CS 229r Information Theory in Computer Science	February 28, 2019
Lecture 10	
Instructor: Madhu Sudan	*Scribe: Alec Sun*

1 The Lecture Road Map

Today we will:

- Wrap up polar codes.
- Start communication complexity and give some:
  - Basic definitions
  - Examples
  - Lower bounds

2 Summary of Polar Codes

The essential objects in the analysis of polar codes are martingales. Let $x_0, \ldots, x_t$ be a $[0,1]$-bounded Martingale. There are two ways to characterize the polarization:

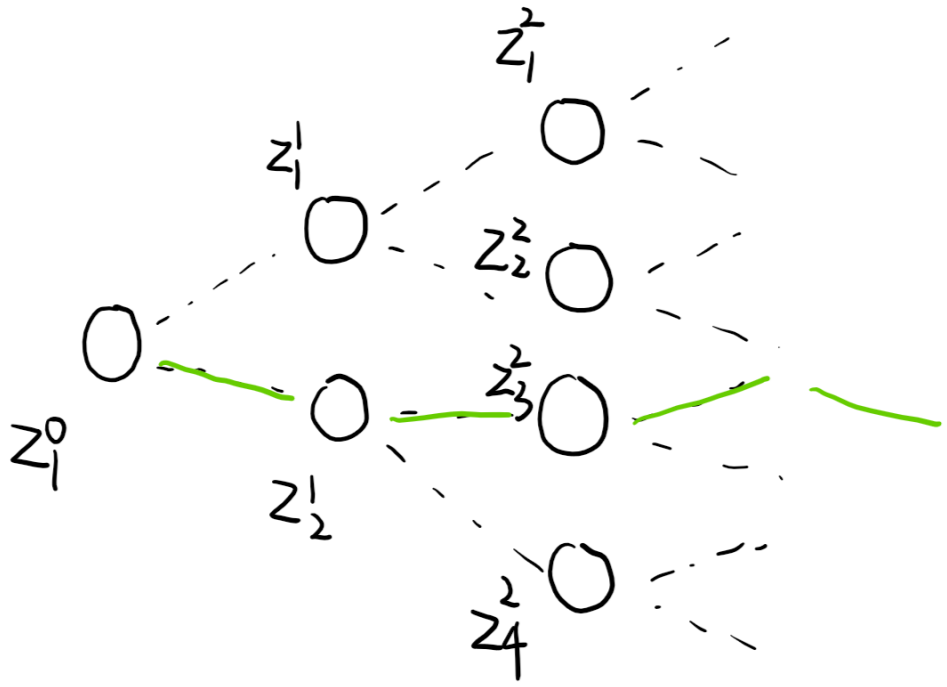
- **Local Polarization.** There is variance in the middle, namely $\forall \tau, \exists \varepsilon$ such that $\forall i$, $X_{i-1} \in (\tau, 1-\tau)$ then $\mathrm{Var}(X_i \mid X_{i-1}) \geq \sigma^2$. There is also suction at the ends, namely $\exists \theta > 0$ such that $\forall c, \exists \tau$ such that $\forall i$, if $X_{i-1} \leq \tau$ then
$$\Pr[X_i < X_{i-1}/c] \geq \theta.$$
In our martingale we had $\theta = 1/2$.
- **Strong Polarization.** If $\forall c, \exists \beta < 1$ such that $\forall t$,
$$\Pr[x_t \in (c^{-t}, 1 - c^{-t})] = O(\beta^t)$$
where the $O(\cdot)$ is respect to t.

A theorem that we did not prove is the implication of strong polarization from local polarization. This theorem is nice because we no longer need to consider the long-term behavior in the analysis and can instead focus on a single step.

