Communication and information complexity

Mark Braverman

Abstract

Communication complexity is an area of computational complexity theory that studies the amount of communication required to complete a computational task. Communication complexity gives us some of the most successful techniques for proving impossibility results for computational tasks.

Information complexity connects communication complexity with Shannon's classical information theory. It treats information revealed or transmitted as the resource to be conserved. On the one hand, information complexity leads to extensions of classical information and coding theory to interactive scenarios. On the other hand, it provides us with tools to answer open questions about communication complexity and related areas.

This note gives an overview of communication complexity and some recent developments in two-party information complexity and applications. The note is based on a talk given by the author at the International Congress of Mathematicians in 2022. It expands on some of the themes from the talk. It also provides references that were omitted during the talk. This is a preliminary version. An updated final version will appear after the ICM.

Mathematics Subject Classification 2020

Primary 68Q11; Secondary 68P30,94A15

Keywords

computational complexity, communication complexity, information theory, information complexity, interactive computation

INTERNATIONAL CONGRESS
OF MATHEMATICIANS
2022 JULY 6–14

Preliminary version, to appear in Proc. Int. Cong. Math. 2022, Vol. 1.
DOI 10.4171/ICM2022/208

1. Computational complexity theory

Computational complexity theory is concerned with modeling, understanding and mapping out the computational resources needed to solve various problems involving manipulation of information. Below we give a brief non-technical overview to set communication and information complexity in context. A principled and extensive treatment of the area can be found in texts such as [2,59].

1.1. Upper and lower bounds

Mapping out the limits of computation involves a combination of upper and lower bounds on the amount of resources being studied.

An *upper bound* is typically an algorithm with some provable properties. The primary goal of such an algorithm is to place a problem in a complexity class. Sometimes such an algorithm is practically useful, or may inspire a practically useful version later on.

For example, the problem of sorting n elements can be solved using $O(n \log n)$ comparisons. This upper bound can be established via the *MergeSort* algorithm [92], which is fairly straightforward to analyze. In practice, the *QuickSort* algorithm often performs better, but it is harder to analyze for the purposes of establishing an upper bound.

Some upper bounds have desirable properties, but are clearly not the most "practical" algorithms for the problem. For example, using recursion, one can show that the problem of raising an $n \times n$ matrix $A \in \mathbb{F}_2^{n\times n}$ to the power n can be done using $O(\log^2 n)$ bits of working memory. But alas, the resulting algorithm would run in $n^{\Theta(\log n)}$ steps of computation, and would be impractical. Beyond its theoretical value, the upper bound placing Matrix Powering in $\mathsf{SPACE}(\log^2 n)$[1] gives us a hint that basic linear algebraic operations may be amenable to parallelization — a direction that has seen a lot of work in practice [29,36].

A *lower bound* involves a proof that some computational task is impossible to accomplish within a given constraint on resources. Lower bounds are often harder to prove than upper bounds, since upper bounds are constructive (providing an algorithm), while to prove a lower bound one needs to rule out *all* possible algorithms for a given problem. Still, in many situations provable lower bounds are possible. As we shall see, in many more situations lower bounds can be proved assuming a plausible conjecture, such as $\mathsf{P} \neq \mathsf{NP}$.

In the sorting problem mentioned earlier, the algorithm may output one of $n!$ possible orderings. Each comparison rules out at most half of the orderings. Therefore, at least $\log_2 n! = \Omega(n \log n)$ comparisons are needed to sort n elements[2]. Thus, the upper bound given by *MergeSort* is asymptotically optimal, and sorting requires $\Theta(n \log n)$ comparisons.

1.2. Abstraction and complexity classes

Abstraction is one of the two core ideas underpinning much of complexity theory, it allows us to develop models of computation.

[1] In fact, it is in the slightly smaller complexity class $\mathbf{NC}^2$ [34]

[2] This simple argument shows that $\Omega(n \log n)$ comparisons are needed in the worst case, but it is not hard to show that the same bound also holds *on average*

There are many different mathematical models that focus on different aspects of computation. Problems can then be grouped into *complexity classes* based on the amount of resources required to solve the problem in a given model. There are hundreds of complexity classes that have been studied explicitly.

The *Turing Machine* is an example of an early and very successful abstraction introduced in the 1930s [99]. It gave a mathematical definition of computation, which is still accepted today. In modern terms, a Turing Machine is equivalent to a standard computer with unlimited (but finite at any point during the computation) memory. The class R of problems corresponds to problems solvable by a Turing Machine. Problems inside R are said to be "computable" and problems outside R are "non-computable"[3].

The taxonomy of computable vs. non-computable is a very coarse one. For example, the tasks of adding two n-bit numbers, breaking an n-bit cartographic cipher, or simulating the n-body problem for 2^n time steps are all computable tasks, yet clearly some are more "tractable" than others. Such observations were the starting point for defining more complexity classes based on the setting and resources being constrained.

One plausible (and robust) definition of tractability is given by the class P — the class of problems that are solvable by a Turing Machine in time polynomial in input size. For example, a problem on graphs $G = (V, E)$ with n vertices and m edges is in P if it can be solved by a Turing Machine running in time at most n^c for some constant c. The class P abstracts enough details that we do not need to be concerned with the exact model of the Turing Machine[4]. We also do not need to worry about dependence on the number of edges m, since $m < n^2$, and any bound polynomial in n is also polynomial in m.

The class P abstracts away many details, yet it still gives a very useful definition of tractability. It is especially useful in its negation — if a problem is (for a "typical" input) $\notin \mathsf{P}$, then it is likely intractable in practice except on very small inputs. In its positive direction, being in P does not guarantee that the problem is "easy". For example, checking whether a graph G contains a clique K_{100} with a 100 nodes is easily seen to be in P, yet no general algorithm for the problem that runs substantially[5] faster than checking all n^{100} possible vertex sets is known, and is suspected to not exist [33].

The class P can be further refined by restricting the running time of the Turing Machine. For example the class $\mathsf{DTIME(n)}$ restricts the number of steps to be linear in input length. In the case of graphs, this would be linear in $n + m$ – the total number of vertices and edges. Note that here we need to be more careful about the memory access model – linear-time algorithms are typically allowed random memory access. The field of *fine-grained complexity* aims to classify problems within P based on their required running time, using plausible assumptions [103].

[3] Formally, the class R contains decision problems, that are called "recursive" or "decidable". To simplify our current discussion, we blur the distinction between decision problems and general computation tasks.

[4] For example, whether data is stored on a tape and addressed sequentially or in a random-access memory array

[5] Here "substantially" means $n^{o(k)}$, where $k = 100$ is the size of the clique we are looking for

It is possible to reduce the allotted time even further and talk about sub-linear time algorithms. Those are particularly important in databases and other "big-data" applications, where one wants to maintain a data-structure, but to answer queries about it without reading it in its entirety.

Computation time (given by the number of steps performed) is only one of many resources one could consider. Other resources commonly considered include memory used (to store the algorithm's data), parallelization (e.g. is there an algorithm that can be completed in a very short amount of time in a parallel computer), whether the algorithms uses randomness (and how much), and latency caused by communication if the computation is a distributed one[6]. Specific applications (such as data structures) feature additional parameters, as one needs to consider the cost of updating the data structure and the cost of querying it. In addition to "physical" resources used by an algorithm, there are sometimes additional desirable properties such as fault-tolerance or privacy-preservation. These additional requirements may interact with the resource constraints (typically by making them harder to satisfy).

Given the long (but still partial!) list of possible resources and resource combinations to consider, it should not be a surprise that there are so many complexity classes! In fact, it may be surprising that classifying algorithmic problems into complexity classes has been such a productive enterprise at all. One possible explanation of this is that reductions — which we will discuss next — allow us to "cull" classes by showing equivalences. These equivalences are often non-trivial and very surprising.

1.3. Reductions and conditional lower bounds

Reduction is the other core idea in computational complexity theory. As discussed earlier, it is easier to prove that a computational task is attainable within given constraints (by demonstrating an algorithm) than to prove that it is unattainable (need to rule out all algorithms). A reduction allows to turn algorithms into (conditional or unconditional) lower bounds.

