A Brief Introduction to Polar Codes

Notes for Introduction to Error-Correcting Codes
Henry D. Pfister

October 8th, 2017

1 Introduction

Polar codes were introduced by Erdal Arıkan in 2009 and they provide the first deterministic construction of capacity-achieving codes for binary memoryless symmetric (BMS) channels [1]. They are the culmination of Arıkan's research into the computational cutoff rate of sequential decoding.

In particular, consider a setup where (U_1, U_2) are two equiprobable bits that are encoded into $(X_1, X_2) = (U_1 \oplus U_2, U_2)$. Then, (X_1, X_2) are mapped to (Y_1, Y_2) by two independent BMS channels with transition probabilities $\mathbb{P}(Y_1 = y \,|\, X_1 = x) = \mathbb{P}(Y_2 = y \,|\, X_2 = x) = W(y|x)$. The Tanner graph for this setup is shown in Figure 1. Since the mapping from (U_1, U_2) to (X_1, X_2) is invertible, one finds that

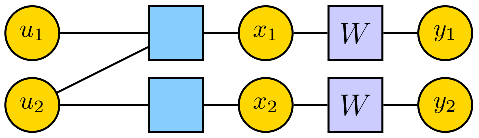

Figure 1: Factor Graph

$$I(U_1, U_2; Y_1, Y_2) = I(X_1, X_2; Y_1, Y_2) = I(X_1; Y_1) + I(X_2; Y_2) = 2I(W),$$

where $C = I(X_1; Y_1) = I(W)$ is the capacity of symmetric BMS channel because X_1 is equiprobable and

$$I(W) \triangleq \sum_y W(y|0) \log_2 \left[W(y|0) / \left(\tfrac{1}{2} W(y|0) + \tfrac{1}{2} W(y|1) \right) \right] = I(X_1; Y_1).$$

Thus, the transformation $(X_1, X_2) = (U_1 \oplus U_2, U_2)$ preserves the sum capacity of the system. Readers unfamiliar with information theory should consult the primer in Appendix A.

Also, the chain rule decomposition

$$I(U_1, U_2; Y_1, Y_2) = I(U_1; Y_1, Y_2) + I(U_2; Y_1, Y_2 | U_1) = 2I(W)$$

implies that one can also achieve the rate $2I(W)$ using two steps. First, information is transmitted through the first virtual channel $W^- \colon U_1 \to (Y_1, Y_2)$. By coding over multiple channel uses, one can achieve any rate up to $I(U_1; Y_1, Y_2)$. If all the U_1's are known from the first stage, then one can decode the information transmitted through the second virtual channel $W^+ \colon U_2 \to (Y_1, Y_2, U_1)$. Again, coding allows one to achieve any rate up to $I(U_2; Y_1, Y_2 | U_1)$.

If one chooses convolutional codes with sequential decoding, however, the expected decoding-complexity per bit becomes unbounded for rates above the computational cutoff rate

$$R_0(W) = 1 - \log_2(1 + Z(W)),$$

where $Z(W) \triangleq \sum_y \sqrt{W(y|0)W(y|1)}$ is the Bhattacharyya parameter of the channel [2]. Arıkan's key observation was that, while the sum capacity of the two virtual channels is preserved, the sum cutoff rate satisfies

$$R_0(W^+) + R_0(W^-) \geq 2R_0(W),$$

with equality iff $R_0(W) \in \{0, 1\}$. Thus, repeated transformations of this type cause the implied virtual channels to polarize into extremal channels whose capacities approach 0 or 1. But, for these extremal channels, coding is trivial and one either sends an information bit or a dummy bit.

From a theoretical point of view, polar codes are beautifully simple. Practically, they approach capacity rather slowly as their blocklength increases [3]. Still, based on recent advances [4], they have been chosen as the short-block codes for the 5G standard.

2 Mathematical Background

2.1 Kronecker Products

For a positive integer n, let $N = 2^n$ and let $(d_0 d_1 \ldots d_{n-1})_2 \triangleq 1 + \sum_{i=0}^{n-1} d_i 2^i$ denote the integer (indexed from 1) with binary digit string $d_0 d_1, \ldots, d_{n-1}$. Much of the following material can be found in [5]. For matrices $A = \{a_{i,j}\}$ and $B = \{b_{i,j}\}$, let $A \otimes B$ represent the Kronecker product

$$A \otimes B \triangleq \begin{bmatrix} a_{1,1}B & a_{1,2}B & \cdots \\ a_{2,1}B & a_{2,2}B & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

Kronecker powers are defined by $A^{\otimes n} \triangleq A \otimes A^{\otimes n-1} = A^{\otimes n-1} \otimes A$. The following well-known identity will also be useful

$$(A \otimes B)(C \otimes D) = AC \otimes BD.$$

An $N \times N$ permutation matrix P_N satisfies $P_N^T P_N = I_N$, where I_N is the $N \times N$ identity matrix, because it preserves the Euclidean distance between pairs of vectors and is therefore orthogonal. An important permutation matrix is the "mod-p perfect shuffle" $S_{p,q}$ with $N = pq$. Like all permutation matrices, it is defined by where it maps the unit row vectors e_m for $m = 1, 2, \ldots, N$. In particular, $S_{p,q}$ is defined by $e_{m'}^T = S_{p,q} e_m^T$ iff $m' = \lfloor (m-1)/p \rfloor + 1 + (m - 1 \bmod p)q$. For example, we have

$$\begin{aligned} S_{2,M/2}(s_1, s_2, \ldots, s_M)^T &= (s_1, s_3, \ldots, s_{M-1}, s_2, s_4, \ldots, s_M)^T \\ S_{3,M/3}(s_1, s_2, \ldots, s_M)^T &= (s_1, s_4, \ldots, s_{M-2}, s_2, s_5, \ldots, s_{M-1}, s_3, s_6, \ldots, s_M)^T. \end{aligned}$$

Let A and B be $m_1 \times n_1$ and $m_2 \times n_2$ matrices, respectively. Then, mod-p perfect shuffle allows one to write the identity

$$B \otimes A = S_{m_1,m_2}^T (A \otimes B) S_{n_1,n_2}.$$

The basic idea is that $A \otimes B$ differs from $B \otimes A$ only in the order of the rows and columns and mod-p shuffle matrices can be used to rearrange the rows and columns of $A \otimes B$ into those of $B \otimes A$. In particular, if A is a 2×2 matrix and B is an $(N/2) \times (N/2)$ matrix, then

$$\begin{aligned} B \otimes A &= S_{2,N/2}^T (A \otimes B) S_{2,N/2} \\ &= R_N (A \otimes B) R_N^T, \end{aligned} \tag{1}$$

where $R_N = S_{2,N/2}^T$ denotes the $N \times N$ reverse shuffle permutation matrix defined by

$$(s_1, s_2, \ldots, s_N) R_N = (s_1, s_3, \ldots, s_{N-1}, s_2, s_4, \ldots, s_N).$$

It is easy to verify that $R_N = S_{2,N/2}^T$ by transposing the previous equation.