What we were doing is looking at a tree-like process. The entropy X_{i-1} is a conditional entropy on the variables above it. Why should a process like this show local polarization effects? Label a node Z_a^{i-1} that branches out to Z_b^i, Z_c^i. Then we are looking at

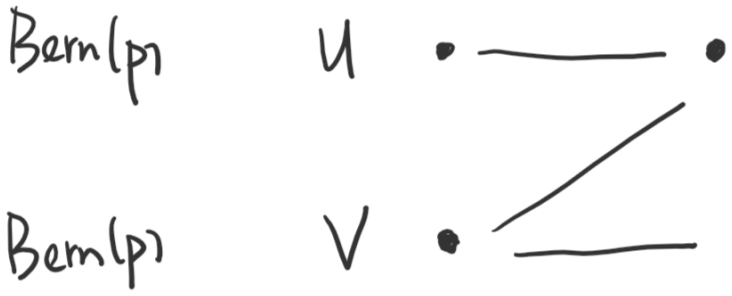
$$\begin{aligned} H(Z_a^{i-1} \mid Z_{<a}^{i-1}) \\ H(U \mid A) \\ H(V \mid B) \\ H(U+V \mid A, B) \\ H(V \mid A, B, U+V) \end{aligned}$$

Here U corresponds to one node conditioned on nodes above it, and V corresponds to a different node conditioned on a different set of nodes above it.



To get a rough sense of how entropy behaves, we can drop the conditional part of entropy and consider the following question: How do $H(U), H(V)$ compare with $H(U+V)$ and $H(V \mid U+V)$? If U, V are i.i.d. Bernoulli variables, what does $U+V$ look like? Does it have variance in the middle? Does it have suction at the ends?

3 Polarization of Bernoulli Random Variable



Let $0 < p < 1/2$ and let $u, v \sim \text{Bern}(p)$. Then

$$u + v \sim \text{Bern}(2p(1-p)).$$

If we define $X_{i-1} = h(p)$ where

$$h(p) = -p \log p - (1-p) \log(1-p),$$

then with probability $1/2$ we will have $X_i = h(2p(1-p))$. If we have $\tau < h(p) < 1 - \tau$ and

$$\varepsilon_2 < p < \frac{1}{2} - \varepsilon,$$

then

$$\varepsilon_2' \leq 2p(1-p) - p.$$

This implies that

$$h(2p(1-p)) - h(p) \geq \varepsilon_2'',$$

where we note as a remark that $\varepsilon_2, \varepsilon_2', \varepsilon_2''$ are constants that depend on τ. establishing variance in the middle once we show that a change from p to $2p(1-p)$ induces a significant change in entropy.

What about suction at the ends?

- At the high end we have

  $$p = \frac{1}{2} - \varepsilon$$

  for $\varepsilon \to 0$. We can use the approximation

  $$h(p) \approx 1 - \Theta(\varepsilon^2),$$

  as well as

  $$2p(1-p) = \frac{1}{2} - \Theta(\varepsilon^2).$$

  Combining these two facts, we have

  $$h(2p(1-p)) \approx 1 - \Theta(\varepsilon^4).$$

  Hence we are dropping by as large of a constant factor as we want, but we need ε to be small enough so that

  $$\Theta(\varepsilon^4) \ll \Theta(\varepsilon^2).$$

- At the low end, we have $p \to 0$, and

  $$h(p) \approx p \log \frac{1}{p}.$$

  We also have $2p(1-p) \approx 2p$. We want the condition

  $$2h(p) - h(2p) \ll h(p)$$

  to hold. We can expand out using the approximation to

  $$2p \approx 2p \log \frac{1}{p} - 2p \log \frac{2}{p} \ll p \log \frac{1}{p}.$$

Thus at the low end at which

$$(1 - X_i) \approx (1 - X_{i-1})^2$$

holds, we will have

$$X_i = \frac{X_{i-1}}{\log \frac{1}{X_{i-1}}}.$$

To summarize, we have provided a rough sketch as to why our martingale polarizes locally. Putting this together, we have a very nice encoding and compression theorem with $n = \text{poly}(1/\varepsilon)$. Hence if we want to be ε close to capacity, we do not have to have super long lengths.

Remark We can prove that the decoding error of our procedure is smaller than any polynomial. But can we get exponentially small error, $2^{-\Omega(n)}$. To date, we know of no construction that can get this low error, but modifying our analysis, we can get errors like $2^{-\Omega(\sqrt{n})}$.

