CS Colloqium

Interactive Information Theory December 10, 2015

Lecturers: Gillat Kol Scribe: Kiran Vodrahalli

Contents

1 Introduction

Gillat Kol from the IAS will speak about Interactive Information Theory.

2 Noisy Channel Setting

Shannon showed that $n/(1-H(\epsilon))$ bits are needed to send over a noisy channel. Now we are interested in the Interactive Noisy Channel. Alice and Bob engage in an n-bit long protocol. ow many bits do they need to send over a channel that flips every bit with probability ϵ, so both can retrieve the transcript with high probability? It was showed that there is only blowup by a constant. We should expect however that the interactive setting is harder.

Example 2.1. Players know v_0 and want to compute $g_m \circ g_{m-1} \circ \cdots \circ g_2 \circ g_1(v_0)$. Alice knows $x = (g_1, g_3, \cdots, g_{m-1})$ and Bob knows $y = (g_2, g_4, \cdots, g_m)$. The description of g_i is huge. This is very susceptible to noise. You need to correct all the time, and be on the same page all the time.

We showed a result previously, and in a recent result, we were able to give a deterministic protocol that achieves capacity.

Theorem 2.2. *($KR'13$).*
The interactive channel capacity is $1 - \Theta(\sqrt{H(\epsilon)})$ *for some interesting binary channels. For small enough* ϵ*, interactive channel capacity* $<$ *channel capacity.*

3 Noiseless Channel Setting

Alice has a string x chosen according to a public distribution, and wants to send x to Bob. How many bits does Alice need to send so Bob can retrieve x with high probability. Shannon and Huffman showed that any message can be compressed to its information content $H(p)$. The motivation for the rest of the talk will be **interactive compression**. Alice and Bob will engage in an interactive communication protocol, and we will ask whether the protocol's transcript can be compressed to its information content.

3.1 Communication Complexity

Alice has some input x, Bob has input y, and they want to compute $f(x, y)$, where f is public. How many bits do they need to exchange? They send $m_1(x), m_2(y, m_1), \cdots$. In distributional communication complexity, we have (x, y) chosen from a public joint distribution μ. Players may use private and public randomness. They need to compute $f(x, y)$ with probability > 0.9 over (x, y) drawn from μ.

Definition 3.1. Communication Complexity of a protocol π: $CC_\mu(\pi)$ = expected number of bits exchanged by the players when running π.

So can the protocol's transcript be compressed to its information content? We need an analogue to entropy for the communication setting, called the **information cost**, used in many prior works.

Definition 3.2. Information Cost.
The amount of information the players learn about each others input from the interaction

$$IC_\mu(\pi) = I(\Pi; Y|X) + I(\Pi; X|Y) \tag{1}$$

where X, Y, Π are random variables and (X, Y) is drawn from μ, and Π is π's transcript. I is mutual information. The first term measures how much Alice learns after seeing the full transcript, and the second term measures the symmetric thing for Bob. Then

$$IC_\mu(f) = \inf_{\pi \text{ computes } f \text{ over } \mu}\{IC_\mu(f)\} \tag{2}$$

Now we formally state the interactive compression problem: Given a protocol π, can π be simulated by π' such that $CC_\mu(\pi') \approx IC_\mu(\pi)$ where recall that CC_μ is the number of bits exchanged and IC_μ is the amount of information expressed by π. We should think of π' as a "compressed" version of π. We want to know if this is a polynomial relation.

Braverman showed that there exists π' is $CC_\mu(\pi') \leq 2^{\mathcal{O}(IC_\mu(\pi))}$. No separation between IC and CC was known. Almost all known techniques were too weak to show the result.

3.2 Why is interactive compression more challenging?

In data compression, Alice knows the whole message and can compress it all at the same time. In interactive compression, no player knows the whole conversation before the protocol takes place. You can try to compress round-by-round, but if a round gives only ϵ information, you will still need a bit to simulate the round. And thus the compression is too much, this is no good. We need to handle the case where there is very little information per round, but there are a lot of rounds.

We showed in $Kol'15$ that if Alice's input and Bob's input are independent, interactive compression is possible in polynomial difference. However, this is not true in general. For every k, you can construct a Boolean function f such that $IC_\mu(f) = \mathcal{O}(k)$, but $CC_\mu(f) \geq 2^k$. Only $\mathcal{O}(k)$ bits of information are revealed, however every protocol will communicate at least 2^k bits. Sometimes, exponential is the best you can hope for.

3.3 Direct Sum

Alice has $x_1, \cdots, x_m$, Bob has $y_1, \cdots, y_m$. For all i, (x_i, y_i) is drawn from μ. They want to compute $f(x_1, y_1), \cdots, f(x_m)$ with high probability on each copy. The Strong Direct Sum problem asks if computing m copies simultaneously requires $\Omega(m)$ times the communication needed to solve a single copy. Braverman and Rao showed that $m \to \infty$ gives that the cost per copy is $IC_\mu(f)$. A corollary of our work is that Strong Direct Sum does not hold! You can do better because of the gap.

3.4 Compression when Alice Controls Information

When x, y are independent, Bob has no information about x. When Alice sends a bit, she knows exactly how much information it will give to Bob. With some work, players can chop the tree into $IC(\pi)$ parts, giving 1 info bit. If each part τ is compressed using the $2^{\mathcal{O}(IC(\tau))}$ protocol, then each part is constant communication and we can do better!

In the general case, this does not work, since no one knows when there is additional information gained, since x, y are correlated!