Let C be a complexity class (i.e. a class of problems solvable within some given resource constraints). For two problems A and B, one can often use an *algorithmic* construction to prove a statement of the form

$$A \in \mathsf{C} \Rightarrow B \in \mathsf{C}. \tag{1.1}$$

For example, if $\mathsf{C} = \mathsf{P}$, all one needs to do is to construct a polynomial-time algorithm that uses a black box for solving A[7] in order to solve B.

Relationship (1.1) is often denoted by $B \leq_{\mathsf{C}} A$, where the reduction from A to B is done using an algorithm from class C. Thus, for example, if $A \in \mathsf{P}$ and $B \leq_{\mathsf{P}} A$, then $B \in \mathsf{P}$.

Taking the contrapositive of (1.1), we get

$$B \notin \mathsf{C} \Rightarrow A \notin \mathsf{C} \tag{1.2}$$

[6] In practice, latency of communication between processing cores is significantly slower than computation within a core

[7] The box is also assumed to run in polynomial time

Thus a lower bound on B translates into a lower bound on A.

In terms of theory building, (1.1) and (1.2) allow one to consolidate problems into complexity classes. As it turns out, many natural complexity classes C have a complete problem P_{C} such that all $A \in \mathsf{C}$ are reducible to P_{C}. For example, Cook-Levin's theorem asserts that boolean circuit satisfiability SAT is complete for the class NP. This means that for any problem $B \in \mathsf{NP}$, $B \leq_{\mathsf{P}} SAT$. Therefore, if $\mathsf{P} \neq \mathsf{NP}$, then for some $B \in \mathsf{NP}$, $B \notin \mathsf{P}$, and by (1.2) $SAT \notin \mathsf{P}$.

In practice, whenever the assumption $\mathsf{P} \neq \mathsf{NP}$ is made, what is actually used is the assumption that $SAT \notin \mathsf{P}$. Under this assumption, to show that $A \notin \mathsf{P}$ it is enough to show that $SAT \leq_{\mathsf{P}} A$. The latter is an algorithmic problem. It may be a simple algorithm taught in introductory classes — such as the reduction [63] showing that

$$SAT \leq_{\mathsf{P}} [\text{deciding whether a given graph } G \text{ is 3-colorable.}]$$

Or, it could be the result of stacking extremely complex reductions, such as optimal inapproximability of 3-SAT — one of the crowning achievements of the Probabilistically Checkable Proofs (PCP) program [3,4]. In either case, the result is ultimately algorithmic — unspooling the reduction would yield an algorithm that, given a black-box access to the problem being proven to be hard, solves SAT in polynomial time.

Reductions are very useful in consolidating complexity classes. Suppose that C_1 and C_2 are two complexity classes with complete problems $P_1 \in \mathsf{C}_1$ and $P_2 \in \mathsf{C}_2$[8]. Then to show that $\mathsf{C}_1 = \mathsf{C}_2$ it suffices to show that $P_1 \in \mathsf{C}_2$ and that $P_2 \in \mathsf{C}_1$ — again, solving two algorithmic problems.

The logic of (1.2) is very powerful in practice, as it allows one to maintain a list of reasonable hardness assumptions, and to prove tight lower bounds modulo these assumptions. Proving those assumptions may be out of reach (proving $\mathsf{P} \neq \mathsf{NP}$ appears to be currently out of reach). The assumptions may even be false, but nonetheless they can be useful in practice! An example of such an assumption is the Strong Exponential Time Hypothesis (SETH) — asserting that certain flavors of SAT on n variables cannot be solved in $2^{0.999n}$ computation steps[9]. There is a fair chance that the SETH assumption is false, although it has been open for about two decades [56,57]. Still, if someone works on an applied data structure, and designing a faster-than-trivial solution for the problem leads to a violation of SETH, the practical implication is that the algorithm designers may assume that there is no better solution than the trivial one (at least for now) — and focus their efforts on other aspects of the design.

In summary, the majority of results in complexity theory — even the deepest and most important ones — are algorithmic reductions, establishing connections between problems and between complexity classes. Most hardness results are conditional ones, using reasonable assumptions such as $\mathsf{P} \neq \mathsf{NP}$, or more ambitious assumptions, such as the SETH.

[8] Technically, the complexity classes also need to be closed under appropriate reductions, but this is rarely a problem

[9] Note that a brute-force search over all possible assignments takes time $2^n \cdot n^{O(1)}$

From the perspective of most engineering applications (with the notable exception of cryptography) this is good enough — one can accept a reasonably-aged conjecture as evidence of computational hardness. Computational complexity theory has thus been spectacularly successful in classifying problems into hardness classes based on conjectures. Proving those conjectures is a different matter altogether — progress in attaining *unconditional* lower bounds — ones where we do not have the luxury of reductions — has been very slow. Devising new attack routes and advancing existing ones is therefore a major challenge in attaining mathematical understanding of computation.

1.4. Unconditional lower bounds: some attack routes

The most general technique for proving unconditional lower bounds on computation is through *diagonalization*. The very first result in the theory of computation [99] used diagonalization to show that the Halting Problem is non-computable. The Halting Problem asks, given a computer program and an input[10], to decide whether the program eventually terminates, or runs indefinitely. The proof of the non-computability of the Halting Problem is straightforward (assuming one accepts that one can program a compiler that takes an encoding of a program and executes it). It is similar to Cantor's proof that there is an uncountable number of Real numbers. Many (perhaps most) proofs of non-computability results work through a reduction to the Halting Problem.

Diagonalization is useful not just for proving non-computability, but for proving hierarchy theorems, stating that giving programs asymptotically more time strictly increases the set of problems that can be solved. For example[11]

$$\mathsf{DTIME}(n^{2.3}) \subsetneq \mathsf{DTIME}(n^{2.4})$$

Still, there are reasons (namely "relativization" [6]) to believe that diagonalization cannot unconditionally prove results such as $\mathsf{P} \neq \mathsf{NP}$, and other currently open unconditional lower bounds.

For unconditional lower bounds that are most likely true but do not follow from diagonalization (such as $\mathsf{P} \neq \mathsf{NP}$), one would need a different set of lower bounds strategies. One approach to take is an incremental one: design an hierarchy of results of increasing difficulty, and incrementally prove them — hopefully discovering and developing new techniques in the process. As an added benefit, even the partial results can be used independently. As discussed earlier, through the magic of reductions, one unconditional lower bound can be converted into many interesting results across multiple settings.

Historically, the most prominent such hierarchy has been that of circuit complexity classes — it is not the main topic of this note, and therefore we will only review it briefly. The circuit complexity program has the advantage that strong enough results under the program will immediately lead to strong lower bounds for complexity classes. One drawback of the

[10] A Turing Machine in the original formulation

[11] Perhaps not surprisingly, the task that is easy to perform is time $n^{2.4}$ and impossible to perform in time $n^{2.3}$ is simulating a Turing Machine for $n^{2.35}$ time steps.

program is that progress in the last 30 years has been slow, and it is unclear at this point what tools would be needed to make further progress.

Other than the circuit complexity program, two additional programs of note are arithmetic circuit complexity [96], and communication complexity. Discussing arithmetic circuit complexity is beyond the scope of this note. Communication complexity is going to be our main focus, and will be discussed in some detail.

Boolean circuit complexity program. A circuit is a directed acyclic graph with edges carrying boolean signals 0 or 1. Nodes with no incoming edges correspond to input variables. Other nodes correspond to gates. A gate computes a boolean function of the values of edges incoming into the node, and places the result on the outgoing edges.

Gates may be of fan-in 2, or of unbounded fan-in[12]. Bounded fan-in gates are typically **OR**, **AND**, and **NOT**. Unbounded fain-in gates may be computing an **OR** or **AND** of their inputs, or a more complicated function. Two particularly important functions are summation modulo k: $\oplus_{\mathbf{k}}(x_{1..m}) = \mathbf{1}_{\sum x_i \equiv 0 \pmod{k}}$ and majority $\mathbf{MAJ}(x_{1..m}) := \mathbf{1}_{\sum x_i \geq m/2}$.

