15-859: Information Theory and Applications in TCS **CMU: Spring 2013**

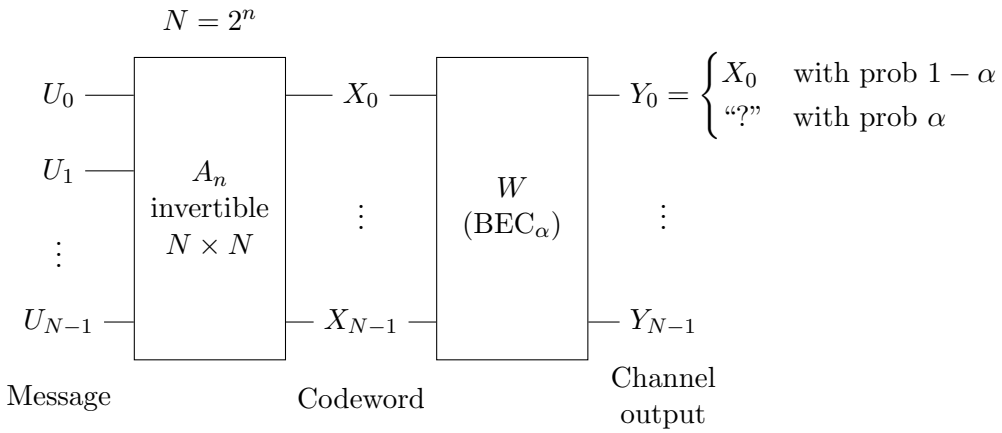
Lecture 11: Polar codes construction

February 26, 2013

Lecturer: Venkatesan Guruswami *Scribe: Dan Stahlke*

1 Polar codes: recap of last lecture

For a linear code, the codewords are related to the messages by a linear transformation (modulo 2). Since the codewords are longer than the messages, the matrix that represents the linear transformation will be rectangular. However, it will simplify the analysis to pad the input message with some extra bits which are constant ("frozen") in order to make the matrix square. It doesn't matter if those bits get corrupted since they are not part of the message and we already know what their value was supposed to be anyway.



The encoding operation is $X_0^{N-1} = A_n U_0^{N-1}$. As described last lecture, the matrix A_n is defined recursively and ends up being in a tensor product form $A_n = G_2^{\otimes n} \cdot B_n$ where B_n denotes bit reversal.

The idea behind the polar code is that when decoding a corrupted message Y_0^{N-1}, the vast majority of the uncertainty ends up being in the subset of U_0^{N-1} that were frozen. Specifically, consider the entropy chain rule:

$$\sum_{i=0}^{N-1} H(U_i | U_0^{i-1}, Y_0^{N-1}) = \sum_{i=0}^{N-1} \underbrace{H(X_i|Y_i)}_{=\alpha} = N\alpha. \tag{1}$$

We want a fraction α of the terms on the left hand side to be nearly 1 and the rest nearly 0. The former will be nearly uncertain and we take these to be the frozen bits, the latter will be nearly certain and we feed the message into these. The goal of this lecture is to prove that the entropy accumulates into an α fraction of the U_i. This is called polarization.

2 Proof of polarization

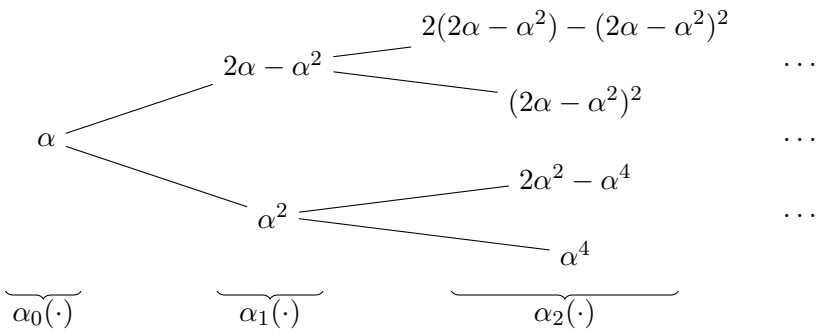
First, give a name to the terms on the left hand side of (1) by defining a function $\alpha_n : \{0, 1, \ldots, 2^n - 1\} \to [0, 1]$ by

$$\alpha_n(i) = H(U_i | U_0^{i-1}, Y_0^{N-1}). \tag{2}$$

This is the probability that U_i is not known given U_0^{i-1} and Y_0^{N-1}. Last lecture it was shown that $\alpha_n(\cdot)$ evolves according to the recurrence

$$\alpha_n(i) = \begin{cases} 2\alpha_{n-1}(\lfloor \frac{i}{2} \rfloor) - \alpha_{n-1}(\lfloor \frac{i}{2} \rfloor)^2 & \text{if } i \text{ even} \\ \alpha_{n-1}(\lfloor \frac{i}{2} \rfloor)^2 & \text{if } i \text{ odd} \end{cases} \tag{3}$$

It is helpful to visualize this recurrence as a tree:



Think of α_n as a random variable on $\{0, 1\}^n$ or $\{0, 1, \ldots, 2^{n-1}\}$ induced by the uniform distribution on $\{0, 1\}^n$. Specifically, with $i(b)$ representing the integer having b as the binary representation, define the random variable

$$Z_n(b) = \alpha_n(i(b)) \in [0, 1] \tag{4}$$

with the uniform distribution on $\{0, 1\}^n$. Then $\mathbb{E}(Z_n)$ is the average value of the nodes on level n of the tree. We want to show that as n grows Z_n approaches a Bernoulli distribution, so that the values of the leafs are all close to either zero or one. Suppose we know Z_n. Then the value of Z_{n+1} is

$$Z_{n+1} = \begin{cases} 2Z_n - Z_n^2 & \text{with probability } 1/2 \\ Z_n^2 & \text{with probability } 1/2 \end{cases} \tag{5}$$

Therefore $\mathbb{E}[Z_{n+1}|Z_n] = \frac{1}{2}\mathbb{E}[2Z_n - Z_n^2] + \frac{1}{2}\mathbb{E}[Z_n^2] = Z_n$ and so $\mathbb{E}[Z_n] = \mathbb{E}[Z_0] = \alpha$. This sequence $Z_0, Z_1, \ldots$ is a martingale.

Definition 1 (Martingale)**.** *A sequence of random values $Z_0, Z_1, \ldots$ such that $\forall n$*

(i) $\mathbb{E}\{|Z_n|\} < \infty$

(ii) $\mathbb{E}\{Z_{n+1}|Z_0, \ldots, Z_n\} = Z_n$

is a martingale.

Since our random values $\{Z_n\}$ are bounded in $[0,1]$, they converge to a fixed point by virtue of the so-called "martingale convergence theorem". We have $\forall \epsilon > 0$,

$$\lim_{n\to\infty} \Pr_{\omega}(|Z_{n+1}(\omega) - Z_n(\omega)| > \epsilon) = 0, \tag{6}$$

where ω is the tree path. Therefore $\mathbb{E}\{|Z_{n+1}(\omega) - Z_n(\omega)|\} \xrightarrow{n\to\infty} 0$. But this can be expanded out to

$$\begin{aligned}\mathbb{E}\{|Z_{n+1}(\omega) - Z_n(\omega)|\} &= \frac{1}{2}\mathbb{E}[2Z_n - Z_n^2 - Z_n] + \frac{1}{2}\mathbb{E}[Z_n - Z_n^2] &(7)\\ &= \mathbb{E}[Z_n - Z_n^2] &(8)\\ &= \mathbb{E}[Z_n(1 - Z_n)] \xrightarrow{n\to\infty} 0. &(9)\end{aligned}$$

So most of the time $Z_n(1 - Z_n) \approx 0$. This only happens if Z_n is close to 0 or 1 with high probability, $\lim_{n\to\infty}[Z_n \in (\epsilon, 1-\epsilon)] = 0$. Since $\mathbb{E}[Z_n] = \mathbb{E}[Z_0] = \alpha$, we have $\Pr(Z_n \approx 0) = 1 - \alpha$ and $\Pr(Z_n \approx 1) = \alpha$, or more precisely $\forall \epsilon > 0$

$$\lim_{n\to\infty} \Pr(Z_n < \epsilon) = 1 - \alpha \tag{10}$$

$$\lim_{n\to\infty} \Pr(Z_n > 1 - \epsilon) = \alpha. \tag{11}$$

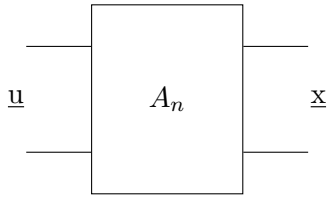
Quantitatively, one can prove

$$\lim_{n\to\infty} \Pr\left(Z_n < 2^{-N^{0.49}}\right) = 1 - \alpha \tag{12}$$

where 0.49 is just some number less than a half (since the random variable gets squared about $n/2$ times).

Corollary 1. *For all large enough n, there exists a subset $F_n \subseteq \{0, 1, \ldots, 2^n - 1\}$ (the "bad bits") with $|F_n| \leq (\alpha + o(1))N$ such that $\alpha_n(i) < 2^{-N^{0.49}}$ for all $i \notin F_n$.*

3 Obtaining the code



Message Codeword

Freeze $u_i, i \in F_n$ to 0. This gives a code of rate $1 - \alpha - o(1)$. Formally, this is a linear code

$$c_n = \{A_n \underline{u} \mid \underline{u} \in \{0,1\}^N, u_i = 0 \ \ \forall i \in F_n\}. \tag{13}$$

The decoding procedure (for BEC_α) is as follows. For $i = \{0, 1, \ldots, 2^n - 1\}$,

(i) If $i \in F_n$ then $u_i = 0$.

(ii) If $i \notin F_n$ then recover u_i from u_0^{i-1}, y_0^{N-1} if possible. If not possible, then FAIL. This is done using maximum likelihood decoding.

