}{2}})$.

□

## 8 Introduction to Analysis of Polarization

Now, we are left with analyzing the polarization of $P_n$ i.e. proving Theorem 6.1. However, we use a slightly different matrix $R_n$ to analyze the polarization. We show that if $R_n$ is $(\varepsilon, \tau)$-polarizing, then so is $P_n$.

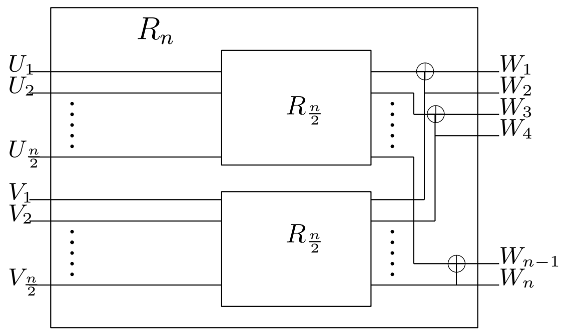

Figure 3: Recursive construction of a different polarizing transform

More formally, for every $1 \le i \le \frac{n}{2}$, $R_n(U,V)_{2i-1} = R_{\frac{n}{2}}(U)_i + R_{\frac{n}{2}}(V)_i$ and $R_n(U,V)_{2i} = R_{\frac{n}{2}}(V)_i$. To prove that strong polarizing of $R_n$ implies strong polarizing of $P_n$, we can proceed along the following lines :

1. First, we can show that $R_n$ is a "bit reversal" of $P_n$ i.e. $R_n = B_n P_n$, where $B_n$ is a permutation matrix that reverses the permutation.

2. Next, we show that $B_n$ and $P_n$ commute with each other i.e. $B_n P_n = P_n B_n$.

3. Using the previous part, we can show that we can obtain distribution of $P_n Z$ from $R_n Z$ by reversing the input bits, instead of the output.

4. Since the distribution of $Z$ is unchanged under bit reversal, the distribution of $P_n Z$ and $R_n Z$ is identical, which proves the polarizing claim.

## References

[Ari09] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, 2009.

[GX15] Venkatesan Guruswami and Patrick Xia. Polar codes: Speed of polarization and polynomial gap to capacity. *IEEE Transactions on Information Theory*, 61(1):3–16, 2015.