A function computable in polynomial time by a Turing Machine can also be computed by a polynomial-size circuit[13]. The class of polynomial-size circuits is denoted by $\mathsf{P/poly}$. As we have just noted, $\mathsf{P} \subset \mathsf{P/poly}$, proving that $SAT \notin \mathsf{P/poly}$ would imply $\mathsf{P} \neq \mathsf{NP}$.

Within circuit complexity, the most natural hierarchy within $\mathsf{P/poly}$ is based on *circuit depth*: the largest number of gates from an input to the output of the circuit. Note that if each gate takes 1 time unit to evaluate, then circuit depth corresponds to the (parallel) latency needed to evaluate the circuit.

When only fan-in-2 gates are allowed, the class NC^i denotes the set of functions that can be evaluated by a circuit of depth $O((\log n)^i)$. When unbounded fan-in **OR** and **AND** gates are allowed, AC^i denotes the set of functions that can be evaluated by a circuit of depth $O((\log n)^i)$. When in addition $\oplus_{\mathbf{k}}$ gates are allowed for some constant k, we get the class denoted by ${\mathsf{AC}_{\oplus\mathsf{k}}}^i$. When the majority gate **MAJ** is allowed, we get the class denoted by TC^i.

A majority gate with $n^{O(1)}$ many inputs can be computed by a depth-$O(\log n)$ boolean circuit with fan-in-2 gates. This gives us the following chain of inclusions:

$$\mathsf{NC}^0 \subseteq \mathsf{AC}^0 \subseteq \mathsf{AC}^0_{\oplus\mathsf{k}} \subseteq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{AC}^1 \subseteq \mathsf{NC}^2 \subseteq \mathsf{P/poly} \tag{1.3}$$

Recall that the program was to progressively prove lower bounds against circuit classes in (1.3), eventually building up to $SAT \notin \mathsf{P/poly}$.

[12] "Fan-in" here is the number of inputs a gate can take. A fan-in-2 **AND** gate takes two inputs $x_1, x_2 \in \{0, 1\}$ and outputs $x_1 \wedge x_2$. A fan-in-n **AND** gate takes n inputs $x_1, \ldots, x_n \in \{0, 1\}$ and outputs $x_1 \wedge x_2 \wedge \ldots \wedge x_n$. A fan-in-$n$ **AND** gate can be computed by a depth-$(\log n)$-binary tree of fan-in-2 **AND** gates.

[13] Roughly, in the circuit, each wire corresponds to the state of one bit of memory at a particular point of time in the computation. This reduction is used in designing ASIC circuits that need to be particularly fast or energy-efficient in performing a particular calculation, such as for cryptographic attacks or for routing internet traffic.

The class NC^0 contains circuits where the depth is constant (since $O((\log n)^0) = O(1)$), and each gate's fan-in is 2. Therefore, the n-bit **AND**, $\mathbf{AND}_n$ cannot be computed in NC^0, making the first inclusion strict.

Making the second inclusion in (1.3) strict already requires significant effort. A progression of results in the 1980s showed that an AC^0 circuit computing the parity $\oplus_\mathbf{2}$ of n variables has to be of size exponential in n [42, 51]. The proof is combinatorial in nature, using the fact that a random restriction of the parity function to a subset of its coordinates yields another parity function. This line of work led to important results in boolean function analysis — showing that functions computed by AC^0 circuits are approximated by low-degree polynomials in Fourier space [71]. Still, these techniques do not appear to lead to any lower bounds against $\mathsf{AC}^0_{\oplus 2}$ — the class of constant depth circuits with unbounded fan-in **OR**, **AND**, and $\oplus_\mathbf{2}$ parity gates.

Lower bounds against $\mathsf{AC}^0_{\oplus 2}$ (or $\mathsf{AC}^0_{\oplus p}$ for an arbitrary constant prime p) — given by Razborov and Smolensky [89, 97] also in the 1980s — require yet another set of ideas, this time algebraic. It turns out that a function computable by a polynomial-size $\mathsf{AC}^0_{\oplus p}$ circuit can be approximated by a low-degree polynomial over the field $\mathbb{F}_p$ (note that $\oplus_\mathbf{p}$ become simple addition over $\mathbb{F}_p$). A dimensionality/counting argument then shows that computing $\oplus_\mathbf{q}$ for any other prime $q \neq p$ cannot be done in $\mathsf{AC}^0_{\oplus p}$. These results only hold for primes. In particular, it is strongly believed, but not known, that $\oplus_\mathbf{5}$ cannot be computed by a polynomial sized circuit in $\mathsf{AC}^0_{\oplus 6}$.

As of late 1980s, diagram (1.3) appears as

$$\mathsf{NC}^0 \subsetneq \mathsf{AC}^0 \subsetneq \mathsf{AC}^0_{\oplus p} \subsetneq \mathsf{AC}^0_{\oplus k} \subseteq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{AC}^1 \subseteq \mathsf{NC}^2 \subseteq \mathsf{P/poly} \tag{1.4}$$

Since then, there has been no progress in diagram (1.4). There are several possible explanations for this. One possible explanation is that we are underestimating the power of TC^0 circuits, and that some of the inclusions are in fact not strict (or, at the very least, lower bounds against TC^0 are not much easier than general circuit lower bounds). There is some indirect evidence for the power of TC^0. Within circuit complexity, one surprising result about TC^0 is that it is capable of computing the Chinese Remainder representation of n-bit integers, leading to additional surprising upper bounds [53]. More informally, TC^0-circuits are able to represent artificial neural nets, which have shown a surprising degree of expressiveness in practice, providing indirect evidence for the computational power of the class.

Another possible reason for the relative lack of progress of the circuit complexity program is that the techniques involved appear to be related to logic and combinatorics (diagonalization is a logic technique, while most existing lower bounds are combinatorial), and that new connections are needed to make progress on this programs (or to obtain unconditional lower bounds in another way). This is something that has been noted very early in the study of theoretical computer science (and what became complexity theory). The following is a quote from John von Neumann [101]:

> *"There exists today a very elaborate system of formal logic, and, specifically, of logic as applied to mathematics. This is a discipline with many good sides, but also with certain serious weaknesses. This is not the occasion to enlarge upon the good sides, which I have certainly no intention to belittle. About the inadequacies, however, this may be said: Everybody who has worked in formal logic will confirm that it is one of the technically most refractory parts of mathematics. The reason for this is that it deals with rigid, all-or-none concepts, and has very little contact with the continuous concept of the real or of the complex number, that is, with mathematical analysis. Yet analysis is the technically most successful and best-elaborated part of mathematics. Thus formal logic is, by the nature of its approach, cut off from the best cultivated portions of mathematics, and forced onto the most difficult part of the mathematical terrain, into combinatorics.*
>
> *The theory of automata, of the digital, all-or-none type, as discussed up to now, is certainly a chapter in formal logic. It would, therefore, seem that it will have to share this unattractive property of formal logic. It will have to be, from the mathematical point of view, combinatorial rather than analytical."*

In the 70+ years since this quote, analysis has played an increasing role in both lower and upper bounds. Boolean function analysis [79] is an example of a relatively new field that has played a critical role in lower-bound reductions (in Probabilistically Checkable Proofs and Unique Games), and in upper bounds (for example in learning theory). Ideas from convex optimization (some dating back to von Neumann and his colleagues) are now used extensively in upper bounds for such "discrete" problems as Max-Flow [75]. Still, more "analytic" concepts of complexity, particularly ones that tensorize[14] are always helpful in moving the field forward. Communication complexity, and especially its subarea of information complexity, fit well within this general thrust.

Communication complexity and unconditional lower bounds. Like many other concepts within computational complexity theory, communication complexity has been primarily developed as an abstraction of concrete computational problems. Within theoretical computer science the model was introduced in 1979 by Yao in [105]. Communication complexity arises naturally when studying the complexity of distributed computing, where oftentimes the delay cost of communication between nodes dominates the computational cost within the nodes.