Looking closely, we find that if $m = (d_0 d_1 \ldots d_{n-1})_2$ and $e_{m'} = e_m R_N$, then $m' = (d_1 d_2 \ldots d_{n-1} d_0)_2$. Therefore, R_N first reorders the vector so that the base-2 expansion of the index experiences a left circular shift. This maps the least significant bit (LSB) of the index to the most significant bit (MSB) of the index and preserves the order of the other $n-1$ bits of the index. Now, the $N \times N$ bit reversal permutation matrix B_N can be defined recursively with in terms of R_N using

$$B_N = R_N \begin{bmatrix} B_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} = R_N (I_2 \otimes B_{N/2}).$$

This holds because applying R_N reorders the vector so that the LSB of the index becomes the MSB of the index. Then, multiplying by $I_2 \otimes B_{N/2}$ separately reorders the first and last halves of the vector according to bit reversals of length $N/2$. Since B_N reverses the bit indexing, it is symmetric and satisfies $B_N^T = B_N$. Thus, we find also that

$$B_N = B_N^T = \begin{bmatrix} B_{N/2}^T & 0 \\ 0 & B_{N/2}^T \end{bmatrix} R_N^T = \begin{bmatrix} B_{N/2} & 0 \\ 0 & B_{N/2} \end{bmatrix} R_N^T = (I_2 \otimes B_{N/2}) R_N^T.$$

2.2 Soft Decoding

For simplicity, we will also assume that channel outputs are normalized into a posteriori probability (APP) estimates that are either log-likelihood ratios (LLRs) or "probability of 1" (P1) values. For example, an LLR-normalized channel output must satisfy

$$y_1 = \ln \frac{\mathbb{P}(Y_1 = y_1 | X_1 = 0)}{\mathbb{P}(Y_1 = y_1 | X_1 = 1)}$$

while a P1-normalized channel output must satisfy

$$y_1 = \mathbb{P}(X_1 = 1 | Y_1 = y_1) = \frac{\mathbb{P}(Y_1 = y_1 | X_1 = 1)}{\mathbb{P}(Y_1 = y_1 | X_1 = 0) + \mathbb{P}(Y_1 = y_1 | X_1 = 1)}.$$

Using this, the bit-node and check-node operations used to combine estimates in LDPC decoding are defined, respectively, by

$$y_1 \circledast y_2 = \begin{cases} y_1 + y_2 & \text{if LLR domain} \\ \frac{y_1 y_2}{y_1 y_2 + (1-y_1)(1-y_2)} & \text{if P1 domain.} \end{cases}$$

and

$$y_1 \boxast y_2 = \begin{cases} 2\tanh^{-1}\left(\tanh(y_1/2)\tanh(y_2/2)\right) & \text{if LLR domain} \\ y_1(1-y_2) + y_2(1-y_1) & \text{if P1 domain.} \end{cases}$$

For convenience, the operation $\boxast$ is also defined between APP values and hard bit values so that channel symmetry can be represented by $W(y|1) = W(1 \boxast y|0)$. For $a, b \in \{0, 1\}$, we also assume that $0 \boxast y = y$, $a \boxast y = y \boxast a$, and $a \boxast (b \boxast y) = (a \oplus b) \boxast y$.

As an example of these operations, consider the $(X_1, X_2) = (U_1 \oplus U_2, U_2)$ mapping defined in the introduction. Let (y_1, y_2) be a pair of normalized observations of (X_1, X_2). Then, $\tilde{u}_1 = y_1 \boxast y_2$ is the optimal normalized bit estimate of U_1 given (Y_1, Y_2). Using this, $\tilde{u}_2 = (u_1 \boxast y_1) \circledast \tilde{y}_2$ is the optimal normalized bit estimate of U_2 given (Y_1, Y_2, U_1).

3 The Polar Transformation

3.1 Algebraic and Factor-Graph Description

The polar transform of size N is defined to be

$$G_N \triangleq B_N \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{\otimes n} = B_N G_2^{\otimes n},$$

Since $B_2 = I_2$, one finds that

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

and this transform is a fundamental building block (e.g., see Figure 1) in Arıkan's construction of polar codes. The main idea is to construct an message vector $\underline{u}$ where the elements u_i with $i \in \mathcal{A} \subseteq \{1, 2, \ldots, N\}$ carry information and the other elements u_j with $j \in \mathcal{A}^c$ contain values known at the transmitter and receiver (e.g., are fixed to 0's). Then, the codeword $\underline{x} = \underline{u} G_N$ is transmitted over the channel.

Various recursive definitions of G_N will also be useful. For example, the recursion $B_N = R_N(I_2 \otimes B_{N/2})$ implies that

$$\begin{aligned} G_N &= R_N(I_2 \otimes B_{N/2}) G_2^{\otimes n} \\ &= R_N(I_2 \otimes B_{N/2})(G_2 \otimes G_2^{\otimes n-1}) \\ &= R_N(G_2 \otimes B_{N/2} G_2^{\otimes n-1}) \\ &= R_N(G_2 \otimes G_{N/2}) \end{aligned}$$

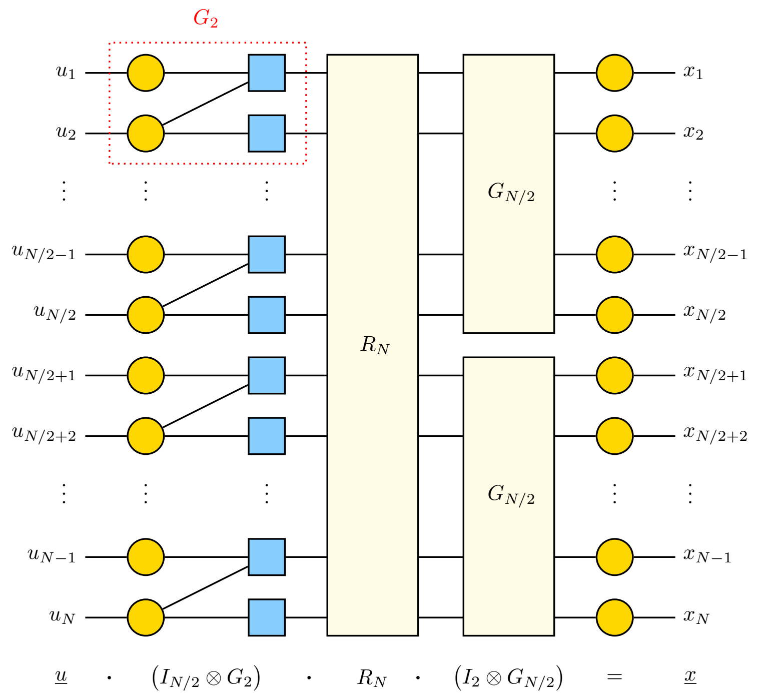

Figure 2: Factor graph associated with $G_N = (I_{N/2} \otimes G_2)R_N(I_2 \otimes G_{N/2})$ decomposition.

$$= R_N(G_2 \otimes I_{N/2})(I_2 \otimes G_{N/2}).$$

With this result, we see that computing $\underline{x} = \underline{u}G_N$ is equivalent to a reverse shuffle of $\underline{u}$ followed by an expanded G_2-butterfly section and then separate polar transforms of length-$N/2$ applied to the first and second halves of the vector. From (1), we know $R_N(G_2 \otimes I_{N/2})R_N^T = (I_{N/2} \otimes G_2)$ and this implies that

$$\begin{aligned} G_N &= R_N(G_2 \otimes I_{N/2})(I_2 \otimes G_{N/2}) \\ &= (I_{N/2} \otimes G_2)R_N(I_2 \otimes G_{N/2}). \end{aligned}$$

The factor graph associated with this decomposition is shown Figure 2. From this, we see that computing $\underline{x} = \underline{u}G_N$ using this formula is equivalent to a local G_2-butterfly section followed by a reverse shuffle and then separate polar transforms of length-$N/2$ applied to the first and second halves of the vector. This representation leads naturally to the recursive encoding algorithm implied by [1] and implemented below in Algorithm 1.

Algorithm 1 Recursive Implementation of Polar Transform in Matlab

```
function x = polar_transform(u)

  % Recurse down to length 1
  if (length(u)==1)
    x = u;
  else
    % Compute odd/even outputs of (I_{N/2} \otimes G_2) transform
    u1u2 = mod(u(1:2:end)+u(2:2:end),2);
    u2 = u(2:2:end);

    % R_N maps odd/even indices (i.e., u1u2/u2) to first/second half
    x = [polar_transform(u1u2) polar_transform(u2)];
  end
```

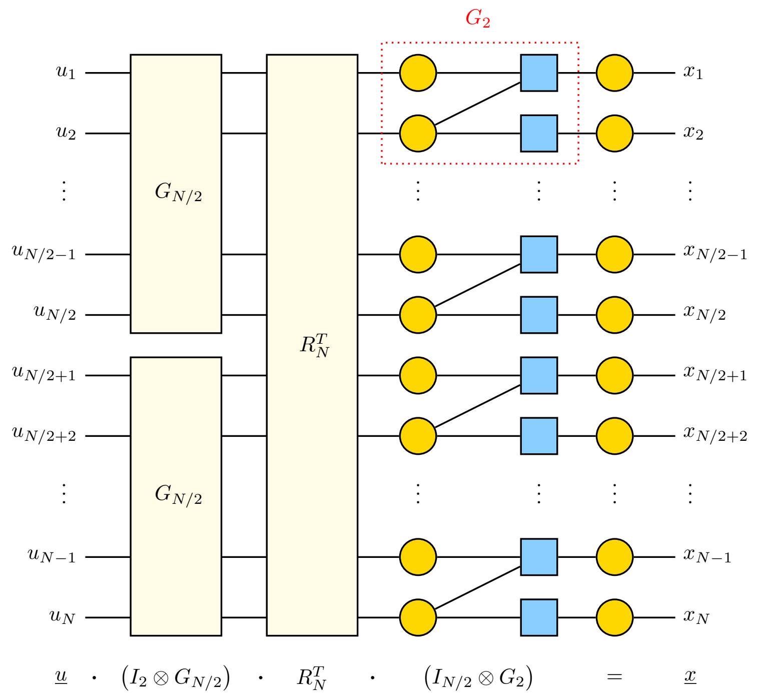

Figure 3: Factor graph associated with $G_N = (I_2 \otimes G_{N/2})R_N^T(I_{N/2} \otimes G_2)$ decomposition.

