CS 229r Information Theory in Computer Science Feb 28, 2019

Lecture 10

Instructor: Madhu Sudan *Scribe: Kevin Liu*

1 Overview

1.1 Outline

1. Polar Codes Wrap up
2. Communication Complexity
   (a) Basic definitions
   (b) Examples
   (c) Lower Bounds

2 Polar Codes Wrap up

Recall that in our analysis of polar codes, we are attempting to show the polarization properties of the bounded martingale $X_i \triangleq H(Z_{j_i}^i | Z_{<j_i}^i)$. We defined two notions of polarization: local polarization, characterized by local single-step properties, and strong polarization, characterized by long-term properties.

2.1 Local Polarization

A sequence $X_0, ..., X_t$ polarizes *locally* if it satisfies two properties:

1. Variance in the Middle: $\forall \tau, \exists \sigma$ s.t. $\forall i$, if $X_{i-1} \in (\tau, 1-\tau)$, then $\text{Var}(X_i | X_{i-1}) \geq \sigma^2$
2. Suction at the ends: $\exists \theta > 0, \forall c, \exists \tau$ s.t. $\forall i$, if $X_{i-1} \leq \tau$, then $\Pr[X_i < \frac{X_{i-1}}{c}] \geq \theta$. On the high end, a similar condition holds for $1 - x$.

2.2 Strong Polarization

A sequence $X_0, ..., X_t$ polarizes *strongly* if:

$$\forall c, \exists \beta < 1 \text{ s.t. } \forall t, \Pr\left[X_t \in (c^{-t}, 1 - c^{-t})\right] < O(\beta^t)$$

Theorem 1. *Local Polarization* $\implies$ *Strong Polarization*

This is useful because we don't need to understand long term behavior, just the next step.

2.3 Back to our Martingale

The quantities we're concerned with $H(Z_a^{i-1} | Z_a^{i-1})$ take the forms $H(U|A)$, $H(V|B)$, $H(U+V|A,B)$, and $H(V|A,B,U+V)$. Why should our martingale show local polarization effects? To get a sense of what's going on, we will compare $H(U)\&H(V)$ with $H(U+V)\&H(V|U+V)$.

Setup: $0 < p < \frac{1}{2}$; $U, V \sim$ Bern(p); $X_{i-1} = h(p)$

Then, $U + V \sim$ Bern(2p(1-p)), so w.p. $\frac{1}{2}$, $X_i = h(2p(1-p))$

1. Variance in the middle

$$\tau < h(p) < 1 - \tau$$
$$\varepsilon_2 < p < \frac{1}{2} - \varepsilon_1$$
$$\implies \varepsilon_2' \leq 2p(1-p) - p$$
$$\implies h(2p(1-p)) - h(p) \geq \varepsilon_2''$$

2. Suction at the ends

   - High End

$$p = \frac{1}{2} - \varepsilon \implies h(p) \approx 1 - \theta(\varepsilon^2)$$
$$2p(1-p) = \frac{1}{2} - \theta(\varepsilon^2)$$
$$h(2p(1-p)) \approx 1 - \theta(\varepsilon^4)$$

     Satisfied if ε is sufficiently small.

   - Low End

$$p \to 0 \implies h(p) \approx p \log \frac{1}{2}$$
$$2p(1-p) \approx 2p$$
$$2h(p) - h(2p) << h(p)?$$
$$2p \log \frac{1}{p} - 2p \log \frac{2}{p} \approx 2p << p \log \frac{1}{p}$$

     Satisfied if p is sufficiently small.

Theorem 2. *Our martingale polarizes locally.*

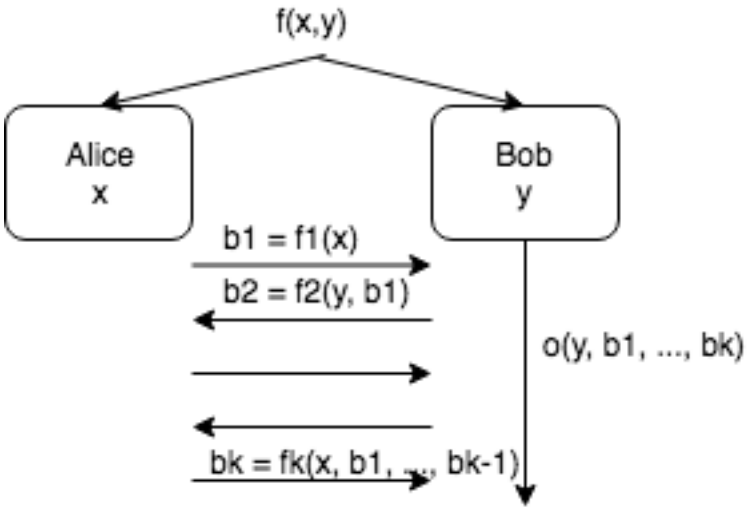
The implication is that we have a nice coding and compression with $n \approx \text{poly}(\frac{1}{\varepsilon})$. The crux is that $O(\beta^t)$ is exponentially small.

