A hard-to-compress interactive task?

Mark Braverman[*]
Department of Computer Science
Princeton University
Email: mbraverm@cs.princeton.edu

Abstract

Whether the information complexity of any interactive problem is close to its communication complexity is an important open problem. In this note we give an example of a sampling problem whose information and communication complexity we conjecture to be as much as exponentially far apart.

1. Introduction

1.1. Basic setup: communication complexity and information complexity

In the basic context of communication complexity, there are two parties (terminals), Alice and Bob. Prior to the communication, they are given a pair of inputs (X,Y): Alice is given X and Bob is given Y. Depending on the context, the input pair (X,Y) may be drawn from a distribution μ. In addition, the parties are given access to a shared source of randomness, which can be accessed without communication over the channel. Alice and Bob need to perform a computational task $T(X,Y)$ on their inputs. We assume that they employ a noiseless binary channel. In this note the task will be to sample a string s from a distribution $\nu = \nu_{X,Y}$ that depends on both players' inputs. In the context of communication complexity, Alice and Bob want to perform $T(X,Y)$ while sending as few bits as possible over the channel. An introduction to communication complexity, and to the state-of-the-art as of 1997 can be found in [KN97].

Computation is performed by means of a protocol. A protocol is a sequence of functions which specify the next message to be sent (and the speaker). In a randomized protocol the parties in are allowed to use sources of randomness. In the case of a sampling task, the players are allowed to communicate using the protocol; after the communication is over, each party outputs a sample. We say that the task is successful if they output the same sample, and the output sample is distributed according to the desired distribution.

The communication complexity $\mathsf{CC}(T,0)$ of a task T is the smallest number of bits that Alice and Bob need to exchange to perform T. We say that a protocol π solves a task T with error ε, if the output of the protocol *on each pair* (X,Y) *of inputs* is statistically ε-close to the output prescribed by $T(X,Y)$. $\mathsf{CC}(T,\varepsilon)$ is the smallest number of bits that Alice and Bob need to exchange to perform T with error probability $\leq \varepsilon$. In the case of a sampling task, this mean that the distribution η sampled by $\pi(X,Y)$ satisfies $\|\eta - \nu_{X,Y}\|_1 \leq \varepsilon$. When error is allowed, it is often assumed to be a small constant, e.g. $\varepsilon = 1/10$.

The communication complexity usually refers to the *worst case* number of bits, although *average case communication complexity* is also sometimes considered. If inputs are given by a distribution μ. the average case communication cost of a protocol π is the expected number of bits exchanged by the protocol on inputs $(X,Y) \sim \mu$. The *average-case communication complexity* of T, $\mathsf{CC}_\mu^{average}(T,\varepsilon)$, is the expected number of bits that the best protocol needs to exchange to solve T with error $\leq \varepsilon$.

1.2. Interactive information complexity

Next we briefly discuss the notion of information complexity. A more detailed overview and references can be found, for example, in [Bra12b]. While communication complexity is concerned with the number of bits Alice and Bob need to exchange to perform a task T, information complexity is concerned with the amount of information they need to exchange irrespective of the actual number of bits transmitted in the process. Informally speaking, the minimum amount of information that needs to be exchanged to perform T is the *information complexity* of T.

For the remainder of the discussion, assume that

[*]Partially supported by an Alfred P. Sloan Fellowship, an NSF CAREER award, and a Turing Centenary Fellowship.