Next, we show that B_N commutes with $G_2^{\otimes n}$ and, hence, $G_N = B_N G_2^{\otimes n} = G_2^{\otimes n} B_N$. Since $B_N B_N = I_N$, we use induction to show that $G_N B_N = B_N G_2^{\otimes n} B_N = G_2^{\otimes n}$. For $N = 2$, this is trivial because $B_2 = I_2$. For the inductive step, we write

$$\begin{aligned}
G_N B_N &= B_N G_2^{\otimes n} B_N \\
&= R_N (I_2 \otimes B_{N/2}) G_2^{\otimes n} (I_2 \otimes B_{N/2}) R_N^T \\
&= (B_{N/2} \otimes I_2) G_2^{\otimes n} (B_{N/2} \otimes I_2) R_N R_N^T \\
&= (B_{N/2} \otimes I_2)(G_2^{\otimes n-1} \otimes G_2)(B_{N/2} \otimes I_2) \\
&= ((B_{N/2} G_2^{\otimes n-1}) \otimes G_2)(B_{N/2} \otimes I_2) \\
&= (B_{N/2} G_2^{\otimes n-1} B_{N/2}) \otimes G_2 \\
&= G_2^{\otimes n-1} \otimes G_2 = G_2^{\otimes n}.
\end{aligned}$$

It is worth noting that this implies that $G_N = G_N^{-1}$ because

$$G_N G_N = G_N B_N G_2^{\otimes n} = G_2^{\otimes n} G_2^{\otimes n} = (G_2 G_2)^{\otimes n} = I_2^{\otimes n} = I_N. \tag{2}$$

Using $G_N = G_2^{\otimes n} B_N$ and $B_N = (I_2 \otimes B_{N/2}) R_N^T$, one can also get the alternative recursive decomposition

$$\begin{aligned}
G_N &= G_2^{\otimes n} (I_2 \otimes B_{N/2}) R_N^T \\
&= (G_2 \otimes G_2^{\otimes n-1})(I_2 \otimes B_{N/2}) R_N^T \\
&= (G_2 \otimes (G_2^{\otimes n-1} B_{N/2})) R_N^T \\
&= (G_2 \otimes G_{N/2}) R_N^T \\
&= (I_2 \otimes G_{N/2})(G_2 \otimes I_{N/2}) R_N^T \\
&= (I_2 \otimes G_{N/2}) R_N^T (I_{N/2} \otimes G_2).
\end{aligned}$$

The factor graph associated with this decomposition is shown in Figure 3. From this, we see that this decomposition is quite natural for a recursive successive-cancellation decoding algorithm because one can first perform a "soft-inversion" of $R_N^T(I_{N/2} \otimes G_2)$ using LDPC code message-passing operations and then recursively call the polar decoding algorithm for two length $N/2$ polar transforms defined by $I_2 \otimes G_{N/2}$. We note that the arguments to the polar decoding algorithm are APP estimates of the bits in the output vector and a priori probability estimates for the bits in the input vector (e.g., the a priori estimates identify frozen bits and their values). The output from the polar decoder is a vector of hard-decision bit estimates for the input and output bits of the polar code. The above decomposition leads naturally to the recursive decoding algorithm defined below in Algorithm 2.

Algorithm 2 Recursive Implementation of P1-Domain Polar Decoder in Matlab

```
function [u,x] = polar_decode(y,f)
% y = bit APP from channel in output order % f = input a priori probs in input order
% x = output hard decision in output order % u = input hard decisions in input order

  % Recurse down to length 1
  N = length(y);
  if (N==1)
    if (f==1/2)
      % If info bit, make hard decision based on observation
      x = round(y); u = x;
    else
      % Use frozen bit for output and hard decision for input (for monte carlo design)
      x = f; u = round(y);
    end
  else
    % Compute soft mapping back one stage
    u1est = cnop(y(1:2:end),y(2:2:end));

    % R_N^T maps u1est to top polar code
    [uhat1,u1hardprev] = polar_decode(u1est,f(1:(N/2)));

    % Using u1est and x1hard, we can estimate u2
    u2est = vnop(cnop(u1hardprev,y(1:2:end)),y(2:2:end));

    % R_N^T maps u2est to bottom polar code
    [uhat2,u2hardprev] = polar_decode(u2est,f((N/2+1):end));

    % Tunnel u decisions back up. Compute and interleave x1,x2 hard decisions
    u = [uhat1 uhat2];
    x = reshape([cnop(u1hardprev,u2hardprev); u2hardprev],1,[]);
  end
return

% Check-node operation in P1 domain
function z=cnop(w1,w2)
  z = w1.*(1-w2) + w2.*(1-w1);
return

% Bit-node operation in P1 domain
function z=vnop(w1,w2)
  z = w1.*w2 ./ (w1.*w2 + (1-w1).*(1-w2));
return
```

3.2 Decoding Analysis and Code Design

Consider the recursive successive cancellation decoder for a polar transform of length N and let $\underline{y} = (y_1, \ldots, y_N)$ be the observations of the output bits $\underline{x} = (x_1, \ldots, x_N)$ through N copies of the BMS channel W. To describe the decoder analysis, we focus on the effective channels seen by each of the inputs bits in $\underline{u} = (u_1, \ldots, u_N)$. Let $W_N^{(i)}$ represent the virtual channel seen by u_i during recursive successive-cancellation decoding of a length-N polar transform. The SCD process uses the entire y_1^N vector and all past decisions $\hat{u}_1^{i-1}$ to generate the soft estimate $\tilde{u}_i$ and hard decision $\hat{u}_i$ for bit i. Assuming all past decisions are correct (i.e., $\hat{u}_1^{i-1} = u_1^{i-1}$), the effective channel maps $u_i \in \{0,1\}$ to $(y_1^N, u_1^{i-1}) \in \mathcal{Y}^N \times \{0,1\}^{i-1}$ and is defined to be

$$W_N^{(i)}\left((y_1^N, u_1^{i-1})|u_i\right) \triangleq \sum_{u_{i+1}^N \in \{0,1\}^{N-i}} \frac{1}{2^{N-1}} W_N(y_1^N | u_1^N G_N),$$

where $W_N(y_1^N|x_1^N) = \prod_{i=1}^N W(y_i|x_i)$. This formula represents an experiment where u_1^N is an i.i.d. equiprobable vector encoded into $x_1^N = u_1^N G_N$ and then transmitted through N independent W channels. The decoder receives the channel observations y_1^N and the side information u_1^{i-1}. In the formula, one can interpret the sum over u_{i+1}^N as marginalizing out the "future" input bits that are unknown to the decoder.