Remark In our polar code, the errors were all independent. Even though our analysis revolved on this fact, this is not the end of the story as there are more recent results on when the errors introduced in the encoding are dependent on each other.

Since the first part of today's lecture is simply a summary of last class, we omit any new exercises on this topic and refer to last lecture's scribe notes for exercises relating to polar coding. A collection of exercises in the next topic, communication complexity, is found at the end of these scribe notes.

4 Definitions in Communication Complexity

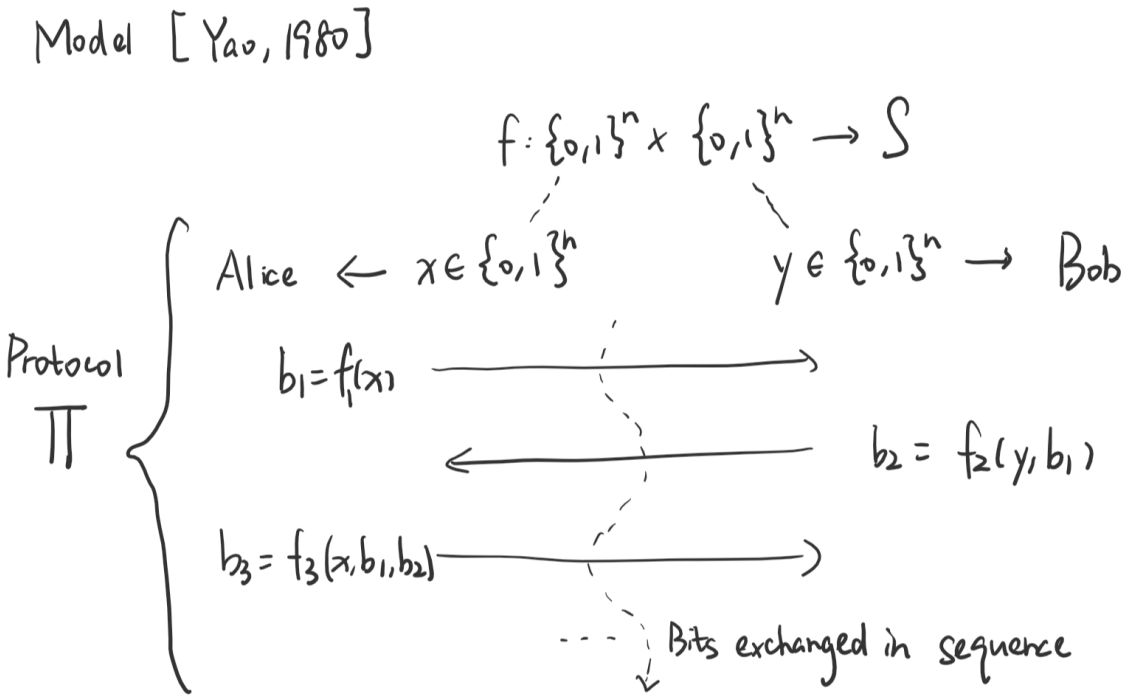
Starting from today we will shift gears and talk about communication complexity. Suppose we have multiple parties that want to compute a shared quantity given some private information that the parties have. One way of doing this is for everyone to send everyone else everything they know. But in this subfield of complexity theory, we are interested in exploring what can be done with less transfer of information between parties. Much of the theory behind communication complexity was developed in a seminal paper by Andrew Yao in 1980.

Definition 1. *Consider two people Alice and Bob. Both people know a function* $f : \{0,1\}^n \times \{0,1\}^n \to R$. *Suppose that Alice knows a private string* $x \in \{0,1\}^n$ *and Bob knows a private string* $y \in \{0,1\}^n$. *The goal is for Bob to output* $f(x,y)$. *Bob certainly needs to hear information about* x *from Alice in order to do this. We model this as rounds of communication alternating between Alice and Bob:*

1. *Alice sends* $b_1 = f_1(x)$ *for some function* f_1.
2. *Bob sends* $b_2 = f_2(y, b_1)$ *for some function* f_2.
3. *In general, Alice sends* $b_i = f_i(x, b_1 \cdots b_{i-1})$ *for some function* f_i *where* $i \equiv 1 \bmod 2$.
4. *At the end, Bob outputs* $f(x,y) = O(y, b_1 \cdots b_k)$ *for some function* O.