Communication complexity theory has been very successful at producing unconditional lower bound results. Early results used combinatorial methods, but more recently analytical and information-theoretic methods (which are also continuous and analytical in many ways) have shown some success. As with circuit complexity, it is possible to construct a hierarchy of various communication complexity classes, although the hierarchy requires

[14] In complexity theory, tensorization is known as direct sum and direct product properties, which we discuss later in this note.

more formalism to define, so we will omit it here. Specific parameters affecting a particular model of communication include the number of players (2 or more), the number of rounds of back-and-forth communication, whether randomness and errors are allowed, etc.

It should be noted that some of the most promising approaches to unconditional circuit lower bounds go through communication complexity. A notable example is Karchmer-Wigderson games [61, 62] — a particular type of two-party deterministic communication complexity models for which a lower bound would give a lower bound against NC^1 circuits from (1.4). Another example is an implication of a result of Beigel and Tauri [10] about $\mathsf{AC}^0_{\oplus k}$ circuits, that certain multi-party communication lower bounds imply lower bounds for such circuits. Of course, to achieve these lower bounds further technical progress is needed in communication complexity lower bounds. We will return to this briefly at the end of this note.

1.5. Shannon's information theory and one-way communication

Note: *Large parts of this section (as well as the next two) were previously presented in the note* [12] *by the author accompanying the talk at ICM 2014 in Seoul.*

We begin with a very high-level overview of Shannon's information theory. We do this for two reasons. The first reason is that we will need its formalism when defining information complexity in Section 3. The second reason is that one-way information and coding theory is an example of a successful theory that gives very precise answers to many natural questions about data transmission. It serves as a kind of inspiration for what a theory of communication complexity (or even computational complexity) could aspire to — even if it turns out that some core aspects of this program cannot be extended to interactive or multi-party settings.

Information and coding theory is an enormous field of study, with subareas dealing with questions ranging from foundations of probability and statistics to applied wireless transmission systems. We will focus only on some of the very basic foundational aspects, which were set forth by Shannon in the late 1940s, or shortly after.

While our overview of information and coding theory in this section focuses on fairly simple facts, we present those in some detail nonetheless, as they will be used as a scaffold for the interactive coding discussion. A thorough introduction into modern information theory is given in [35].

Noiseless coding. Classical information theory studies the setting where one terminal (Alice) wants to transmit information over a channel to another terminal (Bob). Two of the most important original contributions by Shannon are the *Noiseless Coding* (or Source Coding) Theorem and the *Noisy Coding* (or Channel Coding) Theorem. Here we will only focus on the noiseless part of the theory. The Source Coding Theorem asserts that the cost of Alice transmitting n i.i.d. copies of a discrete random variable X to Bob *over a noiseless* *binary channel*[15] scales as Shannon's entropy $H(X)$ as $n \to \infty$[16]:

$$H(X) = \sum_{x \in \mathrm{supp}(X)} \Pr[X = x] \log \frac{1}{\Pr[X = x]}. \tag{1.5}$$

If we denote by X^n the concatenation of n independent samples from X, and by $C(Y)$ the (expected) number of bits needed for Alice to transmit a sample of random variable Y to Bob, then the Source Coding Theorem asserts that[17]

$$\lim_{n\to\infty} \frac{C(X^n)}{n} = H(X). \tag{1.6}$$

This fact can be viewed as the operational definition of entropy, i.e. one that is grounded in reality. Whereas definition (1.5) may appear artificial, (1.6) implies that it is the right one, since it connects to the "natural" quantity $C(X^n)$. Another indirect piece of evidence indicating that $H(X)$ is a natural quantity is its additivity property:

$$H(X^n) = n \cdot H(X), \tag{1.7}$$

and more generally, if XY is the concatenation of random variables X and Y, then

$$H(XY) = H(X) + H(Y) \tag{1.8}$$

whenever X and Y are independent. Note that it is not hard to see that (1.7) and (1.8) fail to hold for $C(X)$, making $H(X)$ a "nicer" quantity to deal with than $C(X)$. Huffman coding (1.9) below blurs the distinction between the two, as they only differ by at most one additive bit, but we will return to it later in the analogous distinction between communication complexity and information complexity.

For noiseless coding in the one-way regime, it turns out that while $H(X)$ does not exactly equal the expected number of bits $C(X)$ needed to transmit a *single* sample from X, it is very close to it. For example, the classical Huffman's coding [55] implies that

$$H(X) \le C(X) < H(X) + 1, \tag{1.9}$$

where the "hard" direction of (1.9) is the upper bound. The upper bound showing that $C(X) < H(X) + 1$ is a *compression result*, showing how to encode a message with low average information content (i.e. entropy) into a message with a low communication cost (i.e. number of bits in the transmission). Note that this result is much less "clean" than the limit result (1.6): in the amortized case the equality is exact, while in the one-shot case a gap is created. This gap is inevitable if only for integrality reasons, but as we will see later, it becomes crucial in the interactive case.

Beyond giving the exact answer to the source coding question (equation (1.6)), Shannon's theory has two important benefits. First, it turns "communication" into a continuous

[15] A noiseless binary channel allows the sender to transmit to a receiver a single bit without error at a unit cost

[16] All logs in this paper are base-2, with ln denoting the natural logarithm

[17] In fact, Shannon's Source Coding Theorem asserts that due to concentration the *worst case* communication cost scales as $H(X)$ as well, if we allow negligible error. We ignore this stronger statement at the present level of abstraction.

resource — much more analytical than combinatorial. This is even more pronounced in the *Noisy Channel Coding* theorem, which allows one to denominate the capacity of a communication channel in bits of information, and to separate the ability of the channel to carry communication from the content of that communication.

Second, it gives us a powerful formalism for talking about information relationships between random variables, which naturally translate informal statements into mathematical expressions. We will give a brief exposition here of notions that we will use in Section 3.

For a single random variable X, entropy $H(X)$ gives a way to quantify the inherent uncertainty in the draw of this variable. For a pair of random variables X and Y, the conditional entropy $H(X|Y)$ can be thought of as the amount of uncertainty remaining in X for someone who knows Y:

$$H(X|Y) := H(XY) - H(Y) = \mathbf{E}_{y\sim Y} H(X|Y = y). \tag{1.10}$$

In the extreme case where X and Y are independent, we have $H(X|Y) = H(X)$. In the other extreme, when $X = Y$, we have $H(X|X) = 0$. The *mutual information* $I(X;Y)$ between two variables X and Y measures the amount of information that revealing Y reveals about X, i.e. the reduction in X's entropy as a result of conditioning on Y. Thus

$$I(X;Y) := H(X) - H(X|Y) = H(X) + H(Y) - H(XY) = I(Y;X). \tag{1.11}$$

Conditional mutual information is defined similarly to conditional entropy:

$$I(X;Y|Z) := H(X|Z) - H(X|YZ) = I(Y;X|Z). \tag{1.12}$$

The expression $I(X;Y|Z)$ is translated into English as "the (expected) amount of information learning variable Y reveals about X to someone who already knows Z".

A very important property of conditional mutual information is the *chain rule*:

$$I(XY;Z|W) = I(X;Z|W) + I(Y;Z|WX) = I(Y;Z|W) + I(X;Z|WY). \tag{1.13}$$

Again, an informal interpretation of (1.13) is that XY reveal about Z what X reveals about Z, plus what Y reveals about Z to someone who already knows X.

2. Communication complexity

For the majority of this discussion we will focus on 2-party computation, returning to the general case at the end of the note.

Communication complexity was introduced by Yao in [105], and is the subject of the texts [68,85]. It has found numerous applications for unconditional lower bounds in a variety of models of computation, including Turing machines, streaming, sketching, data structure lower bounds, and VLSI layout, to name a few. In the basic (two-party) setup, the two parties Alice and Bob are given inputs $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, respectively, and are required to compute a function $F(X,Y)$ of these inputs (i.e. both parties should know the answer at the end of the communication), while communicating over a noiseless binary channel (sending $0/1$ bits to each other). The parties are computationally unbounded, and their only goal is to minimize the number of bits transmitted in the process of computing $F(X,Y)$.