Since the output space of the channel $W_N^{(i)}$ depends on N, it is also useful to define the normalized channel $\tilde{W}_N^{(i)}(z|u_i)$, for $z \in \mathcal{Z}$, as the output remapping of $W_N^{(i)}$ defined by

$$(y_1^N, u_1^{i-1}) \to z = \begin{cases} \ln \frac{\mathbb{P}(Y_1^N = y_1^n, U_1^{i-1} = u_1^{i-1}) \,|\, U_i = 0)}{\mathbb{P}(Y_1^N = y_1^n, U_1^{i-1} = u_1^{i-1}) \,|\, U_i = 1)} & \text{for the LLR domain} \\ \mathbb{P}(U_i = 1 \,|\, Y_1^N = y_1^n, U_1^{i-1} = u_1^{i-1}) & \text{for the P1 domain,} \end{cases}$$

which is a sufficient statistic for U_i given (Y_1^N, U_1^{i-1}). Based on this, the transition probability satisfies

$$\tilde{W}_N^{(i)}(z|u_i) = \sum_{(y_1^N, u_1^{i-1}) \in S(z)} W_N^{(i)}\left((y_1^N, u_1^{i-1})|u_i\right),$$

where

$$S(z) = \begin{cases} \left\{ (y_1^N, u_1^{i-1}) \in \mathcal{Y}^N \times \{0,1\}^{i-1} \,\middle|\, z = \ln \frac{\mathbb{P}(Y_1^N = y_1^n, U_1^{i-1} = u_1^{i-1}) \,|\, U_i = 0)}{\mathbb{P}(Y_1^N = y_1^n, U_1^{i-1} = u_1^{i-1}) \,|\, U_i = 1)} \right\} & \text{for the LLR domain} \\ \left\{ (y_1^N, u_1^{i-1}) \in \mathcal{Y}^N \times \{0,1\}^{i-1} \,\middle|\, z = \mathbb{P}(U_i = 1 \,|\, Y_1^N = y_1^n, U_1^{i-1} = u_1^{i-1}) \right\} & \text{for the P1 domain} \end{cases}$$

In this case, the output space $\mathcal{Z}$ stays fixed for all N and is either $\mathcal{Z} = \mathbb{R}$ (for the LLR domain) or $\mathcal{Z} = [0,1]$ (for the P1 domain). Later, we will see how to recursively characterize $\tilde{W}_N^{(2i-1)}$ and $\tilde{W}_N^{(2i)}$ in terms of $\tilde{W}_{N/2}^{(i)}$.

3.3 The Erasure Channel

To understand the recursive formula for $\tilde{W}_N^{(i)}$, we focus first on the case of $W = \text{BEC}(\epsilon)$. This will allow us to leverage the simplicity of the erasure channel. Using this, it is easy to verify from Figure 1 that the soft estimate $\tilde{u}_1$ is an erasure unless both y_1 and y_2 are received correctly through their respective channels. This implies that $\tilde{W}_2^{(1)} = \text{BEC}(1 - (1-\epsilon)^2)$. Likewise, if the decoder is given knowledge u_1, then y_1 and y_2 act as independent observations of u_2 and we find that $\tilde{W}_2^{(2)} = \text{BEC}(\epsilon^2)$. One can easily verify that these results are generic and hold for message-passing decoding on the factor graph, MAP decoding on the factor graph, and the recursive successive cancellation decoder described in Algorithm 2.

To characterize the normalized virtual channels for larger N, we use induction based on decomposition shown in Figure 4. Let $\underline{u}' = (u_1', \ldots, u_{N/2}')$ denote the input bits to the upper $G_{N/2}$ block and $\underline{u}'' = (u_1'', \ldots, u_{N/2}'')$ denote the input bits to the lower $G_{N/2}$ block. In this figure, we see that input bits u_1, u_2 are mapped through the G_2 polar transform to $u_1' = u_1 \oplus u_2$ and $u_1'' = u_2$. The bits u_1' and u_1'' are effectively transmitted through two independent copies of the normalized virtual channel $\tilde{W}_{N/2}^{(1)}$.

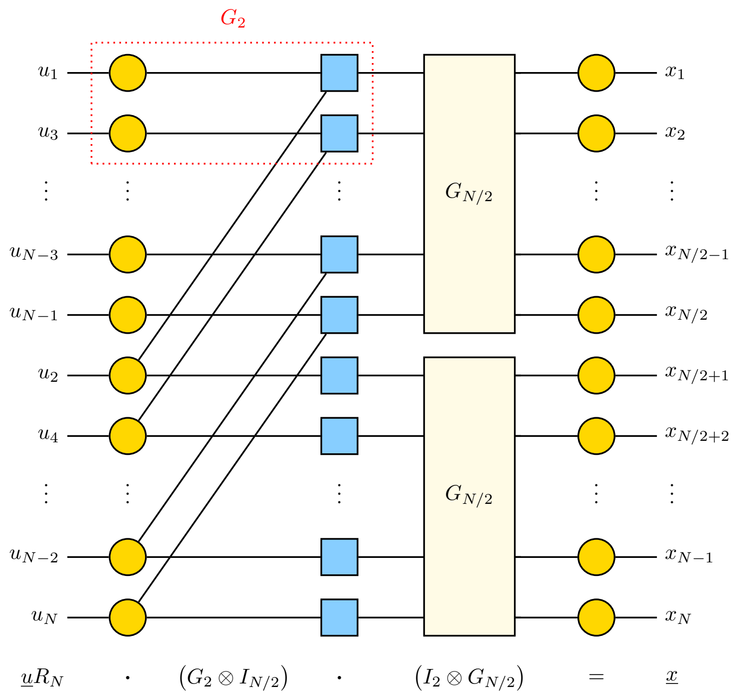

Figure 4: Factor graph associated with $G_N = R_N(G_2 \otimes I_{N/2})(I_2 \otimes G_{N/2})$.

Before making a hard decision, the successive-cancellation decoder combines the APP estimates $\tilde{u}'_1$ and $\tilde{u}''_1$ with the check-node operation to get $\tilde{u}_1 = \tilde{u}'_1 \boxast \tilde{u}''_1$. Thus, $\tilde{u}_1$ is an erasure unless both $\tilde{u}'_1$ and $\tilde{u}''_1$ both received correctly through their respective virtual channels. Therefore, we see that $\tilde{W}_N^{(1)}$ is an erasure channel if $\tilde{W}_{N/2}^{(1)}$ is an erasure channel and we define $\epsilon_N^{(i)}$ to be the erasure rate of the channel $\tilde{W}_N^{(i)}$. Moreover, we find that $\epsilon_N^{(1)} = 1 - (1 - \epsilon_{N/2}^{(1)})^2$. If u_1 is known by the decoder (e.g., if $\tilde{u}_1$ is not an erasure), then the APP estimate of u_2 is given by $\tilde{u}_2 = (u_1 \boxast \tilde{u}'_1) \circledast \tilde{u}''_1$. Thus, $\tilde{W}_N^{(2)}$ will be an erasure channel with erasure probability $\epsilon_N^{(2)} = (\epsilon_{N/2}^{(1)})^2$.

Now, assume that the bits u_1^{2i-2} are all known at the decoder. Figure 4 implies that the bits $u'_1, u'_2, \ldots, u'_{i-1}$ and $u''_1, u''_2, \ldots, u''_{i-1}$ will also be known at both $G_{N/2}$ decoders. Moreover, the bits u_{2i-1}, u_{2i} are mapped through the G_2 polar transform to $u'_i = u_{2i-1} \oplus u_{2i}$ and $u''_i = u_{2i}$. Since the bits $u'_1, u'_2, \ldots, u'_{i-1}$ and $u''_1, u''_2, \ldots, u''_{i-1}$ are known at their respective $G_{N/2}$ decoders, the bits u'_i and u''_i are effectively transmitted through two independent copies of normalized virtual channel $\tilde{W}_{N/2}^{(i)}$ (see Figure 4).

