Report: Lossless Coding with Polar Codes

Harm Cronie and Satish Korada

Abstract

The abstract goes here.

I. Introduction

Polar codes introduced by Arikan in [] are the first family of error-correcting codes of which it can be proved that they achieve capacity on any binary-memoryless symmetric channel with a decoding complexity that scales logarithmically in the blocklength. In [] it is shown that polar codes are also optimal for lossless and lossy source coding. However, in [] the argument that polar codes are optimal in a lossless setting is based on channel coding arguments. In this report we derive and analyze lossless polar source encoders and show that unlike in the channel coding setting one can devise polar encoders that perform very well in a practical finite length setting.

II. System model and notation

We denote the binary entropy function by $h(p)$. The indicator function is denoted by $\mathbf{1}\left(\right) A)$

III. Polar Lossless Source Codes

Let U denote a random variable corresponding to a binary memoryless source. We assume that $\Pr[U=0]=p$ and the entropy of the source $H(U)$ is given by

$$H(U) = -\left(p\log_2 p + (1-p)\log_2(1-p)\right). \tag{1}$$

A. Polarization encoding for two source symbols

The two main ingredients of polar source coding are *sequential encoding* and *polarization*. A motivation behind sequential encoding is the chain rule of entropy which basically states that nothing is lost when we use a sequential encoder. The idea of polar coding for source coding is similar as that of polar coding for channel coding. For a given sequence of source symbols we create another sequence of sources symbols with the property that for the latter sequence under sequential encoding, the entropies behave in a beneficial way for encoding.

1) Polarization of entropy: Instead of encoding a sequence of source symbols directly, we consider an invertible linear transformation of these source symbols. Encoding of the result of this transformation will become very easy for large block lengths. We illustrate the main concepts of polarization encoding for the case of two source symbols and show how to construct an encoder and decoder.

Let U_1 and U_2 be two random variables corresponding to two outputs of the discrete memoryless source U. Furthermore, let $\mathbf{G}_2$ be the following linear transformation

$$\mathbf{G}_2 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \tag{2}$$

Now, let V_1 and V_2 be two random variables defined as

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \mathbf{G}_2 \begin{bmatrix} U_1 \\ U_2 \end{bmatrix}. \tag{3}$$

In terms of entropies we have the following relation

$$H(U_1,U_2) = H(U_1)+H(U_2) = H(V_1,V_2) = H(V_1)+H(V_1|V_2), \tag{4}$$

where the first equality follows from the independence of U_1 and U_2, the second from the invertibility of $\mathbf{G}_2$ and the last from the chain rule of entropy. Instead of encoding U_1 and U_2 we consider encoding V_1 and V_2 in a sequential fashion. The encoder is based on the following lemma.

Lemma 1: Application of $\mathbf{G}_2$ polarizes

$$H_1(V_1) \geq \frac{H(U_1)+H(U_2)}{2} \geq H(V_2|V_1). \tag{5}$$

Furthermore, if $H(U) = \frac{H(U_1)+H(U_2)}{2} \in [\delta, 1-\delta]$ for $\delta > 0$, the two inequalities are strict.

Proof: First note $H(V_1, V_2) = H(U_1, U_2)$ since $\mathbf{G}_2$ is invertible. Furthermore, by the chain rule of entropy we have

$$H(V_1, V_2) = H(V_1) + H(V_2|V_1). \tag{6}$$

Furthermore, $\Pr[V_1 = v_1]$ and $\Pr[V_2 = v_2|V_1 = v_1]$ are given by

$$p(v_1) = \sum_{u_1,u_2} p(v_1, u_1, u_2) = \sum_{u_1,u_2} p(v_1|u_1, u_2)p(u_1, u_2) = \sum_{u1,u2} \mathbf{1}\left(v_1 = u_1 + u_2\right) p(u_1, u_2) \tag{7}$$

$$p(v_2|v_1) = \sum_{u_1,u_2} p(v_2, u_1, u_2|v_1) = \sum_{u_1,u_2} \frac{p(v_1, v_2|u_1, u_2)p(u_1, u_2)}{p(v_1)} = \sum_{u_1,u_2} \mathbf{1}\left(v_1 = u_1 + u_2\right) \mathbf{1}\left(v_2 = u_2\right) \frac{p(u_1, u_2)}{p(v_1)} \tag{8}$$

A direct computation gives

$$p(v_1) = \begin{cases} p^2 + (1-p)^2 & v_1 = 0 \\ 2p(1-p) & v_1 = 1 \end{cases} \tag{9}$$

$$p(v_2|v_1) = \begin{cases} \frac{p^2}{p^2+(1-p)^2} & v_1 = 0, v_2 = 0 \\ \frac{(1-p)^2}{p^2+(1-p)^2} & v_1 = 0, v_2 = 1 \\ \frac{1}{2} & v_1 = 1, v_2 = 0 \\ \frac{1}{2} & v_1 = 1, v_2 = 1 \end{cases} \tag{10}$$