In a typical setup, F is a function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. Examples of functions commonly discussed and used include the Equality function

$$EQ_n(X,Y) := \mathbf{1}_{X=Y}(X,Y) = \bigwedge_{i=1}^{n} ((X_i \wedge Y_i) \vee (\neg X_i \wedge \neg Y_i)), \tag{2.1}$$

and the Disjointness function

$$Disj_n(X,Y) := \bigwedge_{i=1}^{n} (\neg X_i \vee \neg Y_i). \tag{2.2}$$

The basic notion in communication complexity is the *communication protocol*. A communication protocol over a binary channel formalizes a conversation, where each message only depends on the input to the speaker and the conversation so far:

Definition 2.1. A (deterministic) protocol π for $F : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ is defined as a finite rooted binary tree, whose nodes correspond to partial communication transcripts, such that the two edges coming out of each vertex are labeled with a 0 and 1. Each leaf ℓ is labeled by an output value $f_\ell \in \{0,1\}$. Each internal node v is labeled by a player's name and either by a function $a_v : \mathcal{X} \to \{0,1\}$, or $b_v : \mathcal{Y} \to \{0,1\}$ corresponding to the next message of Alice or Bob, respectively.

The protocol $\pi(X,Y)$ is executed on a pair of inputs (X,Y) by starting from the root of the tree. At each internal node labeled by a_v the protocol follows the child $a_v(X)$ (corresponding to Alice sending a message), and similarly at each internal node labeled by b_v the protocol follows $b_v(Y)$. When a leaf ℓ is reached the protocol outputs f_ℓ.

By a slight abuse of notation, $\pi(X,Y)$ will denote both the transcript and the output of the protocol; which one it is will be clear from the context. The communication cost of a protocol is the depth of the corresponding protocol tree. A protocol *succeeds* on input (X,Y) if $\pi(X,Y) = F(X,Y)$. Its communication cost on this pair of inputs is the depth of the leaf reached by the execution. The *communication complexity* $CC(F)$ of a function F is the lowest attainable communication cost of a protocol that successfully computes F. In the case of deterministic communication we require the protocol to succeed on all inputs.

A deterministic communication protocol π induces a partition of the input space $\mathcal{X} \times \mathcal{Y}$ into sets S_ℓ by the leaf ℓ that $\pi(X,Y)$ reaches. Since at each step the next move of the protocol depends only on either X or Y alone, each S_ℓ is a combinatorial rectangle of the form $S_\ell = S_\ell^{\mathcal{X}} \times S_\ell^{\mathcal{Y}}$. This key combinatorial property is at the heart of many combinatorial communication complexity lower bounds. To give an example of such a simple combinatorial proof, consider the *rank* bound. Let $N = |\mathcal{X}|$, $M = |\mathcal{Y}|$, and consider the $N \times M$ matrix M_F over $\mathbb{R}$ whose (X,Y)-th entry is $F(X,Y)$. Each protocol π with leaf set $\mathcal{L}$ of size L, induces a partition of $\mathcal{X} \times \mathcal{Y}$ into combinatorial rectangles $\{S_\ell\}_{\ell \in \mathcal{L}}$. Let M_ℓ be the matrix whose entries are equal to $M_{X,Y}$ for $(X,Y) \in S_\ell$ and are 0 elsewhere. Since $\{S_\ell\}_{\ell \in \mathcal{L}}$ is a partition of $\mathcal{X} \times \mathcal{Y}$, we have $M_F = \sum_{\ell \in \mathcal{L}} M_\ell$. Assuming π is always correct, each M_ℓ is *monochromatic*, i.e. either all-0, or all-1 on S_ℓ, depending on the value of f_ℓ. Thus, $\text{rank}(M_\ell) \le 1$, and

$$\text{rank}(M_F) \le \sum_{\ell \in \mathcal{L}} \text{rank}(M_\ell) \le L. \tag{2.3}$$

In fact, a stronger bound of $L-1$ holds unless M_F is the trivial all-1 matrix. Thus any protocol computing F must have a communication cost of at least $\log(\text{rank}(M_F)+1)$, and it follows that the communication complexity of F is at least $\log(\text{rank}(M_F)+1)$. As an example of an application, if $F = EQ_n$ is the Equality function, then $M_{EQ_n} = I_{2^n}$ is the identity matrix, and thus $CC(EQ_n) \geq n+1$. In other words, the trivial protocol where Alice sends Bob her input X (n bits), and Bob responds whether $X = Y$ (1 bit), is optimal.

As in many other areas of theoretical computer science, there is much to be gained from randomization. For example, in practice, the Equality function does not require linear communication as Alice and Bob can just hash their inputs and compare the hash keys. The shorter protocol may return a false positive, but it is correct with high probability, and reduces the communication complexity from $n+1$ to $O(\log n)$.

More generally, a randomized protocol is a protocol that tosses coins (i.e. accesses random bits), and produces the correct answer with high probability. The *distributional setting*, where there is a prior probability distribution μ on the inputs and the players need to output the correct answer with high probability with respect to μ is closely related to the randomized setting, as will be seen below. In the randomized setting there are two possible types of random coins. *Public coins* are generated at random and are accessible to both Alice and Bob at no communication cost. *Private coins* are coins generated privately by Alice and Bob, and are only accessible by the player who generated them. If Alice wants to share her coins with Bob, she needs to use the communication channel. In the context of communication complexity the pubic-coin model is clearly more powerful than the private coin one. Fortunately, the gap between the two is not very large [78], and can be mostly ignored. For convenience reasons, we will focus on the public-coin model.

The definition of a randomized public-coin communication protocol π_R is identical to Definition 2.1, except a public random string R is chosen at the beginning of the execution of the randomized π_R, and all functions at the nodes of π_R may depend on R in addition to the respective input X or Y. We still require the answer f_ℓ to be unequivocally determined by the leaf ℓ alone. The communication cost $|\pi_R|$ of π_R is still its worst-case communication cost (for historic reasons; an average-case notion would also have been meaningful to discuss here).

The randomized communication complexity of F with error $\varepsilon > 0$ is given by

$$R_\varepsilon(F) := \min_{\pi_R : \forall X,Y \, \Pr_R[\pi_R(X,Y)=F(X,Y)] \geq 1-\varepsilon} |\pi_R|. \tag{2.4}$$

For a distribution μ on $\mathcal{X} \times \mathcal{Y}$ the *distributional* communication complexity $D_{\mu,\varepsilon}(F)$ is defined as the cost of the best protocol that achieves expected error ε with respect to μ. Note that in this case fixing public randomness R to a uniformly random value does not change (on average) the expected success probability of π_R with respect to μ. Therefore, without loss of generality, we may require π to be deterministic:

$$D_{\mu,\varepsilon}(F) := \min_{\pi : \mu\{X,Y : \ \pi(X,Y)=F(X,Y)\} \geq 1-\varepsilon} |\pi|. \tag{2.5}$$

It is easy to see that for all μ, $D_{\mu,\varepsilon}(F) \leq R_\varepsilon(F)$. By an elegant minimax argument [106], a partial converse is also true: for each F and ε, there is a distribution against which the distributional communication complexity is as high as the randomized:

$$R_\varepsilon(F) = \max_\mu D_{\mu,\varepsilon}(F). \tag{2.6}$$

For this reason, we will be able to discuss distributional and randomized communication complexity interchangeably.