Define the functions $(f_1, \ldots, f_k, O)$ *as the* protocol *for the* communication Π.

The goal of communication problems is to design Π that computes the function f and minimizes k, the number of bits exchanged.



Remark In some settings we might also care about the number of *rounds*, the number of times that Alice and Bob have to alternate sending one-way streams of information. Often this is captured by a restriction called *bounded rounds*, when we want to find the minimum number of bits needed to be transferred subject to an upper bound on the number of rounds.

Remark A special case of this model is *one-way* communication, when Alice sends a single message to Bob and Bob must output $f(x, y)$.

Remark Here we do not care about the computation time of either Alice or Bob. We give them infinite computational power because what matters is only the information-theoretic number of bits needed to be transferred.

Before we move to negative results in communication complexity, we show some positive results. Let us start with an easy example to "get our juices flowing." Clearly there should be better ways to communicate that are not sending the entire message.

Example 2. *Suppose f is the parity function*

$$f(x, y) = \bigoplus_{i=1}^{n} (X_i \oplus y_i).$$

Then Alice only needs to send one bit

$$b = \bigoplus_{i=1}^{n} x_i$$

to Bob. Bob will output

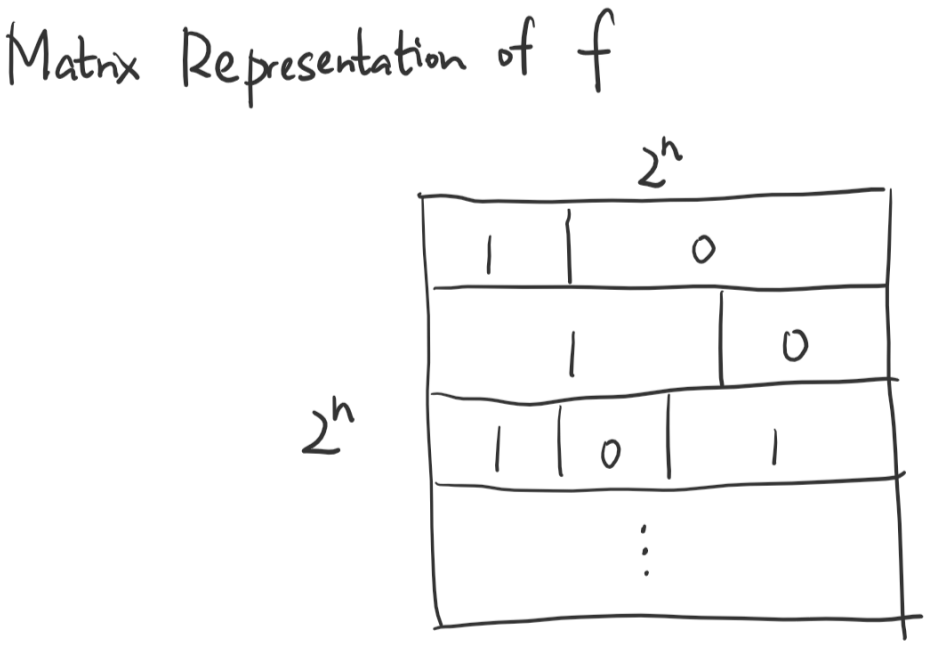
$$b \oplus \left(\bigoplus_{i=1}^{n} y_i \right).$$

In a sense this is a fake example because when Alice received the information x she could have been able to compress it down into 1 bit to begin with, so she really only holds 1 bit of information when it comes to computing the parity function.