Similar to the previous arguments $\tilde{u}_{2i-1}$ is an erasure unless both $\tilde{u}'_i$ and $\tilde{u}''_i$ both received correctly through their respective virtual channels. Therefore, we see that $\tilde{W}_N^{(2i-1)}$ is an erasure channel if $\tilde{W}_{N/2}^{(i)}$ is an erasure channel and $\epsilon_N^{(2i-1)} = 1 - (1 - \epsilon_{N/2}^{(i)})^2$. If u_i is known by the decoder (e.g., if $\tilde{u}_i$ is not an erasure), then the APP estimate of u_{2i} is given by $\tilde{u}_{2i} = (u_i \boxast \tilde{u}'_i) \circledast \tilde{u}''_i$. Thus, $\tilde{W}_N^{(2i)}$ is an erasure channel with erasure probability $\epsilon_N^{(2i)} = (\epsilon_{N/2}^{(i)})^2$. Thus, we find that $\epsilon_1^{(1)} = \epsilon$ and

$$\begin{aligned} \epsilon_N^{(2i-1)} &= 1 - (1 - \epsilon_{N/2}^{(i)})^2 \\ \epsilon_N^{(2i)} &= (\epsilon_{N/2}^{(i)})^2. \end{aligned}$$

Algorithm 3 Density Evolution Analysis of Polar Code over BEC

```
function E = polar_bec(n,e)
% Compute effective-channel erasure rates for polar code of length N=2^n on BEC(e)
E = e;
for i=1:n
  % Interleave updates to keep in polar decoding order
  E = reshape([1-(1-E).*(1-E); E.*E],1,[]);
end
```

3.4 General BMS Channel

Consider a BMS channel W with input alphabet $\mathcal{X} = \{0,1\}$ and output alphabet $\mathcal{Y}$. Now, we can define two new BMS channels W^- and W^+. Let $W^- = W \boxcircle W$ be the BMS channel with output alphabet $\mathcal{Y} \times \mathcal{Y}$ and transition probabilities

$$W^-\left((y_1,y_2)|u_1\right) = \left(W \boxcircle W\right)\left((y_1,y_2)|u_1\right) \triangleq \frac{1}{2}\sum_{u_2\in\{0,1\}} W(y_1|u_1\oplus u_2)W(y_2|u_2).$$

This is the virtual channel associated with the decoding U_1 from (Y_1,Y_2) when (Y_1,Y_2) are observations of $(X_1,X_2) = (U_1 \oplus U_2, U_2)$ through two independent copies of W. Likewise, let W' be the BMS channel with output alphabet $\mathcal{Y}\times\mathcal{Y}\times\{0,1\}$ and transition probabilities

$$W'\left((y_1,y_2,u_1)|u_2\right) \triangleq \frac{1}{2}W(y_1|u_1\oplus u_2)W(y_2|u_2).$$

This is the virtual channel associated with the decoding U_2 from (Y_1,Y_2,U_1) when (Y_1,Y_2) are observations of $(X_1,X_2) = (U_1\oplus U_2, U_2)$ through two independent copies of W.

Exercise 1. Show that the binary-input channels W^- and W' are symmetric whenever W is symmetric.

Symmetry of the W channel also allows the output remapping $(y_1,y_2,u_1)\mapsto(y_1',y_2) = (u_1\oplus y_1, y_2)$ to transform W' into an equivalent channel with transition probabilities

$$\begin{aligned}\sum_{u_1\in\{0,1\}} W'\left((u_1\oplus y_1,y_2,u_1)|u_2\right) &= \sum_{u_1\in\{0,1\}}\frac{1}{2}W(u_1\oplus y_1|u_1\oplus u_2)W(y_2|u_2)\\ &= W(y_1|u_2)W(y_2|u_2).\end{aligned}$$

The resulting channel is equivalent to having two independent observations of the same input. Therefore, we define $W^+ = W \circledast W$ be the BMS channel with output alphabet $\mathcal{Y}\times\mathcal{Y}$ and transition probabilities

$$W^+\left((y_1,y_2)|u_2\right) = \left(W\circledast W\right)\left((y_1,y_2)|u_2\right) \triangleq W(y_1|u_2)W(y_2|u_2).$$

While the output alphabet of the channels $W^-(y|x)$ and $W^+(y|x)$ nominally consists of all pairs $y = (y_1,y_2)\in\mathcal{Y}\times\mathcal{Y}$, these channels can always be "normalized" to output a minimal sufficient statistic for the input. Similar to our previous discussion, we let $\tilde{W}(z|x)$ be transition probabilities of the channel $X\to Z$ defined by remapping the output of the $X\to Y$ channel (defined by $W(y|x)$) with

$$Z = \begin{cases} \ln\frac{W(Y|0)}{W(Y|1)} & \text{if LLR domain}\\ W(Y|1)/\left(\frac{1}{2}W(Y|0)+\frac{1}{2}W(Y|1)\right) & \text{if P1 domain.}\end{cases}$$

For general BMS channels, the recursive argument used above for erasure channels applies almost exactly. The only difference is that resulting channels are not characterized by a single parameter and the channel combining operations are related to general "density evolution" operations rather then BEC arguments. Thus, we find that $\tilde{W}_1^{(1)} = W$ and

$$\begin{aligned} W_N^{(2i-1)} &= W_{N/2}^{(i)-} = W_{N/2}^{(i)} \boxcircle W_{N/2}^{(i)}\\ W_N^{(2i)} &= W_{N/2}^{(i)+} = W_{N/2}^{(i)} \circledast W_{N/2}^{(i)}. \end{aligned} \tag{3}$$

In practice, it is much more computationally efficient to normalize these channels throughout the recursion. This leads to the observation that

$$\begin{aligned}\tilde{W}_N^{(2i-1)} &= W_{N/2}^{(i)} \tilde{\boxast} W_{N/2}^{(i)} = \tilde{W}_{N/2}^{(i)} \tilde{\boxast} \tilde{W}_{N/2}^{(i)} \\ \tilde{W}_N^{(2i)} &= W_{N/2}^{(i)} \tilde{\circledast} W_{N/2}^{(i)} = \tilde{W}_{N/2}^{(i)} \tilde{\circledast} \tilde{W}_{N/2}^{(i)},\end{aligned}$$

where $\tilde{\boxast}$ and $\tilde{\circledast}$ represent the $\boxast$ and $\circledast$ operations followed by channel normalization. The operators $\tilde{\boxast}$ and $\tilde{\circledast}$ are known as the bit and check node "density evolution" operators and were originally defined for the analysis of LDPC codes.

Once you have a decoder, the bit-error rates of these effective channels are easily estimated via Monte Carlo. To do this, one transmits a known vector and uses the decoder to estimate the input bits while passing genie-aided decisions for the output bits. This is why Algorithm 2 returns noisy hard decisions for frozen bits. For example, Algorithm 4 uses this method to characterize the effective channels of a polar code over the BSC

Algorithm 4 Monte Carlo Estimate of Polar Code over the BSC

```
function [biterrd] = polar_bsc(n,p,M)
% Send M blocks for Monte Carlo estimate of length N=2^n polar code on BSC(p)

% Setup parameters
N = 2^n;
f = zeros(1,N);
biterrd = zeros(1,N);

% Monte Carlo evaluation of error probability
for i=1:M
  % Transmit all-zero codeword through BSC(p)
  y = zeros(1,N)+p;
  y(rand(1,N)<p)=1-p;
  % Decode received vector using all-zero frozen vector
  [uhat,xhat] = polar_decode(y,f);
  biterrd = biterrd + uhat;
end
biterrd = biterrd/M;
```

3.5 The Design of Polar Codes

We have now seen how one can use recursive channel-evolution to evaluate the quality of each of the virtual channels. As described, this procedure uses "normalized" channels to prevent an increase in the output dimension at each stage. In particular, the code-design process corresponds to using this evolution procedure to evaluate the error probability of each virtual channel and then choosing the set $\mathcal{A}$ to contain the indices of channels with sufficiently low error probability. For the other channels, one simply transmits fixed dummy bits (e.g., zeros) that are known to the receiver in advance. Thus, these can be recovered by the receiver without error.