Alice and Bob are given a pair of inputs (X,Y) distributed according to a distribution μ. The have access to public and private randomness. Let π be any protocol. An execution of the protocol on the pair of (random) inputs $(X,Y) \sim \mu$ gives rise to the transcript $\Pi = \Pi(X,Y)$ of the protocol which is itself a random variable. The *information cost* of a protocol Π is the amount of information that the protocol reveals to Alice and Bob about their input. For example, the amount revealed to Alice – who knows X – about Y is given by the conditional mutual information $I(Y;\Pi|X)$. Thus the information cost of π is given by

$$\mathsf{IC}_\mu(\pi) := I(Y;\Pi|X) + I(X;\Pi|Y). \tag{1}$$

The task of finding the communication complexity of a task T can be reformulated as the task of finding a low-communication protocol for computing T. In the same vein, we can use (1) to define the task of finding the *information complexity* of T as the task of minimizing the information complexity of the protocol for T:

$$\mathsf{IC}_\mu(T) := \inf_{\pi \text{ protocol performing } T} \mathsf{IC}_\mu(\pi). \tag{2}$$

The operational meaning of $\mathsf{IC}_\mu(T)$ is equal to the scaling limit of communication complexity (the statement below is slightly informal, see [BR11, Bra12a] for further discussion):

Theorem 1 *Let T^n be the task of performing n independent copies of the task T. Suppose that T allows for some non-zero error $\varepsilon > 0$. Then we have:*

$$\mathsf{IC}_\mu(T) = \lim_{n\to\infty} \mathsf{CC}_{\mu^n}(T^n)/n. \tag{3}$$

Information complexity can be viewed as the interactive analogue of Shannon's entropy. In this context, Theorem 1 can be viewed as the analogue of Shannon's Source Coding Theorem for interactive two-terminal computing. In one-way communication, in addition to the Source Coding Theorem that ties entropy $H(X)$ to the communication cost of X *in the limit*, encoding schemes such as Huffman coding show that the expected cost of transmitting *a single* copy of X is $\leq H(X)+1$ – within one bit of the information-theoretic optimum. Huffman coding can be viewed as a very efficient scheme for compressing X (over a noiseless channel). In the interactive setting, no such scheme is known, and, in fact it is not clear such a scheme exists. The problem of interactive compression can be formulated as follows (cf [Bra12b, Bra12a]):

Problem 2 *How close is the average case ε-error communication complexity $\mathsf{CC}_\mu^{average}(T,\varepsilon)$ of T to its information complexity $\mathsf{IC}_\mu(T,\varepsilon)$? In particular, is it true that*

$$\mathsf{CC}_\mu^{average}(T,\varepsilon) = O(\mathsf{IC}_\mu(T,\varepsilon))? \tag{4}$$

The only general result for a bound on the communication complexity C in terms of the information complexity I that we currently have is of the form $C < 2^{O(I)}$ [Bra12b]. This is a far cry from the linear relationship we ask for in Problem 2. In this note we give an example of a sampling task for which the gap may be as high as $C = 2^{\Theta(I)}$.

There is another viewpoint onto Problem 2. By Theorem 1, $\mathsf{IC}_\mu(F,\varepsilon)$ is equal to the amortized communication complexity of F as long as $\varepsilon > 0$. Thus (4) can be rephrased as

$$\mathsf{CC}_\mu^{average}(T,\varepsilon) = O(\mathsf{CC}_{\mu^n}(T^n,\varepsilon)/n)?$$

or, more familiarly, as

$$\mathsf{CC}_{\mu^n}(T^n,\varepsilon) = \Omega(n \cdot \mathsf{CC}_\mu^{average}(T,\varepsilon))? \tag{5}$$

This latter formulation is known as the *direct sum problem* for distributional communication complexity. In general, the direct sum problem asks whether solving n copies of a problem is n times as hard as solving one copy. It is usually clear that it is at most as hard, but could there be savings in solving n copies in parallel? While this question is usually formulated for T being the task of computing a function, it also makes sense in the context of sampling, which is what we consider here.

2. Main construction

2.1. Background: the Greater-Than (GT_n) function

First, let us consider the problem of comparing two n-bit numbers.