How can one prove lower bounds for the randomized setting? This setting is much less restrictive than the deterministic one, making lower bounds more challenging. Given a function F, one can guess the hard distribution μ, and then try to lower bound the distributional communication complexity $D_{\mu,\varepsilon}(F)$ — that is, show that there is no low-communication protocol π that computes F with error $\leq \varepsilon$ with respect to μ. Such a protocol π of cost $k = |\pi|$ still induces a partition $\{S_\ell\}_{\ell\in\mathcal{L}}$ of the inputs according to the leaf they reach, with $L \leq 2^k$ and each S_ℓ a combinatorial rectangle. However, it is no longer the case that when we consider the corresponding submatrix M_ℓ of M_F it must be monochromatic — the output of π is allowed to be wrong on a fraction of S_ℓ, and thus for some inputs the output of π on S_ℓ may disagree with the value of F. Still, it should be true that for *most* leaves the value of F on S_ℓ is strongly biased one way or the other, since the contribution of S_ℓ to the error is

$$e(S_\ell) = \min\left(\mu(S_\ell \cap F^{-1}(0)), \mu(S_\ell \cap F^{-1}(1))\right). \tag{2.7}$$

In particular, a fruitful lower bound strategy is to show that all "large" rectangles with respect to μ have $e(S_\ell)/\mu(S_\ell) \gg \varepsilon$, and thus there must be many smaller rectangles — giving a lower bound on $L \leq 2^{|\pi|}$. One simple instantiation of this strategy is the *discrepancy* bound: for a distribution μ, the discrepancy $Disc_\mu(F)$ of F with respect to μ is the maximum over all combinatorial rectangles R of

$$Disc_\mu(R,F) := |\mu(F^{-1}(0)\cap R) - \mu(F^{-1}(1)\cap R)|.$$

In other words, if F has low discrepancy with respect to μ, then only very small rectangles (as measured by μ) can be unbalanced. With some calculations, it can be shown that for all $\varepsilon > 0$ (see [68] and references therein),

$$D_{\mu,\frac{1}{2}-\varepsilon}(F) \geq \log_2(2\varepsilon/Disc_\mu(F)). \tag{2.8}$$

Note that (2.8) not only says that if the discrepancy is low then the communication complexity is high, but also that it remains high even if we are only trying to gain a tiny advantage over random guessing in computing F! An example of a natural function to which the discrepancy method can be applied is the n-bit Inner Product function $IP_n(X,Y) = \langle X,Y\rangle \mod 2$. This simple discrepancy method can be generalized to a richer family of *corruption bounds* that can be viewed as combinatorial generalizations of the discrepancy bound. More on this method can be found in the survey [70].

One of the early successes of applying combinatorial methods in communication complexity was the proof that the *randomized* communication complexity of the set disjointness problem (2.2) is linear, $R_{1/4}(Disj_n) = \Theta(n)$. The first proof of this fact was given in the 1980s [60], and a much simpler proof was discovered soon after [88]. The proofs exhibit a specific distribution μ of inputs on which the distributional communication complexity $D_{\mu,1/4}(Disj_n)$ is $\Omega(n)$. Note that the uniform distribution would not be a great fit, since uniformly drawn sets are non-disjoint with a very high probability. It turns out that the following family of distributions μ is hard: select each coordinate pair (X_i, Y_i) i.i.d. from a distribution on $\{(0,0),(0,1),(1,0)\}$ (e.g. uniformly). This generates a distribution on pairs of disjoint sets. Now, with probability $1/2$ choose a uniformly random coordinate $i \in_U [n]$ and set $(X_i, Y_i) = (1,1)$ (and with probability $1/2$ do nothing). Thus, under μ, X and Y are disjoint with probability $1/2$.

Treating communication complexity as a generalization of one-way communication and applying information-theoretic machinery to it is a very natural approach (perhaps the most natural, given the success of information theory in communication theory). Interestingly, however, this is not how the field has evolved. For example, a 2009 survey [70] was able to present the vast majority of communication complexity results up until then without dealing with information theory at all. It is hard to speculate why this might have been the case. One possible explanation is that the mathematical machinery needed to tackle the (much more complicated) interactive case from the information-theoretic angle wasn't available until the 1990s; another possible explanation is that linear algebra, linear programming duality, and combinatorics (the main tools in communication complexity lower bounds) are traditionally more central to theoretical computer science research and education than information theory.

A substantial amount of literature exists on communication complexity within the information theory community. See for example [81,82] and references therein. The flavor of the results is usually different from the ones discussed above. In particular, there is much more focus on bounded-round communication, and significantly less focus on techniques for obtaining specific lower bounds on the communication complexity of specific functions such as the disjointness function. The most relevant work to our current discussion is a more recent line of work by Ishwar and Ma, which studied interactive amortized communication and obtained characterizations closely related to the ones discussed below [73,74], building on earlier works of Wyner and Ziv [104] from the 1970s.

Within the theoretical computer science literature, in the context of communication complexity[18], information theoretic tools were explicitly introduced in [31] in the early 2000s for the simultaneous message model (i.e. 2 non-interactive rounds of communication). Building on this work, [8] developed tools for applying information theoretic reasoning to fully interactive communication, in particular giving an alternative (arguably, more intuitive) proof for the $\Omega(n)$ lower bound on the communication complexity of $Disj_n$. The motivating questions for [31], as well as for subsequent works developing information complexity, were the *direct sum* [39] and *direct product* questions for (randomized) communication complexity.

[18] As with many other concepts within theoretical computer science, it was introduced earlier in a more applied context, namely quantifying information-theoretic privacy of communication protocols [7, 90]. The two lines of work only converged later, after information complexity was developed in the context of direct sum in communication complexity.

The direct sum problem. In general, a direct sum theorem quantifies the cost of solving a problem F^n consisting of n sub-problems in terms of n and the cost of each sub-problem F. The value of such results to lower bounds is clear: a direct sum theorem, together with a lower bound on the (easier-to-reason-about) sub-problem, yields a lower bound on the composite problem (a process also known as hardness amplification). For example, the Karchmer-Wigderson program for boolean formulae lower bounds can be completed via a (currently open) direct sum result for a certain communication model [62].

The direct sum property, while useful, is often untrue — sometimes in unexpected or profound ways. Consider the example of matrix-vector multiplication over $\mathbb{F}_2$. The matrix $A \in \mathbb{F}_2^{n\times n}$ is chosen at random and fixed. The input is $x \in \mathbb{F}_2^n$, and the n-bit output is Ax. The computational model is boolean circuits (as in P/poly discussed earlier). A simple counting argument shows that with high probability, for a randomly chosen A, computing Ax requires a circuit of size $\tilde{\Omega}(n^2)$[19]. On the other hand, computing $Ax_1, \ldots, Ax_n$ for n vectors in parallel amounts to multiplying A by an $n \times n$ matrix. This can be done in time (and also circuit size) $n^\omega = O(n^{2.38}) \ll n \times n^2$, showing a violation of direct sum for this model. We will return to the direct sum problem for randomized communication complexity in the next section.

Direct product results further sharpen direct sum theorems by showing a "threshold phenomenon", where solving F^n with insufficient resources is shown to be impossible to achieve except with an exponentially small success probability. Classic results in complexity theory, such as Raz's Parallel Repetition Theorem [86] can be viewed as a direct product result. Direct product theorems are also important in the context of cryptography: by repeating a challenge n times, one hopes to boost the security of a system exponentially.

In the next section, we will formally introduce information complexity. We will first look at it as a generalization of Shannon's entropy to interactive tasks. We will then discuss its connections to the direct sum and product questions for randomized communication complexity, as well as other connections.

3. Information complexity

Interactive information complexity. In this section we will work towards developing information complexity as the analogue of Shannon's entropy for interactive computation. It will sometimes be convenient to work with general *interactive two-party tasks* rather than just functions. A task $T(X, Y)$ is any action on inputs (X, Y) that can be performed by a protocol. $T(X, Y)$ can be thought of as a set of distributions of outputs that are acceptable given an input (X, Y). Thus "computing $F(X, Y)$ correctly with probability $1 - \varepsilon$" is an example of a task, but there are examples of tasks that do not involve function or relation computation, for example "Alice and Bob need to sample strings A and B, respectively, distributed according to $(A, B) \sim \mu_{(X,Y)}$". For the purposes of the discussion, it suffices to think about T as the task of computing a function with some success probability. The communication complexity of a task T is then defined analogously to the communication complexity of functions. It is

[19] Here $\tilde{\Omega}(\cdot)$ hides factors polynomial in $\log n$.

the least amount of communication needed to successfully perform the task $T(X,Y)$ by a communication protocol $\pi(X,Y)$.