The successive cancellation decoder is successful as long as each of the virtual channels in $\mathcal{A}$ is decoded correctly. Let $(U_1, \ldots, U_N)$ be the message vector and $(\hat{U}_1, \ldots, \hat{U}_N)$ be the vector of hard decisions produced by successive cancellation decoder. Let

$$\delta_N^{(i)} = \mathbb{P}\left(\hat{U}_i \neq U_i \,|\, \hat{U}_1^{i-1} = U_1^{i-1}\right)$$

be the hard decision error probability of the virtual channel $W_N^{(i)}$. If we let the r.v.

$$I = \min\left\{i \in \{1, 2, \ldots, N\} \,|\, \hat{U}_i \neq U_i\right\}$$

denote the index of the first incorrectly decoded bit, then we can upper bound the probability of block error by

$$\begin{aligned}
P_B &= \mathbb{P}\left(I \in \{1,2,\ldots,N\}\right) \\
&\le \sum_{i=1}^{N} \mathbb{P}\left(I = i\right) \\
&\le \sum_{i\in\mathcal{A}} \mathbb{P}\left(\hat{U}_i \neq U_i \,|\, \hat{U}_1^{i-1} = U_1^{i-1}\right) \\
&\le \sum_{i\in\mathcal{A}} \delta_N^{(i)},
\end{aligned}$$

because $\hat{U}_i = U_i$ by definition for all $i \in \mathcal{A}^c$. Likewise, the bit error probability is upper bounded by

$$\begin{aligned}
P_b &\le \sum_{i=1}^{N} |\{j \in \mathcal{A} \,|\, j \ge i\}| \, \mathbb{P}\left(I = i\right) \\
&\le \sum_{i\in\mathcal{A}} |\{j \in \mathcal{A} \,|\, j \ge i\}| \, \mathbb{P}\left(\hat{U}_i \neq U_i \,|\, \hat{U}_1^{i-1} = U_1^{i-1}\right) \\
&\le \sum_{i\in\mathcal{A}} |\{j \in \mathcal{A} \,|\, j \ge i\}| \, \delta_N^{(i)}.
\end{aligned}$$

If we want to minimize these upper bounds and send $k = |\mathcal{A}|$ information bits, then we can simply choose $\mathcal{A}$ to contain the k indices with the smallest values of $\delta_N^{(i)}$.

Once the set of information bits has been chosen, it is easy to compute the generator and parity-check matrices of the code. For a matrix D and a subset $\mathcal{B} \subseteq \mathbb{N}$, let us define $D_{\mathcal{B}}$ (resp. $D^{\mathcal{B}}$) to be the submatrix of D formed by selecting only the columns (resp. rows) of D whose indices are in $\mathcal{B}$. If the frozen bits are all zero (i.e., $u_i = 0$ for all $i \notin \mathcal{A}$), then it follows that $\underline{x} = \underline{u}_{\mathcal{A}} G_N^{\mathcal{A}}$, where $G_N^{\mathcal{A}}$ is the $|\mathcal{A}| \times N$ generator matrix for the polar code. Similarly, a parity-check matrix for the code can be derived from 2 because $\underline{x} G_N = \underline{u} G_N G_N = \underline{u}$ implies that

$$[\underline{x} G_N]_{\mathcal{A}^c} = \underline{x}\,[G_N]_{\mathcal{A}^c} = \underline{u}_{\mathcal{A}^c} = \underline{0}.$$

Thus, a natural choice for the parity-check matrix H satisfies $H^T = [G_N]_{\mathcal{A}^c}$.

For the erasure channel BEC(ϵ), each effective channel is a BEC with erasure probability $\epsilon_N^{(i)} = 2\delta_N^{(i)}$ because the randomly breaking ties results in a error rate that is half of the erasure rate. These effective channels are characterized by Algorithm 3. For the BSC, the effective channels are characterized by Algorithm 4 using a Monte Carlo approach. The test program listed as Algorithm 6 combines the design, encoding, and decoding procedures of polar coding system.

Algorithm 5 Design a polar code based on effective-channel error rates

```
function f = polar_design(biterrd,d)
% Design a polar code based on effective-channel error rates

% Sort into increasing order and compute cumulative sum
[SE,order] = sort(biterrd);
CSE = cumsum(SE);

% Find best frozen bits
k = sum(double(CSE<d));
f = zeros(1,length(biterrd));
f(order(1:k)) = 1/2;
```

3.6 Achieving Capacity

The most exciting thing about polar codes is that they allow the deterministic construction of a code that achieves capacity on any BMS channel. To show this, Arıkan used an elegant martingale argument based on the fact that each G_2-butterfly step preserves mutual information. Later, this approach was simplified to avoid the explicit use of martingales [6]. We use the simplified approach below to show that the fraction of unpolarized channels converges to 0 as $n \to \infty$.

The following lemma is required for the argument below. A proof is provided at the end of the section.

Lemma 2. *For all $\delta \in \left[0, \frac{1}{2}\right]$ and any symmetric W satisfying $I(W) \in [\delta, 1-\delta]$, the quantity*

$$\Delta(W) \triangleq \frac{1}{2}\left(I(W^{+}) - I(W^{-})\right) \tag{4}$$

satisfies $\Delta(W)^2 \geq \kappa(\delta)$ where

$$\kappa(\delta) \triangleq \min_{h^{-1}(\delta) \leq p \leq h^{-1}(1-\delta)} \left(h\left(2p(1-p)\right) - h(p)\right)^2$$

and $h(p)$ is the binary entropy function. Also, $\kappa(\delta) > 0$ for $\delta \in \left(0, \frac{1}{2}\right]$.

Let the average (over all channels) of the mutual information after n steps be denoted by

$$\begin{aligned}
\mu_{n+1} &\triangleq \frac{1}{2^{n+1}} \sum_{i=1}^{2^{n+1}} I\left(W_{2^{n+1}}^{(i)}\right) \\
&\overset{(a)}{=} \frac{1}{2^n} \sum_{i=1}^{2^n} \left(\frac{1}{2} I\left(W_{2^{n+1}}^{(2i-1)}\right) + \frac{1}{2} I\left(W_{2^{n+1}}^{(2i)}\right)\right) \\
&\overset{(b)}{=} \frac{1}{2^n} \sum_{i=1}^{2^n} \left(\frac{1}{2} I\left(W_{2^n}^{(i)-}\right) + \frac{1}{2} I\left(W_{2^n}^{(i)+}\right)\right) \\
&\overset{(c)}{=} \frac{1}{2^n} \sum_{i=1}^{2^n} I\left(W_{2^n}^{(i)}\right) = \mu_n,
\end{aligned}$$

where (a) breaks the sum into odd/even terms, (b) follows from (3), and (c) follows from

$$I(W) = \frac{1}{2}\left(I(W^{+}) + I(W^{-})\right).$$

By induction, $\mu_n = \mu_0 = I(W)$.

Combining with (4), we observe that

$$I(W)^2 + \Delta(W)^2 = \frac{1}{2}\left(I(W^{+})^2 + I(W^{-})^2\right). \tag{5}$$

Let the average (over all channels) of the squared mutual-information after n steps be denoted by

$$\begin{aligned}
\nu_{n+1} &\triangleq \frac{1}{2^{n+1}} \sum_{i=1}^{2^{n+1}} I\left(W_{2^{n+1}}^{(i)}\right)^2 \\
&\overset{(a)}{=} \frac{1}{2^n} \sum_{i=1}^{2^n} \left(\frac{1}{2} I\left(W_{2^{n+1}}^{(2i-1)}\right)^2 + \frac{1}{2} I\left(W_{2^{n+1}}^{(2i)}\right)^2\right) \\
&\overset{(b)}{=} \frac{1}{2^n} \sum_{i=1}^{2^n} \left(\frac{1}{2} I\left(W_{2^n}^{(i)+}\right)^2 + \frac{1}{2} I\left(W_{2^n}^{(i)-}\right)^2\right)
\end{aligned}$$

$$\overset{(c)}{=} \frac{1}{2^n} \sum_{i=1}^{2^n} \left(I\left(W_{2^n}^{(i)}\right)^2 + \Delta\left(W_{2^n}^{(i)}\right)^2 \right)$$

$$= \nu_n + \sum_{i=1}^{2^n} \Delta\left(W_{2^n}^{(i)}\right)^2, \tag{6}$$

where (a) breaks the sum into odd/even terms, (b) follows from (3), and (c) follows from (5). From this, we see that $\nu_{n+1} \geq \nu_n$. Since $\nu_n \in [0,1]$ by definition, this implies that ν_n converges to a limit and, hence, that $\nu_{n+1} - \nu_n \to 0$.