Now, note that $\frac{H(U_1)+H(U_2)}{2} = h(p)$ and $H(V_1) = h(p^2 + (1-p)^2)$. To show that $H(V_1) \geq \frac{H(U_1)+H(U_2)}{2}$ it suffices to show that $h(p^2 + (1-p)^2) \geq h(p)$. This latter statement follows from the convexity of $h(p)$. The second inequality follows from (5). Finally, note that $h(p) = h(p^2 + (1-p)^2)$ if and only if $p \in \{0, 1, \frac{1}{2}\}$ and we have equality only for these values of p. ■

2) A succesive source encoder: Instead of encoding U_1 and U_2 we will encode V_1 and V_2 in a succesive fashion. Moreover, we make use of the fact that $H(V_1) \geq H(V_2|V_1)$. The encoder takes as its input a realization u_1, u_2 of U_1 and U_2 and statistics of the source U which in this case is given by p. The source encoder performs the following steps to encode.

Algorithm 1: Source encoder for U_1 and U_2

1) The encoder computes $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \mathbf{G}_2 \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$.
2) The entropy of V_1 is high and the encoder will always output the value of v_1 it has computed in the previous step.
3) The encoder computes $p(v_2|v_1)$ where it uses the value of v_1 computed in the first step and estimates v_2 as

$$\hat{v}_2 = \text{round}\left(1 - p(v_2|v_1)\right) \tag{11}$$

4) The encoder compares $\hat{v_2}$ to v_2 and outputs v_2 in case they are not equal. Otherwise the only output remains v_1.

A schematic diagram of the encoder is shown in Figure **??**. TODO. We should analyze this encoder and show that it works but it has shortcomings. Then we can motivate a successive application of $\mathbf{G}_2$ to obtain polarization such that these short comings disappear. Furthermore, Lemma 1 should be extended to densities of p which is useful to prove polarization results.

B. Polarization encoding of a sequence of source symbols

To extend the idea to a longer sequence of source symbols, we use a recursively defined map as in []. The recursive nature of this map facilitates us as follows. First, we it allows for the computation of the probabilities required for the encoder and decoder. Second, it allows to prove polarization results.

1) Recursive polarization maps: Consider a sequence of $n = 2^m$ source random variables $U_1, \ldots, U_n$. We apply the following transformation to obtain a sequence of random variables $V_1, \ldots, V_n$.

$$\begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} = \mathbf{G}_2^{\otimes m} \begin{bmatrix} U_1 \\ \vdots \\ U_n \end{bmatrix}. \tag{12}$$

Note that $\mathbf{G}_2^{\otimes m} = \left(\mathbf{G}_2^{\otimes m}\right)^{-1}$ and we can factor the application of $\mathbf{G}_2^{\otimes m}$ as follows.

$$\mathbf{G}_2^{\otimes m} = \prod_{i=1}^{m} \left(\mathbf{I}_{2^{i-1}} \otimes \mathbf{G}_2 \otimes \mathbf{I}_{2^{m-i}}\right) \tag{13}$$

2) Encoding algorithm: We construct a similar encoder for n source symbols as we have done for two source symbols. We use a sequential encoder and by the chain rule of entropy we have

$$H(V_1,\ldots,V_n)=\sum_{i=1}^{n}H(V_i|V_{i-1},\ldots,V_1). \tag{14}$$

For now we assume that the application of (12) *polarizes* the entropy terms $H(V_i|V_{i-1},\ldots,V_1)$. We assume that we have a set of indices I for which the entropy terms are close to 1. By I^c we denote the set of indices for which the entropies are not close to 1. Now we can define an algorithm for encoding a sequence of n source symbols.

Algorithm 2: Encoding algorithm: The encoding algorithms takes as an input a source realization $\mathbf{u}$ of size $n=2^m$ and the probabilistic model of the source which is defined by p. Furthermore, we assume that the encoder has knowledge of I and I^c.

1) The encoder applies $\mathbf{G}_2^{\otimes m}$ to $\mathbf{u}$ to obtain $\mathbf{v}=\mathbf{G}_2^{\otimes m}\mathbf{u}$.
2) The encoder constructs a vector $\mathbf{c}$ consisting of the v_i for which $i\in I$.
3) For $i\in I^c$ the encoder computes $p(v_i|v_1\ldots,v_{i-1})$ and computes an estimate of v_i as

$$\hat{v}_i=\text{round}\left(1-p(v_i|v_1,\ldots,v_i)\right).$$

   In case $\hat{v}_i\neq v_i$, the encoder stores the index i in a set E.
4) The output of the encoder consists of $\mathbf{c}$ and E.