Definition 3. *Let $CC(f)$ denote the minimum number of bits needed to be transferred to compute f.*

5 The Tiling Method

Example 4. *If $CC(f) = k$ and this is satisfied by a one-way function, consider the $2^n \times 2^n$ matrix M_f where the entry is position (x, y) is $f(x, y)$. Rearrange the rows to that the first group of rows are those in which Alice sends 0^k, followed by the rows in which Alice sends $0^{k-1} \circ 1$, and so on up to 1^k. In order for the procedure to work, the columns within each group corresponding to different y must consist of all zeroes or all ones, because Bob must be able to determine $f(x, y)$ only given the k-bit message that Alice sends.*



Let us now consider multi-way communication schemes. Suppose that Alice sends a bit at Stage 1, partitioning the rows into two groups labeled by 0 and 1. Then Bob sends a bit, partitioning the columns within Alice's row groups into two groups labeled by 0 and 1. In general, at the start of Stage i, there will have been 2^{i-1} different rectangular groups in the matrix corresponding to previous choices of $i-1$ bits that Alice and Bob have sent each other and whose union is the $2^n \times 2^n$ rectangle. Then when either Alice or Bob sends a bit at this stage, we will partition each rectangle either by rows or by columns into two smaller rectangles. We can summarize this as follows:

Definition 5. *At every stage the set of (x, y) that are consistent with the transcript so far form a "rectangle" that that corresponds to the Cartesian product $S \times T, S \subseteq \{0,1\}^n, T \subseteq \{0,1\}^n$. After k bits of communication there will be 2^k rectangles. In order for Bob to output $f(x, y)$ after k bits are exchanged, 2^k rectangles must cover all the ones in M_f in order to differentiate 1 from 0.*

Let M_f be the sum of 2^k matrices each of rank 1. Then M_f has rank at most 2^k. If rank (M_f) is large then we can guess that f needs a lot of round of communication. Indeed, we can prove that:

Theorem 6 (Communication Complexity Lower-Bound)**.** *For all f, $CC(f) \geq \log rank\ (M_f)$.*

There is a related conjecture:

Theorem 7 (Communication Complexity Upper-Bound)**.** *For all f, $CC(f) \leq \text{poly}(\log rank\ (M_f))$.*

The known lower-bound for communication complexity allows us to construct functions that have high communication complexity and provably show that have high communication complexity.

Example 8. *The simplest possible example is $M_f = I_{2^n \times 2^n}$. Then $f(x, y) = 1$ if $x = y$ and $f(x, y) = 0$ if $x \neq y$. This seems to be a "hard" function to compute, and in particular it is true that the deterministic communication complexity is n. But this function is actually "easy" if we were to allow randomness in communication.*

6 Communication With Randomness

By convention, protocol designs for communications with random coin flips usually have a person send the result of the coin flip that signifies that they have tosses a coin, and then the bit or bits that they send as a result of the coin flip. We describe two variants of randomized communication complexity using coin flips.

Definition 9 (Private Coins)**.** *In this model, Alice and Bob are allowed to toss coins whose results are known only to the person who tossed them. The two people must output $f(x, y)$ correctly for all tuples (x, y) with probability at least $2/3$ over the randomness of the protocol. We denote this complexity by Priv-CC(f).*

Definition 10 (Public Coins)**.** *Alice and Bob share a random string corresponding to a string of public coins flips, and they use this string to aid their communication. We denote this complexity by Public-CC(f).*

There are two main results regarding these definitions.

Proposition 11. *For all* $f : \{0,1\}^n \times \{0,1\}^n \to R$,

$$CC(f) \leq 2^{O(\textit{Priv-CC}(f))}.$$

Proposition 12. *For all* $f : \{0,1\}^n \times \{0,1\}^n \to R$,

$$\textit{Priv-CC}(f) \leq \textit{Public-CC}(f) + O(\log n).$$

Equality is tight for the two inequalities above. We consider randomized protocols that achieve equality using error correcting codes.