The *information complexity* of a task T is defined as the least amount of information Alice and Bob need to exchange (i.e. reveal to each other) about their inputs to successfully perform T. This amount is expressed using mutual information (specifically, conditional mutual information (1.12)). We start by defining the *information cost* of a protocol π. Given a prior distribution μ on inputs (X,Y) the information cost is

$$\mathsf{IC}(\pi,\mu) := I(Y;\Pi|X) + I(X;\Pi|Y), \tag{3.1}$$

where Π is the random variable representing a realization of the protocol's transcript, including the *public* randomness it uses[20]. In other words, (3.1) represents the sum of the amount of information Alice learns about Y by participating in the protocol and the amount of information Bob learns about X by participating. Note that the prior distribution μ may drastically affect $\mathsf{IC}(\pi,\mu)$. For example, if μ is a singleton distribution supported on one input (x_0,y_0), then $\mathsf{IC}(\pi,\mu) = 0$ for all π, since X and Y are already known to Bob and Alice respectively under the prior distribution μ. Definition (3.1), which will be justified shortly, generalizes Shannon's entropy in the non-interactive regime. Indeed, in the transmission case, Bob has no input, thus $X \sim \mu$, $Y = \perp$, and Π allows Bob to reconstruct X, thus $\mathsf{IC}(\pi,\mu) = I(X;\Pi) = H(X) - H(X|\Pi) = H(X) - 0 = H(X)$.

The *information complexity* of a task T can now be defined similarly to communication complexity in (2.5):

$$\mathsf{IC}(T,\mu) := \inf_{\pi \text{ successfully performs } T} \mathsf{IC}(\pi,\mu). \tag{3.2}$$

One notable distinction between (2.5) and (3.2) is that the latter takes an infimum instead of a minimum. This is because while the number of communication protocols of a given communication cost is finite, this is not true about information cost. One can have a sequence $\pi_1, \pi_2, \ldots$ of protocols of ever-increasing communication cost, but whose information complexity $\mathsf{IC}(\pi_n,\mu)$ converges to $\mathsf{IC}(T,\mu)$ in the limit. Moreover, as we will discuss later, this phenomenon is already observed in very simple tasks T, such as computing the conjunction of two bits.

Our discussion of information complexity will be focused on the slightly simpler to reason about *distributional* setting, where inputs are distributed according to some prior μ. In (3.2), if T is the task of computing a function F with error ε w.r.t. μ, the distribution μ is used twice: first in the definition of "success", and then in measuring the amount of information learned. It turns out that it is possible to define worst-case information complexity [13] as the information complexity with respect to the worst-possible prior distribution in the spirit of the minimax relationship (2.6). In particular, the direct sum property of information complexity which we will discuss below holds for prior-free information complexity as well.

[20] The protocol is also allowed to use private randomness, known to only one of the two parties, that is not automatically included in the transcript. Unlike the context of communication complexity, in information complexity private randomness is more useful than public randomness [27].

3.1. Direct sum for information and amortized communication

Information complexity as defined here has been extensively studied (see e.g. survey [102]). In particular, it is surprisingly simple to show that information complexity is additive for tasks over independent pairs of inputs. Let T_1 and T_2 be two tasks over pairs of inputs (X_1, Y_1), (X_2, Y_2), and let μ_1, μ_2 be distributions on pairs (X_1, Y_1) and (X_2, Y_2), respectively. Denote by $T_1 \otimes T_2$ the task composed of successfully performing both T_1 and T_2 on the respective inputs (X_1, Y_1) and (X_2, Y_2). Then information complexity is additive over these two tasks:

Theorem 3.1. $\mathsf{IC}(T_1 \otimes T_2, \mu_1 \times \mu_2) = \mathsf{IC}(T_1, \mu_1) + \mathsf{IC}(T_2, \mu_2)$.

Proof. (Sketch; a complete proof of a slightly more general statement can be found in [13]). The "easy" direction of this theorem is the '$\leq$' direction. Take two protocols π_1 and π_2 that perform T_1 and T_2 respectively, and consider the concatenation $\pi = (\pi_1, \pi_2)$ (which clearly performs $T_1 \otimes T_2$). Consider what Alice learns from an execution of π with prior $\mu_1 \times \mu_2$. A straightforward calculation using, for example, repeated application of the chain rule (1.13) yields

$$I(Y_1Y_2; \Pi_1\Pi_2 | X_1X_2) = I(Y_1; \Pi_1 | X_1) + I(Y_2; \Pi_2 | X_2).$$

And similar statement is true about what Bob learns as well. Therefore $\mathsf{IC}(\pi, \mu_1 \times \mu_2) = \mathsf{IC}(\pi_1, \mu_1) + \mathsf{IC}(\pi_2, \mu_2)$. By passing to the limit as $\mathsf{IC}(\pi_1, \mu_1) \to \mathsf{IC}(T_1, \mu_1)$ and $\mathsf{IC}(\pi_2, \mu_2) \to \mathsf{IC}(T_2, \mu_2)$ we obtain the '$\leq$' direction.

The '$\geq$' direction is more interesting, even if the proof is not much more complicated. In this direction we are given a protocol π for solving $T_1 \otimes T_2$ with information cost $I = \mathsf{IC}(\pi, \mu_1 \times \mu_2)$, and we need to construct out of it two protocols for T_1 and T_2 of information costs I_1 and I_2 that add up to $I_1 + I_2 \leq I$. We describe the protocol $\pi_1(X_1, Y_1)$ below:

$\boldsymbol{\pi_1(X_1, Y_1)}$:

- Bob samples a pair $(X_2, Y_2) \sim \mu_2$, and sends X_2 to Alice;
- Alice and Bob execute $\pi((X_1, X_2), (Y_1, Y_2))$, and output the portion relevant to T_1 in the performance of $T_1 \otimes T_2$.

It is not hard to see that the tuple (X_1, Y_1, X_2, Y_2) is distributed according to $\mu_1 \times \mu_2$, and hence by the assumption on π, π_1 successfully performs T_1. Note that there is a slight asymmetry in π_1: X_2 is known to both Alice and Bob while Y_2 is only known to Bob. For the purpose of correctness, the protocol would have worked the same if Bob also sent Y_2 to Alice, but it is not hard to give an example where the information cost of π_1 in that case is too high. The information cost of π is thus given by the sum of what Bob learns about X_1 from π_1 and what Alice learns about Y_1 (note that (X_2, Y_2) are not part of the input):

$$I_1 = I(X_1; \Pi | X_2Y_1Y_2) + I(Y_1; \Pi | X_1X_2).$$

The protocol $\pi_2(X_2, Y_2)$ is defined similarly to π_1 in a skew symmetric way:

$\boldsymbol{\pi_2(X_2, Y_2)}$:

- Alice samples a pair $(X_1, Y_1) \sim \mu_1$, and sends Y_1 to Bob;
- Alice and Bob execute $\pi((X_1, X_2), (Y_1, Y_2))$, and output the portion relevant to T_2 in the performance of $T_1 \otimes T_2$.

We get that π_2 again successfully performs T_2, and its information cost is:

$$I_2 = I(X_2; \Pi | Y_1 Y_2) + I(Y_2; \Pi | X_1 X_2 Y_1).$$

Putting I_1 and I_2 together using the Chain Rule (1.13) we get:

$$\begin{aligned} I_1 + I_2 = I(X_1; \Pi | X_2 Y_1 Y_2) + I(Y_1; \Pi | X_1 X_2) + I(X_2; \Pi | Y_1 Y_2) + I(Y_2; \Pi | X_1 X_2 Y_1) = \\ I(X_2; \Pi | Y_1 Y_2) + I(X_1; \Pi | X_2 Y_1 Y_2) + I(Y_1; \Pi | X_1 X_2) + I(Y_2; \Pi | X_1 X_2 Y_1) = \\ I(X_1 X_2; \Pi | Y_1 Y_2) + I(Y_1 Y_2; \Pi | X_1 X_2) = I. \end{aligned}$$

Once again, passing to the limit, gives us the '$\geq$' direction, and completes the proof. ■

If we denote an n-time repetition of a task T by $T^{\otimes n}$, then repeatedly applying Theorem 3.1 yields

$$\mathsf{IC}(T^{\otimes n}, \mu^n) = n \cdot \mathsf{IC}(T, \mu). \tag{3.3}$$

Thus information complexity is additive and has the *direct sum property*: the cost of n copies of T scales as n times the cost of one copy. This fact can be viewed as an extension of the property $H(X^n) = n \cdot H(X)$ to interactive problems, but what does it teach us about communication complexity?