Definition 3 *The Greater Than function $GT_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is defined as the comparison between the numbers $x,y \in \{0,\ldots,2^n-1\}$ defined by the input strings:*

$$GT_n(X,Y) = \begin{cases} 0 & \textit{if } \sum_{i=1}^n 2^{n-i}X_i \leq \sum_{i=1}^n 2^{n-i}Y_i \\ 1 & \textit{otherwise} \end{cases}$$

It is known that in the deterministic communication model (when errors are not allowed), the communication complexity of GT_n is $\Theta(n)$. When randomization is allowed, one can use a noisy variant of the Binary Search algorithm [FPRU94] to locate the first bit of disagreement between X and Y in $O(\log n)$ communication, thus solving the GT_n problem. This bound turns out to be tight: $\Omega(\log n)$ communication is necessary in general for solving the GT_n problem [Vio11, BW12].

We will be interested in the $\Omega(\log n)$-hardness of GT_n in the distributional setting, where inputs (X,Y) arrive according to a distribution μ, and the parties try to compute $GT(X,Y)$ for all but an ε-fraction of inputs. The hardness of the problem, of course, depends on μ. For example, if μ is just the uniform distribution on pairs of strings, then GT_n is very easy, since with high probability X and Y will differ in one of their first few bits.

For our purposes, it is useful to explicitly mention the following distribution $\mu_{GT,n}$ under which the communication complexity of GT_n is $\Theta(\log n)$. A pair (X,Y) is sampled as follows:

1. Sample an index $k \in \{1,\ldots,n\}$ uniformly at random.
2. Sample $z_1,\ldots,z_{k-1},w,x_{k+1},\ldots,x_n,y_{k+1},\ldots,y_n$ — uniformly random bits.
3. Let $X = z_1,\ldots,z_{k-1},w,x_{k+1},\ldots,x_n$, $Y = z_1,\ldots,z_{k-1},\overline{w},y_{k+1},\ldots,y_n$.

With the distribution $\mu_{GT,n}$ defined, we are ready to define the sampling problem $\mathbf{S}$ which we conjecture to separate information complexity from communication complexity.

3. The sampling problem

We start by defining the sampling problem $\mathbf{S}$. The problem will use strings over an alphabet Σ of size k, where k is a parameter. In addition, it will use a parameter $N = 2^n$. The problem makes sense for any $N = 2^n$ which is a power of 2, but for the rest of the paper we fix $N = 2^n = 2^{2^k}$.

Definition 4 *The sampling problem* $\mathbf{S} = \mathbf{S}(k,N)$*, where* $N = 2^n$ *is defined as follows.*

- *Alice and Bob are given a pair of numbers* $a,b \sim_{\mu_{GT,n}} \in \{0,\ldots,N-1\}$ *(i.e. two n-bit numbers that take* $\Omega(\log n)$ *communication to compare);*
- *Alice is given a uniformly random function* $F_A : \Sigma^{2a} \to \Sigma$ *(which is only known to Alice);*
- *Independently, Bob is given a uniformly random function* $F_B : \Sigma^{2b+1} \to \Sigma$*;*
- *Alice and Bob need to sample a uniformly random string* $s \in \Sigma^{2N}$ *subject to the constraints*
  1. $s_{2a+1} = F_A(s_{1..2a})$*; and*
  2. $s_{2b+2} = F_B(s_{1..2b+1})$.

In other word, s is sampled uniformly from the subset S of strings which satisfy the two constraints. The size of S, $|S| = k^{2N-2}$; therefore, the KL-divergence between s and the uniform distribution on Σ^{2N} is $2\log k$.

The naïve protocol π_0 for $\mathbf{S}$ proceeds in rounds, where in odd rounds Alice samples the next symbol of s, and in even rounds Bob does. In rounds $i \neq 2a+1$, Alice just sends a uniformly random $s_i \in_U \Sigma$. In round $i = 2a+1$, Alice computes and sends $s_i = F_A(s_{1..2a})$. This incurs the very high communication cost of $\Theta(N \log k) = \Theta(2^{2^k} \log k)$. However, we argue that the information cost of π_0 is only $2\log k$, which is tight. Denote by μ the distribution of the inputs a, F_a, b, F_b to $\mathbf{S}$.