Example 13. *Consider a function* $E : \{0,1\}^n \to \{0,1\}^{2n}$ *such that for every* $x \neq y$, *their outputs differ in a lot of bits,*

$$\#\{i : E(x)_i \neq E(y)_i\} \geq \frac{n}{5}.$$

We can construct such a E *either explicitly or randomly using the probabilistic method. Either approach is known in the literature as "hashing."*

Here is a procedure for testing whether or not $x = y$. *Suppose we pick an index* i *known publicly uniformly at random from* $[n]$. *The probability that* $E(x)_i \neq E(y)_i$ *is at least* $1/5$ *by assumption. Hence if we repeat this* $O(1)$ *times, we can achieve a success probability of greater than* $2/3$, *implying that the identity function has a public coin communication complexity of* $O(1)$. *By Proposition 12, the private coin communication complexity is* $O(\log n)$.

7 High-Complexity Problems

We now turn to problems that we know or believe are hard under communication complexity:

1. **Inner Product.** This is the $\mathbb{F}_2$ inner product

$$\text{IP}(x, y) = \sum_{i=1}^{n} x_i y_i \bmod 2.$$

   We know that this problem is hard.

2. **Set Disjointness.** We view x, y as characteristic vectors of sets $S, T \subseteq [n]$ respectively. We want to know whether or not they intersect. This problem is defined by

$$\text{DISJ}(X, Y) = \bigvee_{i=1}^{n} (x_i \wedge y_i)$$

   and is much more elusive to prove lower bounds with. However, it is the most popular problem for proving things in communication complexity.

The problem of Set Disjointness will be discussed next lecture.

8 Exercises for Communication Complexity

1. Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be such that all rows of M_f are distinct. Show that $\mathrm{CC}(f) \geq \log n$.

2. For two $n \times n$ matrices A and B, show that the rank of $A \otimes B$ over any field is equal to the products of the ranks of A and B.

3. Consider x, y as vectors over $\mathbb{F}_2^n$ and let $f(x, y)$ be their inner product mod 2. Prove using the rank method that the communication complexity is at least $n - 1$.

9 Solutions to Exercises for Communication Complexity

1. We claim that the rank of M_f over $\mathbb{F}_2$ is at least n, which will imply that the communication complexity is at least $\log n$. Suppose for contradiction that the rank is at most $n - 1$. Any subspace of dimension at most $n - 1$ has at most 2^{n-1} distinct vectors by considering a basis and noting that coefficient of 0 and 1 associated with the basis vectors generate all distinct vectors in the subspace. If the 2^n row vectors of M_f are distinct, the dimension of their span therefore must be at least n.

2. We give only the proof for algebraically closed fields, noting that plenty of references exist for the general case. We use the result from linear algebra for algebraic closed fields that we can find matrices P and Q such that PAP^{-1} and QBQ^{-1} are upper triangular. Note that conjugation by P and Q leaves the rank invariant. Then

$$(P \otimes Q)(A \otimes B)(P \otimes Q)^{-1} = (PAP^{-1}) \otimes (QBQ^{-1})$$

   is a tensor product of upper-triangular matrices. By the definition of the Kronecker product, the right hand side is upper-triangular and the upper-triangular property lets us see that the rank of the right hand side is (rank A) $\cdot$ (rank$B$). Tje rank of the left hand side is rank $A \otimes B$ since conjugation leaves rank invariant. Hence we are done.

3. Consider a monochromatic rectangle $R = A \times B$ filled with zeros, which means that $\langle a, b \rangle = 0$ for all $a \in A, b \in B$. Let $r =$ rank (span A) and $s =$ rank (span B), meaning that

$$\begin{aligned} |A| &\leq 2^r \\ |B| &\leq 2^s \\ |A||B| &\leq 2^{r+s} \end{aligned}$$

   Since the vector spaces span A and span B are orthogonal, we must have $r + s \leq n$, so the size of a monochromatic zero rectangle is at most 2^n. We can easily verify that there are at least 2^{2n-1} zeros appearing in the matrix M_f, so by the Tiling Method we have

$$\begin{aligned} CC(f) &\geq \log\left(\frac{2^{2n-1}}{2^n}\right) \\ &= n - 1. \end{aligned}$$