In a similar way we can construct a decoding algorithm. We assume that the decoder has knowledge of I and I^c.

Algorithm 3: Decoding algorithm: The decoder takes as an input $\mathbf{c}$, E and the probabilistic model of the source which is defined by a p.

1) For $i\in I$ the decoder sets v_i to the corresponding value in $\mathbf{c}$
2) For each $i\in I^c$ the decoder computes $p(v_i|v_1\ldots,v_{i-1})$. In case $i\neq E$ the decoder sets v_i to

$$v_i=\text{round}\left(1-p(v_i|v_1,\ldots,v_i)\right).$$

   Otherwise the decoder sets v_i to

$$v_i=\text{round}\left(p(v_i|v_1,\ldots,v_i)\right).$$

- From a practical point of view it is of importance how to compute the probabilites and entropies. This can be done in $\log(n)$ operations per source bit due to the structure of the transformations.
- This algorithm and some variants we can analyze and we can show that they are asymptotically optimal. For finite lengths we can compute the size of I required and its variance etc and how to choose k.

3) Computation of $p(v_i|v_{i-1},\ldots,v_1)$ and $H(V_i|V_{i-1},\ldots,V_1)$: For the encoder and decoder to be feasible it is of importance that the quantities involved can be computed efficiently. For the encoder and decoder described in this section the polarization map and the sequence of probabilities $p(v_i|v_{i-1},\ldots,v_1)$ have to be computed.

We will use an alternative description of the application of $\mathbf{G}_2^{\otimes m}$ which will be convenient later. For this purpose we introduce some additional notation. Let $\mathbf{V}^{(i)}$ denote a vector random variable indexed by an index i. We denote by $\mathbf{V}_{l,j}^{(i)}$ a vector random variable which is defined as follows. $\mathbf{V}^{(i)}$ is implicitly divided into 2^l equal parts and $\mathbf{V}_{l,j}^{(i)}$ contains its $j+1$-th part.

Now, let $\mathbf{V}^{(0)}$ be defined as

$$\mathbf{V}^{(0)}=[U_1,\ldots,U_n]^T, \tag{15}$$

and define

$$\mathbf{V}_{1,0}^{(1)}=\mathbf{V}_{1,0}^{(0)}+\mathbf{V}_{1,1}^{(0)} \tag{16}$$

$$\mathbf{V}_{1,1}^{(1)}=\mathbf{V}_{1,1}^{(0)}. \tag{17}$$

These two equations define $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(1)}$ is equal to the first term corresponding to $i=1$ in (13) applied to $\mathbf{V}^{(0)}$. For the other terms we can recursively define for $i\leq m-1$

$$\begin{array}{ll}\mathbf{V}_{i+1,j}^{(i+1)}=\mathbf{V}_{i+1,j}^{(i)}+\mathbf{V}_{i+1,j+1}^{(i)} & \text{for } j \text{ even}\\ \mathbf{V}_{i+1,j}^{(i+1)}=\mathbf{V}_{i+1,j}^{(i)} & \text{for } j \text{ even}\end{array} \tag{18}$$

Now, we have recursively defined a sequence of $m+1$ random vectors $\mathbf{V}^{(0)},\ldots,\mathbf{V}^{(m)}$ where $\mathbf{V}_j^{(m)}$ is equal to V_{j+1}. A schematic diagram of this recursion is given in Figure 1. This figure can be seen as the trellis defining the relation between $U_1,\ldots,U_n$ and $V_1,\ldots,V_n$.

Lemma 2: Elements of vectors are independent! This would allow us to drop conditioning.

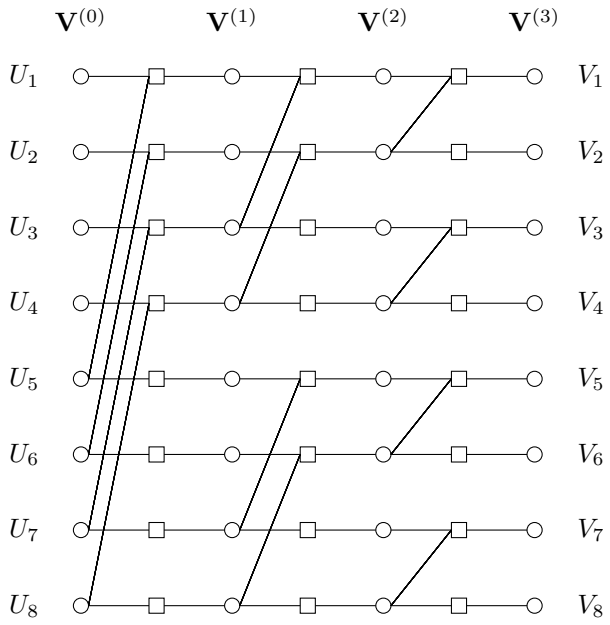

Fig. 1: Example of a polar construction for $n = 8$.