Claim 5 $\mathsf{IC}_\mu(\pi_0) = \mathsf{IC}_\mu(\mathbf{S},0) = 2\log k.$

Proof. We first note that the transcript of π_0 is distributed exactly as the output s of $\mathbf{S}$ given a,F_a,b,F_b, and therefore

$$\mathsf{IC}_\mu(\mathbf{S},0) = \mathsf{IC}_\mu(\pi_0) = I_\mu(s;a,F_a,b,F_b).$$

Finally,

$$I_\mu(s;a,F_a,b,F_b) = \mathbf{E}\,\mathbb{D}(s|_{a,F_a,b,F_b} \| s) = 2\log k.$$

□

To separate information complexity from communication complexity for sampling problems, we need to show that $\mathsf{CC}_\mu^{average}(\mathbf{S},\varepsilon) = \omega(\log k)$ for some small constant ε. A general result in [Bra12b] implies that the gap between information complexity and communication complexity can be at most exponential. We conjecture, that $\mathbf{S}$ is, in fact, an example of such exponential separation:

Conjecture 6 *For a small constant* $\varepsilon > 0$*, such as* $\varepsilon = 1/10$*, the communication complexity*

$$\mathsf{CC}_\mu^{average}(\mathbf{S},\varepsilon) = k^{\Omega(1)} = 2^{\Omega(\mathsf{IC}_\mu(\mathbf{S},0))}.$$

It is important to re-emphasize that our communication model allows for public shared randomness. This difference is less important in the context of computation problems, but is significant for sampling problems, since without public randomness, even the simple task of sampling a uniformly random string from $\{0,1\}^n$ would require $\sim n$ bits of communication.

In the next section we will briefly discuss some obvious strategies for solving $\mathbf{S}$ using low communication, and see why these are consistent with Conjecture 6.

4. Discussion

We start by considering two communication protocols, both of which turn out to yield communication complexity of $O(k)$, although in different ways.

The first protocol, π_1, ignores the exact way in which s was generated. Instead, observe that a randomly selected string $t \in \Sigma^{2N}$ has a probability of exactly $1/k$ of being consistent with Alice's input (i.e. of satisfying $t_{2a+1} = F_A(t_{1..2a})$). Similarly, it has a probability of $1/k$ of being consistent with Bob's input, and a probability of $1/k^2$ of being consistent with both inputs. One strategy for sampling a consistent s is as follows:

1. Using public randomness and no communication, consider k^2 strings $s_1, \ldots, s_{k^2}$ drawn uniformly at random from Σ^{2N};

2. Let A be the subset (of approximately k) strings consistent with Alice's input, and let B be the subset consistent with Bob's input;

3. Alice and Bob communicate to determine whether $A \cap B = \emptyset$, if not, they output the first element of $A \cap B$; otherwise they repeat the entire process.

It is clear that the first string in the intersection between A and B is distributed according to the correct distribution of s, and therefore π_1 is correct. We need only to consider its expected communication cost. The probability that $A \cap B \neq \emptyset$ is approximately $1 - 1/e$, and the process will terminate after an expected constant number of iterations. The communication cost, therefore, is proportional to the cost of finding the intersection of A and B. The naïve solution to this problem (Alice sends Bob A, Bob returns $A \cap B$) thakes $O(k \log k)$ communication. With some work, and allowing for a small error, one can bring the communication cost down to $O(k)$ [HW07, BGPW13]. The communication complexity of this method, cannot, however, be reduced below $O(k)$, since deciding whether two sets of size $\sim k$ are disjoint (and finding a point of intersection) is known to require $\Omega(k)$ communication [KS92, Raz92].