Now, we define the fraction of δ-unpolarized channels after n steps to be

$$\theta_n(\delta) \triangleq \frac{1}{2^n} \left| \left\{ i \in [2^n] \,\middle|\, I\left(W_{2^n}^{(i)}\right) \in [\delta, 1-\delta] \right\} \right|.$$

Using Lemma 2, it follows that

$$\sum_{i=1}^{2^n} \Delta\left(W_{2^n}^{(i)}\right)^2 \geq \theta_n(\delta)\kappa(\delta).$$

Combining with (6), we see that

$$0 \leq \theta_n(\delta) \leq \frac{\nu_{n+1} - \nu_n}{\kappa(\delta)}.$$

Since $\nu_{n+1} - \nu_n \to 0$, this implies that $\theta_n(\delta) \to 0$ for all $\delta \in (0, 1/2)$. In addition, standard results from real analysis imply the existence of a sequence $\delta_n \to 0$ such that $\theta_n(\delta_n) \to 0$. Thus, the fraction of unpolarized channels converges to 0 as $n \to \infty$. Since $\mu_n = I(W)$ for all n, this also implies that a fraction $I(W)$ of the virtual channels $W_N^{(i)}$ become perfect (i.e., $I(W) \to 1$) and a fraction $1 - I(W)$ of the channels become useless (i.e., $I(W) \to 0$).

Due to the sequential nature of successive decoding, Arıkan also uses a more sophisticated bound to show that successive decoding achieves a probability of error that can be made arbitrarily small for any rate less than capacity.

Proof of Lemma 1. If X_1, X_2 are binary and $I(W) = I(X_1; Y_1) = I(X_2; Y_2) = 1 - h(p)$, then Ms. Gerber's Lemma shows that

$$I(W^-) = 1 - H(X_1 \oplus X_2 | Y_1, Y_2) \leq 1 - h\left(2p(1-p)\right).$$

Since $I(W^+) = 2I(W) - I(W^-)$, it follows that

$$\begin{aligned} \Delta(W) &= \frac{1}{2}\left(I(W^+) - I(W^-)\right) \\ &= \frac{1}{2}\left(2I(W) - I(W^-) - I(W^-)\right) \\ &= I(W) - I(W^-) \\ &\geq 1 - h(p) - \left(1 - h\left(2p(1-p)\right)\right) \\ &= h\left(2p(1-p)\right) - h(p). \end{aligned}$$

Since $0 < p < 2p(1-p) \leq \frac{1}{2}$ for $p \in \left(0, \frac{1}{2}\right)$ and $h(p)$ is strictly increasing on the same set, it follows that $\Delta(W) > 0$ as long as $p \in \left(0, \frac{1}{2}\right)$. For $I(W) \in [\delta, 1-\delta]$, one can minimize this bound over this range to see that $\Delta(W)^2 \geq \kappa(\delta)$. Finally, we note that $\kappa(\delta) > 0$ for $\delta \in \left(0, \frac{1}{2}\right]$ because $\kappa(\delta) > 0$ as long as $0 < h^{-1}(\delta)$ and $h^{-1}(1-\delta) < \frac{1}{2}$. □

A Information Theory Primer

Information theory is a very broad area and there are many excellent textbooks on the subject (e.g., [7]). This section is meant only to introduce basic concepts important for understanding polar codes.

A.1 Entropy and Mutual Information

Definition 3. The **entropy** (in bits) of a discrete random variable X with probability distribution $p(x)$ is denoted

$$H(X) \triangleq \sum_{x\in\mathcal{X}} p(x)\log_2 \frac{1}{p(x)} = \mathbb{E}\left[\log_2 \frac{1}{p(X)}\right],$$

where $0\log_2 0 = 0$ by continuity. The notation $H(p)$ is used to denote $H(X)$ when $X \sim p(x)$. When there is no ambiguity, H will be used instead of $H(X)$. The unit of entropy is determined by the base of the logarithm with base-2 resulting in "bits" and the natural log (i.e., base-e) resulting in "nats".

Remark 4. Roughly speaking, the entropy $H(X)$ measures the uncertainty in the random variable X. From a compression point of view, it equals the minimum average bit rate (in bits per symbol) to which one can compress long sequences $X_1, X_2, \ldots$ drawn i.i.d. according to $p(x)$. It also equals the exponential growth rate, for i.i.d. sequences, of the smallest set that contains almost all the probability. For any $\delta \in (0,1)$, one can show that

$$H(X) = \lim_{N\to\infty} \min_{S\in\left\{T\subseteq\mathcal{X}^N \,\middle|\, \sum_{(x_1,\ldots,x_N)\in T}\prod_{i=1}^N p(x_i)>1-\delta\right\}} \frac{1}{N}\log_2|S|.$$

Definition 5. The **joint entropy** (in bits) of a pair of r.v. $(X,Y) \sim p_{X,Y}(x,y)$ is denoted

$$H(X,Y) \triangleq \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x,y)\log_2 \frac{1}{p_{X,Y}(x,y)} = \mathbb{E}\left[\log_2 \frac{1}{p_{X,Y}(X,Y)}\right].$$

Notice that this is identical to $H(Z)$ with $Z = (X,Y)$.

Definition 6. For a pair of r.v. $(X,Y) \sim p_{X,Y}(x,y)$, the **conditional entropy** (in bits) of Y given X is denoted

$$H(Y|X) \triangleq \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{X,Y}(x,y)\log_2 \frac{1}{p_{Y|X}(y|x)} = \mathbb{E}\left[\log_2 \frac{1}{p_{Y|X}(Y|X)}\right].$$

Notice that this equals entropy of the conditional distribution $p_{Y|X}(y|x)$ averaged over x.

Definition 7. For a pair of r.v. $(X,Y) \sim p_{X,Y}(x,y)$, the **mutual information** (in bits) between X and Y is denoted

$$I(X;Y) \triangleq \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{X,Y}(x,y)\log_2 \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)} = \mathbb{E}\left[\log_2 \frac{p_{X,Y}(X,Y)}{p_X(X)p_Y(Y)}\right].$$

Lemma 8. *Basic properties of joint entropy and mutual information:*

1. *(chain rule of entropy)* $H(X,Y) = H(X) + H(Y|X)$. *If* X *and* Y *are independent,* $H(X,Y) = H(X) + H(Y)$.
   Proof: Take the expectation of $\log_2 \frac{1}{p_{X,Y}(X,Y)} = \log_2 \frac{1}{p_X(X)} + \log_2 \frac{1}{p_{Y|X}(Y|X)}$ *and note that* $P_{Y|X}(y|x) = p_Y(y)$ *for all* x,y *if* X *and* Y *are independent.*

2. *(mutual information) The mutual information satisfies* $I(X;Y) = I(Y;X)$ *and*

$$I(X;Y) = H(X) + H(Y) - H(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

   Proof: Take the expectation of $\log_2 \frac{p_{X,Y}(X,Y)}{p_X(X)p_Y(Y)} = \log_2 \frac{1}{p_X(X)} + \log_2 \frac{1}{p_Y(Y)} - \log_2 \frac{1}{p_{X,Y}(X,Y)}$ *and apply the chain rule as needed. Also, symmetry follows from swapping* X,Y *and* x,y *in the sum because* $p_{X,Y}(x,y) = p_{Y,X}(y,x)$.

Example 9. Let $\mathcal{X} = \mathcal{Y} = \{0,1\}$ and $p_{X,Y}(x,y) = \rho/2$ if $x \neq y$ and $p_{X,Y}(x,y) = (1-\rho)/2$ if $x = y$. Since $p_X(x) = p_Y(y) = \frac{1}{2}$, it follows that $H(X) = 1$ and $H(Y) = 1$. Since $p_{Y|X}(y|x) \in \{\rho, 1-\rho\}$, it follows that $H(Y|X) = h(\rho)$. Thus, we have $I(X;Y) = H(Y) - H(Y|X) = 1 - h(\rho)$. The conditional distribution $p_{Y|X}$ called the **binary symmetric channel** with error probability ρ and denoted by BSC(ρ).

The previous definitions of entropy and mutual information all extend naturally to any finite number of random variables by treating multiple random variables as a single random vector. However, there are some new concepts that can only be defined in terms of 3 random variables. Let X, Y, and Z be random variables with joint distribution $p_{X,Y,Z}(x,y,z)$.