Claim: The probability that this algorithm fails to decode a random[1] input message $\underline{u}$ under BEC_α noise is (using the union bound)

$$\Pr[\text{FAIL}] \le \sum_{i \notin F_n} H(u_i | u_0^{i-1}, y_0^{N-1}) \tag{14}$$

$$\le \sum_{i \notin F_n} \alpha_n(i) \tag{15}$$

$$\le N \cdot 2^{-N^{0.49}} \xrightarrow{n \to \infty} 0. \tag{16}$$

Efficiency: For BEC_α one can compute F_n efficiently (poly(N)) by computing the tree of values $\alpha_n(i)$. Once F_n is known, recovery is efficient. Step (ii) can be implemented recursively mirroring the analysis that we have done.

4 General channels

For general channels, the recursive setup remains the same, and A_n is the same, but F_n will depend on the channel. Consider recovery of $u_{2i} = v_i \oplus w_i$ and $u_{2i+1} = w_i$ from $v_0^{i-1}, w_0^{i-1}, y_0^{N-1}$. Define

$$\beta := H(U_{2i} | \underbrace{V_0^{i-1}, W_0^{i-1}}_{U_0^{2i-1}}, Y_0^{N-1}) \tag{17}$$

$$\gamma := H(U_{2i+1} | V_0^{i-1}, W_0^{i-1}, Y_0^{N-1}). \tag{18}$$

We know β is small because of Fano's inequality, so we can estimate each bit from the previous ones. The random variables $V_i | V_0^{i-1}, Y_0^{N/2-1}$ and $W_i | V_0^{i-1}, Y_{N/2}^{N-1}$ are i.i.d. and have entropies β and γ. By conservation of entropy, $\beta + \gamma = 2\alpha$ since $u_{2i} = v_i \oplus w_i$ and $u_{2i+1} = w_i$. It can be shown that one of β, γ is greater than α and the other less, so there is polarization.

Lemma 1 (proof omitted)**.** *Let (B_1, D_1) and (B_2, D_2) be i.i.d. pairs of discrete random variables, $B_1, B_2 \in \{0, 1\}$. If $H(B_i | D_i) \in (\delta, 1-\delta)$ for $\delta > 0$ then $H(B_1 + B_2 | D_1, D_2) - H(B_1, D_1) \ge \gamma(\delta) > 0$.*

By this lemma, $\beta > \alpha$ and $\gamma < \alpha$.

In order to get quantitative bounds that imply high probability decoding, we compute not $H(U_i | \dots)$ but so called "Bhattacharya parameters" of the channel,

$$Z(W) = \sum_{y \in Y} \sqrt{p(y|0) p(y|1)} \tag{19}$$

$$= \langle v_0, v_1 \rangle, \tag{20}$$

where $v_0 = \left(\sqrt{p(y|0)} \right)_y$ and $v_1 = \left(\sqrt{p(y|1)} \right)_y$. For the BEC_α channel we have $Z(\mathrm{BEC}_\alpha) = \alpha$ so analysis was just tracking the perceived erasure probabilities of all the intermediate bits during the decoding process.

[1] Since the code is linear, random $\iff$ fixed.

5 Some comments

We conclude the discussion of polar codes with some comments:

- The technique actually gives an alternate proof of Shannon's noisy coding theorem, for the case of binary input symmetric output memoryless channels. As polar codes are linear, they can only achieve capacity of channels for which the uniform input distribution X achieves $\max_X I(X;Y)$.

- The method generalizes to prime alphabets without changing the code construction. For non-prime alphabets a different polarizing map has been constructed.

- Taking Y to be empty in our analysis gives us a source code for compressing N i.i.d copies of the random variable X:

  $$X_0^{N_1} \to U_0^{N-1} = M_n X_0^{N-1} \ .$$

  For all but a vanishing fraction of indices i, we will have $H(U_i|U_0^{i-1}) \approx 0$ or $H(U_i|U_0^{i-1}) \approx 1$. As the sum of these entries equals $NH(X)$, the number of bits for which $H(U_i|U_0^{i-1}) \approx 0$ is about $(1 - H(X))N$, and we just need to store the remaining $\approx H(X)N$ bits of U_0^{N-1} and we can recover the rest, and therefore also X_0^{N-1}.

- For the construction of the channel code (or the source code as in the previous item), we need to know which indices have conditional entropy close to 0 and which ones have conditional entropy close to 1. Taking the above source code setting, it might seem that the entropies $H(U_i|U_0^{i-1})$ for an initial segment of i's will be ≈ 1, and as we condition more and more, the later $H(U_i|U_0^{i-1})$ will be ≈ 0.

  Unfortunately there seems to be no simple characterization of which entropies polarize to 0 and which polarize to 1. However, there is an algorithmic way to estimate these quantities (or to be accurate, their proxies, the associated Bhattacharyya parameters) which can be used to compute the generator matrix of the code (or the compression matrix for source coding).

- Soon after their discovery, polar codes have been applied to many other classic information theory problems such as Slepian-Wolf, Wyner-Ziv, Gelfand-Pinsker, etc.

- A well written reference on polar codes is the survey "Polarization and polar codes" by Eren Sasoglu which has been published as Volume 8, No. 4 in the series Foundations and Trends in Communications and Information Theory.