Next steps:

1. We've proved that Pr[error] < polynomial. Is that the best we can do?
2. What if errors are not independent? We can still use polar codes, but need to be more careful with construction and analysis.

3 Communication Complexity

Communication Complexity encapsulates a broad class of questions and was introduced in Andrew Yao's seminal paper in 1979. The general setup is there are multiple players, each with private information. The goal is to compute a joint function of the private information while minimizing messages exchanged between players.



3.1 Definitions

3.1.1 Setup

- Two players: Alice and Bob
- Function $f : \{0,1\}^n \times \{0,1\}^n \to R$ known to both players.
- Alice and Bob exchange messages. Messages from Alice to Bob denoted $b_i = f_i(x, b_{<i})$. Messages from Bob to Alice denoted $b_i = f_i(y, b_{<i})$.
- Finally, Bob outputs $o(y, b_1, ..., b_k)$, presumably equal to $f(x, y)$.

$(f_1, ..., f_k, o)$ determine the protocol for communication π.

Goal: Design π that computes f and minimizes k. Denote CC(f) = min # of bits exchanged needed to compute f. Note we don't care about computational complexity of any of the f_i, only care about the number of bits exchanged.

Variants:

1. Bounded number of rounds: communicate with long messages and minimize number of bits exchanged subject to upper-bound on number of rounds.
2. One-way communication: Alice sends messages to Bob, Bob outputs $f(x, y)$

3.2 A Trivial Example: Parity

$f(x, y) = \oplus_{i=1}^{n}(x_i \oplus y_i)$

- Alice sends $\oplus_{i=1}^{n} x_i$ to Bob.
- Bob outputs $(\oplus_{i=1}^{n} x_i) \oplus (\oplus_{i=1}^{n} y_i)$

1-bit communication protocol solves the problem: CC(f) = 1

3.3 Analysis

Let M_f be the $2^n \times 2^n$ matrix representation of f where rows correspond to Alice's input and columns correspond to Bob's input.

Idea: Each message partitions the matrix of possible sets of inputs. At every stage of the exchange, the set of (x, y) that is consistent with the transcript so far form a "rectangle" $(S \subseteq \{0,1\}^n, T \subseteq \{0,1\}^n)$

After k bits of communication, have 2^k rectangles and the value of f should be the same for each remaining consistent x, y. Bob should therefore be able to output.

Equivalently, 2^k rectangles cover all the 1s in M_f. Note, since rectangle matrices are of rank 1. Since M_f is the sum of 2^k matrices, each of rank 1, M_f has rank $\leq 2^k$.

Proposition 3. *CC(f)* $\geq \log$ *rank*(M_f)

Conjecture (still open). $\forall f, \mathrm{CC}(f) \leq \mathrm{poly}(\log \mathrm{rank}(M_f))$

Thus, generally hard problems have M_f with high rank.

$$M_f = I_{2^n \times 2^n} \implies f(x,y) = EQ(x,y)$$

However, we mostly care about randomized communication, not deterministic.

3.4 Randomized Communication Complexity

In the randomized setting, Alice and Bob can independently toss a coin and communicate accordingly. Randomness can be either private or public.

1. Private Coins PRIV-CC(f): Alice and Bob are allowed to toss coins. At the end, Bob must output $f(x, y)$ correctly w.p. $\geq \frac{2}{3}$.
2. Public Randomness RCC(f): Alice and Bob share a random string and use it to communicate.

We have bounds on each in relation to the others:

Proposition 4. *CC(f)* $\leq 2^{PRIV\text{-}CC(f)}$

Proposition 5. *PRIV-CC(f)* $\leq$ *RCC(f)* $+ O(\log n)$

3.5 Randomized Protocol for EQ

EQ is tight for the inequalities above.

Assume we have an error correcting code with encoder $E : \{0,1\}^n \to \{0,1\}^{2n}$, where for every $x \neq y$, $\#\{i | E(x)_i \neq E(y)_i\} \geq \frac{n}{5}$ (code disagrees in a lot of bits when $x \neq y$).

Public randomness: i chosen by common randomness. Alice sends $E(x)_i$. Bob checks it with $E(y)_i$. Repeat this process O(1) times.

Private randomness: Alice chooses i randomly, and sends it along with $E(X_i)$ to Bob for $O(\log n)$ cost.

Thus, EQ is actually considered very easy in the randomized sense.

3.6 Hard Problems

1. $\mathbb{F}_2$ Inner Product

$$\mathrm{IP}(x, y) = \sum_{i}^{n} x_i y_i (\mathrm{mod}\ 2)$$

Requires linear communication, even with randomness

2. Set-Disjointness
x, y are characteristic vectors of sets $S, T \subseteq [n]$. Do they intersect?

$$\mathrm{Disj}(x, y) = \vee_{i=1}^{n} (x_i \wedge y_i)$$

This problem was very elusive to prove lower bounds for, and was very popular - analogous to SAT in computational complexity. It is linear, even with randomness, which we will see in later lectures.