Definition 10. For three r.v. $(X,Y,Z) \sim p_{X,Y,Z}(x,y,z)$ defined on $\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}$, the **conditional mutual information** (in bits) between X and Y given Z is denoted

$$I(X;Y|Z) \triangleq \sum_{(x,y,z)\in\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}} p_{X,Y,Z}(x,y,z)\log_2 \frac{p_{X,Y|Z}(x,y|z)}{p_{X|Z}(x|z)p_{Y|Z}(y|z)} = \mathbb{E}\left[\log_2 \frac{p_{X,Y|Z}(X,Y|Z)}{p_{X|Z}(X|Z)p_{Y|Z}(Y|Z)}\right].$$

From this, we see that $I(X;Y|Z) = H(X|Z) + H(Y|Z) - H(X,Y|Z)$. Thus, the conditioning is simply inherited by each entropy in the standard decomposition.

Lemma 11. *The chain rule of mutual information states that* $I(X;Y,Z) = I(X;Y) + I(X;Z|Y)$.

Proof. This follows from the expectation of the decomposition

$$\begin{aligned}\log_2 \frac{p_{X,Y,Z}(X,Y,Z)}{p_X(X)p_{Y,Z}(Y,Z)} &= \log_2 \frac{p_{X,Y}(X,Y)p_{Z|X,Y}(Z|X,Y)}{p_X(X)p_Y(Y)p_{Z|Y}(Z|Y)} \\ &= \log_2 \frac{p_{X,Y}(X,Y)}{p_X(X)p_Y(Y)} + \log_2 \frac{p_{X,Z|Y}(X,Z|Y)}{p_{Z|Y}(Z|Y)p_{X|Y}(X|Y)}.\end{aligned}$$

□

A.2 Channel Coding

In engineering, one often wants to communicate information across an unreliable medium. For example, think of a system that modulates the current in a wire (by adjusting the voltage at one end) and measures the current at the other end. Due to thermal fluctuations, the difference between the modulated current at the measured current will always contain some randomness. One can analyze this situation by first discretizing time and then defining a simple mathematical model.

Definition 12. A **discrete memoryless channel** (DMC) is defined by a finite input alphabet $\mathcal{X}$, a finite output alphabet $\mathcal{Y}$, and a conditional probability distribution $W(y|x)$. For $N \in \mathbb{N}$ channel uses, let the channel input vector be a random vector $\underline{X} = (X_1, \ldots X_N) \in \mathcal{X}^N$. Then, the channel output vector is a random vector $\underline{Y} = (Y_1, \ldots, Y_N) \in \mathcal{Y}^N$ where

$$W_N(\underline{y}|\underline{x}) \triangleq \mathbb{P}\left(\underline{Y} = \underline{y}\middle|\underline{X} = \underline{x}\right) = \prod_{t=1}^{N} W(y_i|x_i).$$

Definition 13. A DMC with binary inputs is called **symmetric** if there is a permutation $\pi : \mathcal{Y} \to \mathcal{Y}$ satisfying $W(\pi(y)|1) = W(y|0)$ and $\pi(\pi(y)) = y$ for all $y \in \mathcal{Y}$.

Example 14. For example, the **binary symmetric channel** (BSC) with error probability ρ has $\mathcal{X} = \mathcal{Y} = \{0,1\}$ and is defined by

$$W(y|x) = (1-\rho)\mathbb{I}(x = y) + \rho\mathbb{I}(x \neq y).$$

Example 15. For example, the **binary erasure channel** (BEC) with erasure probability ϵ has $\mathcal{X} = \{0,1\}$, $\mathcal{Y} = \{0,1,*\}$, and is defined by

$$W(y|x) = \epsilon\mathbb{I}(y = *) + (1-\epsilon)\mathbb{I}(y = x).$$

Channel coding is the process of improving performance by adding redundancy (e.g., by encoding an K bit message into $N > K$ bits).

Definition 16. For a code/decoder pair, the **block error probability** $\mathrm{P_B}$ is the probability that the decoder does not map the received sequence to the transmitted codeword. A code rate R is **achievable** if there exists a sequence of encoder/decoder pairs with rate $R_N \to R$ and block error rate $\mathrm{P}_{\mathrm{B},N} \to 0$. The **channel capacity** is the supremum of all achievable code rates.

Remark 17. One can get a qualitative feel for achievable rates via the following argument. The key is that, for i.i.d. sequences $(X_1, \dots, X_N)$ with large N, the probability distribution essentially becomes uniform over a set of $2^{NH(X)}$ "typical" sequences. Thus, for $(X, Y) \sim p(x)W(y|x)$, the i.i.d. sequence $((X_1, Y_1), \dots, (X_N, Y_N))$ takes one of $2^{NH(X,Y)}$ different typical values with essentially uniform probability. If we ignore the X values, then the number of $(Y_1, \dots, Y_N)$ typical sequences is roughly $2^{NH(Y)}$. If we fix the $(X_1, \dots, X_N)$ sequence to a typical value $(x_1, \dots, x_N)$, then the number of $((x_1, Y_1), \dots, (x_N, Y_N))$ typical sequences is roughly $2^{NH(Y|X)}$. This last set of sequences can be seen as the likely set of output sequences if $\underline{x}$ is transmitted. Thus, if the likely output sets of each codeword fill the space but do not overlap, then we get $2^{NR}2^{NH(Y|X)} = 2^{NH(Y)}$ or $R = H(Y) - H(Y|X) = I(X;Y)$.

Theorem 18 (Channel Coding Theorem). *For a DMC, the channel capacity is given by*

$$C \triangleq \max_{p(x)} I(X;Y) = \max_{p(x)} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p(x)W(y|x)\log_2 \frac{W(y|x)}{\sum_{x'} p(x')W(y|x')}.$$

Thus, for any $R \le C$, there exists a sequence of encoder/decoder pairs such that $R_N \to R$ and $\mathrm{P}_{\mathrm{B},N} \to 0$. Conversely, if a sequence of encoder/decoder pairs satisfies $R_N \to R$ and $\mathrm{P}_{\mathrm{B},N} \to 0$, then $R \le C$.

Remark 19. For a symmetric DMC, the capacity C is achieved by the uniform input distribution $p(x) = 1/|\mathcal{X}|$.

References

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inform. Theory*, vol. 55, pp. 3051–3073, July 2009.

[2] E. Arikan, "An upper bound on the cutoff rate of sequential decoding," *IEEE Trans. Inform. Theory*, vol. 34, pp. 55–63, Jan. 1988.

[3] S. H. Hassani, K. Alishahi, and R. Urbanke, "Finite-length scaling for polar codes," *IEEE Trans. Inform. Theory*, vol. 60, no. 10, pp. 5875–5898, 2014.

[4] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inform. Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[5] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins University Press, 3rd ed., 2012.

[6] M. Alsan and E. Telatar, "A simple proof of polarization and polarization for non-stationary memoryless channels," *IEEE Trans. Inform. Theory*, vol. 62, no. 9, pp. 4873–4878, 2016.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley Series in Telecommunications, Wiley, 2nd. ed., 2006.

Algorithm 6 Main Test Program for Polar Coding System

```
% Setup code parameters
n = 12; N = 2^n;
e = 0.5; p = 0.11;
d = 0.1; bec = 1;

% Compute the quality of all effective channels
if (bec)
  [biterrd] = polar_bec(n,e);
else
  [biterrd] = polar_bsc(n,p,1000);
end

% Design polar code
f = polar_design(biterrd,d);
A = (f==1/2);
k = sum(A);
rate = k/N

% Run a few sims to compare with union bound
M = 10;
biterr = zeros(1,M);
for i=1:M
  % Set frozen bits, add random data, and encode
  u = f;
  u(A) = rand(1,k)<0.5;
  x = polar_transform(u);

  % Transmit
  if (bec)
    y = x;
    y(rand(1,N)<e)=1/2;
  else
    y = zeros(1,N)+p;
    y(x==1) = 1-y(x==1);
    err = rand(1,N)<p;
    y(err) = 1-y(err);
  end

  % Decode and compute error rate for info bits
  [uhat,xhat] = polar_decode(y,f);
  biterr(i) = mean(uhat(A)~=u(A));
end

% Display average bit and block error rate
mean(biterr)
mean(biterr>0)
```