The second protocol, π_2, attempts to exploit the structure of the distribution of s. Note that if Alice and Bob knew whether $a \leq b$ or $a > b$, and moreover, were given a number c that separates them (i.e. such that $a \leq c \leq b$ or $a \geq c \geq b$), then the communication complexity of the problem would be only $O(\log k)$:

1. Given a c such that $a \leq c \leq b$ or $a \geq c \geq b$, for concreteness assume that $a \leq c \leq b$;

2. Using public randomness and no communication, generate strings $t_1, t_2, \ldots \in_U \Sigma^{2c+1}$;

3. Alice sends Bob the index of the fist string s' which is consistent with F_A (i.e. $s'_{2a+1} = F_A(s_{1..2a})$);

4. Using public randomness and no communication, generate strings $r_1, r_2, \ldots \in_U \Sigma^{2N}$ which extend s' with uniformly random symbols;

5. Bob sends Alice the index of the fist string s which is consistent with F_B (i.e. $s_{2b+2} = F_B(s_{1..2b+1})$);

6. s is the output of the protocol.

It is again easy to see that this protocol outputs strings with the correct distribution. The only steps that use any communication are steps 3 and 5. Each of these involves sending the index of the first acceptable string. As discussed above, the probability of a string to be acceptable is $1/k$, and therefore communicating the index of the first acceptable string requires $O(\log k)$ bits.

Of course, the protocol described above requires Alice and Bob to find a number c between a and b. If a and b are $n = \log N$ bit numbers, finding such a c can be done in $O(\log n) = O(\log \log N)$ communication using a variant of Binary Search. Alice and Bob will exchange hashes of prefixes of a and b to find the first location of disagreement, and then use it to produce c. Notice that producing c is at least as hard as deciding whether $a > b$. Therefore, since our distribution of (a,b)'s is hard for the $GT(a,b)$ problem, protocol π_2 will require at least $\Omega(\log \log N)$ bits of communication. By our choice of $N = 2^{2^k}$, we see that π_2 is also a $\Theta(k)$-communication protocol.

Conclusion

The main outstanding open problem in this paper is proving or disproving Conjecture 6. Assuming the conjecture is true, it would give an example of a sampling problem for which the information complexity (and thus the amortized communication complexity) is much lower than the one-shot communication complexity.

It would be interesting to generalize our sampling problem to a decision problem. Alternatively, if no such decision problem exists, it would be interesting to understand what property makes protocols for decision problems easier to compress. One possibility is that in protocols for decision problems the answer is determined by messages sent by Alice and Bob. In contrast, sampling problems often require the knowledge of public randomness and/or one of the parties' inputs to compute the output.

Acknowledgments

I would like to thank Boaz Barak, Ankit Garg, Denis Pankratov, Anup Rao, and Omri Weinstein for enlightening discussions that helped refine the sampling problem presented in this paper.

References

[BGPW13] Mark Braverman, Ankit Garg, Denis Pankratov, and Omri Weinstein. From information to exact communication. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 151–160. ACM, 2013.

[BR11] Mark Braverman and Anup Rao. Information equals amortized communication. In *FOCS*, pages 748–757, 2011.

[Bra12a] Mark Braverman. Coding for interactive computation: progress and challenges. In *50th Annual Allerton Conference on Communication, Control, and Computing*, 2012.

[Bra12b] Mark Braverman. Interactive information complexity. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 505–524, New York, NY, USA, 2012. ACM.

[BW12] Mark Braverman and Omri Weinstein. A discrepancy lower bound for information complexity. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 459–470. Springer, 2012.

[FPRU94] Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

[HW07] Johan Håstad and Avi Wigderson. The randomized communication complexity of set disjointness. *Theory of Computing*, 3(1):211–219, 2007.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, Cambridge, 1997.

[KS92] Bala Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, November 1992.

[Raz92] Razborov. On the distributed complexity of disjointness. *TCS: Theoretical Computer Science*, 106, 1992.

[Vio11] Emanuele Viola. The communication complexity of addition. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 18, page 152, 2011.