We are interested in constructing a similar encoder as we have constructed for two source symbols. The idea is to apply the transformation to a sequence of n source symbols and sequentially compute the sequence of probabilities $p(v_1), p(v_2|v_1), \ldots, p(v_n|v_{n-1},$ while making use of the actual realization of $V_1, \ldots, V_n$.

$$p\left(\mathbf{V}_{i+1,j}^{(i+1)} = v|\mathbf{V}_{i+1,0:j-1}^{(i+1)}\right) = p\left(\mathbf{V}_{i+1,j}^{(i)} + \mathbf{V}_{i+1,j+1}^{(i)} = v|\mathbf{V}_{i+1,0:j-1}^{(i)}\right) \quad j \text{ even} \tag{19}$$

$$p\left(\mathbf{V}_{i+1,j}^{(i+1)} = v|\mathbf{V}_{i+1,0:j-1}^{(i+1)}\right) = p\left(\mathbf{V}_{i+1,j-1}^{(i)} + \mathbf{v}_{i+1,j-1}^{(i+1)} = v \cup \mathbf{V}_{i+1,j}^{(i)} = v|\mathbf{V}_{i+1,0:j-1}^{(i)}\right) \quad j \text{ odd} \tag{20}$$

By the chain rule of entropy we have

$$H(V_1, \ldots, V_n) = \sum_{i=1}^{n} H(V_i|V_{i-1}, \ldots, V_1). \tag{21}$$

We construct an encoder based on the fact that some of the entropies approach 0 and some of the entropies approach 1. The V_i for which the entropies approach 0 do not ha

4) Asymptotic limit of polarization:

C. Asymptotic polarization

IV. Analysis of Algorithms for Encoding

A. An algorithm for encoding

- density evolution based, ordering on indices, one can select next indices to encode
- based on source realization

B. Complexity

V. Application

VI. Todo

- Scheme has differences with standard parity check approach where we would fix parity check matrix. Here we do actual BP decoding for encoding. Furthermore, we do on the fly comparison with inverseof encoded sequence. In source coding we also have estimation of probability model, etc.
- No need to compute good indices, especially when we use the RNG approach.
- Comparison with inverse of source sequence.
- Scheme incorporates the actual source as a source of randomness.
- Practical w.r.t. complexity

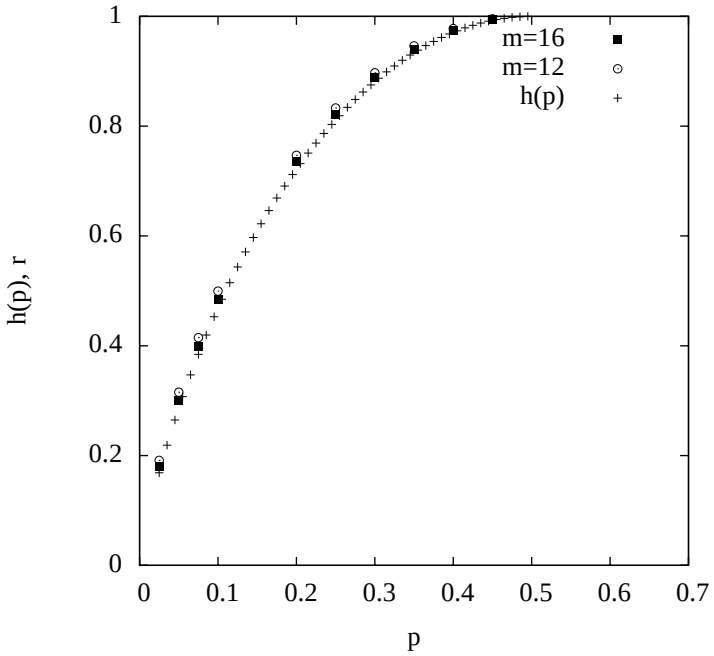

Fig. 2: Initial result.

- Universal
- Application for transmission
- Expected word length can be computed together with variance
- Explain transformation and recursive implementation
- Reasons for working out: different algorithm, finite length analysis, polarization entropy, good practical results, fixed indices depends on distribution of source
- complexity wise codes are very good compared to LDPC
- non-binary
- sub-optimal decoding algorithms
- Some differences with channel coding setting, we actually compute entropies??
- Analyze algorithms, potentially non-binary

VII. Conclusion

The conclusion goes here.

VIII. Proof of