Information equals to amortized communication. Let us return to the communication complexity setting, fixing T to be the task of computing a function $F(X, Y)$ with some error at most $\varepsilon > 0$ over a distribution μ (the case $\varepsilon = 0$ seems to be different from $\varepsilon > 0$). We will denote by F_ε^n the task of computing n copies of F on independent inputs distributed according to μ^n, with error at most ε *on each copy* (note that computing F correctly with error at most ε on all copies simultaneously is a harder task).

It is an easy observation that the information cost of a protocol π is always bounded by its length $|\pi|$, and therefore information complexity is always bounded by communication complexity. Therefore, by (3.3),

$$\frac{1}{n} \cdot D_{\mu^n}(F_\varepsilon^n) \geq \frac{1}{n} \cdot \mathsf{IC}(F_\varepsilon^n, \mu^n) = \mathsf{IC}(F_\varepsilon, \mu). \tag{3.4}$$

It turns out that the converse is also true in the limit, as $n \longrightarrow \infty$ **[22]**[21]:

$$\lim_{n \to \infty} \frac{1}{n} \cdot D_{\mu^n}(F_\varepsilon^n) = \mathsf{IC}(F_\varepsilon, \mu). \tag{3.5}$$

[21] More precisely, the converse adds error that vanishes exponentially in n (and thus goes to 0 as $n \to \infty$). Such a statement would be false with no errors allowed **[77, 80]**. Therefore, (3.5) only holds when $\mathsf{IC}(F_\varepsilon, \mu)$ is continuous in ε as we approach from ε^+. In particular, this means that in many applications we need $\varepsilon > 0$ for it to hold, as there is often a discontinuity at $\varepsilon = 0$.

Equation (3.5) can be viewed as the interactive version of the Source Coding Theorem (1.6). In particular, it gives an operational characterization of information complexity exclusively in terms of communication complexity. The link given by (3.5) has been further refined in [100], establishing the second-order term in the equation.

3.2. Direct sum and direct product for communication

Direct sum and interactive compression. Recall that the direct sum property asserts that solving n copies of a problem requires n times the resources it takes to solve one copy. It is one of the most generic tools one can deploy (or hope to deploy) in the quest for unconditional lower bounds.

Theorem 3.1 implies that the direct sum property holds exactly for information complexity. In addition, (3.5) immediately gives us a handle on the direct sum question for *communication complexity*.

The *direct sum* question for communication complexity asks whether

$$D_{\mu^n}(F_\varepsilon^n) = \Omega(n \cdot D_\mu(F_\varepsilon))? \tag{3.6}$$

By (3.5), the question (3.6) is equivalent to

$$\mathsf{IC}(F_\varepsilon, \mu) = \Omega(D_\mu(F_\varepsilon))? \tag{3.7}$$

Or, switching directions,

$$D_\mu(F_\varepsilon) = O(\mathsf{IC}(F_\varepsilon, \mu))? \tag{3.8}$$

Note that the equivalence works on a per-problem basis, so whenever (3.8) holds for a given problem, direct sum for communication complexity holds for that problem. On the other hand, to show that direct sum for communication complexity fails in general, it suffices to give one example of a function where $D_\mu(F_\varepsilon) = \omega(\mathsf{IC}(F_\varepsilon, \mu))$.

One natural way to interpret (3.8) is through the lens of interactive compression — an interactive analogue of Huffman coding (1.9), where it does hold that $H(X) > C(X) - 1$. Huffman (one way) coding shows how to encode a low-entropy "uninformative" signal into a short one. Its interactive version seeks to simulate a low information cost "uninformative" protocol π with a low communication protocol π'.

It turns out that such a compression scheme is impossible, disproving the direct sum conjecture through the information complexity route. In a series of breakthrough works, Ganor, Kol, and Raz [43–45] give an example of a function whose information complexity is exponentially smaller than its communication complexity. That is, in [44] — building on earlier works by the same authors — they present an F such that

$$D_\mu(F_\varepsilon) = 2^{\Omega(\mathsf{IC}(F_\varepsilon, \mu))} \gg \mathsf{IC}(F_\varepsilon, \mu). \tag{3.9}$$

In fact, the exponential gap is the largest possible, as it can be shown [13] for all F,

$$D_\mu(F_\varepsilon) = 2^{O(\mathsf{IC}(F_\varepsilon, \mu))}. \tag{3.10}$$

To prove the strongest possible direct sum theorem (3.6) we would have needed π' to be compressed all the way down to $O(I)$ bits of communication (the strongest possible interactive compression result). Even though such a compression is impossible, weaker interactive compression results lead to weaker (but still non-trivial) direct sum theorems. At present, the two strongest compression results, which partially resolve Problem 3.2, compress π to $\tilde{O}(\sqrt{C \cdot I})$ communication[22] [9] and $2^{O(I)}$ communication (3.10), respectively. Note that these results are incomparable since $C > I$ can be much (e.g. double-exponentially) larger than I.

These result lead to direct sum theorems for randomized communication complexity. As the compression introduces an additional small amount of error, the first result implies for any constant $\rho > 0$:

$$D_{\mu^n}(F_\varepsilon^n) = \tilde{\Omega}(\sqrt{n} \cdot D_\mu(F_{\varepsilon+\rho})), \tag{3.11}$$

and the second one implies

$$D_{\mu^n}(F_\varepsilon^n) = \Omega(n \cdot \log(D_\mu(F_{\varepsilon+\rho}))). \tag{3.12}$$

In summary, we know that perfect compression a la Huffman is impossible in the two-party interactive setting. Mapping out the exact limits of interactive compression remains open:

Problem 3.2. *(Interactive compression problem). Given a protocol π whose communication cost is C and whose information cost is I, what is the smallest amount of communication needed to (approximately) simulate π?*

As noted above, we know that whenever $I \ll C$, the protocol can be compressed to $o(C)$ bits of communication. At the same time, it is unknown, for example, whether compression to $I^{O(1)} \cdot (\log C)^{O(1)}$ or even to $I^{O(1)} \cdot C^{o(1)}$ is possible. A candidate problem for such a lower bound on compression is presented in [14].

Direct product for communication complexity. Next, we turn our attention to the more difficult *direct product* problem for communication complexity. The direct sum question talks about the amount of resources needed to achieve a certain probability of success on n copies of F. What if that amount of resources is not provided? For example, (3.4) implies that unless $n \cdot \mathsf{IC}(F_\varepsilon, \mu)$ bits of communication are allowed in the computation of F_ε^n, the computation of *some* copy of F will have $< 1 - \varepsilon$ success probability. What does it tell us about the success probability of *all* copies simultaneously? It only tells us that the probability of the protocol succeeding on all copies simultaneously is bounded by $1 - \varepsilon$. This is a very weak bound, since solving the n copies independently leads to a success probability of $(1 - \varepsilon)^n$, which is exponentially small for a constant ε. How can this gap be reconciled? In particular, can one show that Alice and Bob cannot "pool" the errors from all n copies onto the same instances, thus keeping the success probability for each coordinate, as well as the global success probability, close to $1 - \varepsilon$? The direct product problem addresses precisely this question. Let us

[22] Here, the $\tilde{O}(\cdot)$ notation hides poly-logarithmic factors.

denote by $\mathsf{suc}(F,\mu,C)$ the highest success probability (w.r.t. μ) in computing F that can be attained using communication $\leq C$. Thus $\mathsf{suc}(F,\mu,C) \geq 1-\varepsilon$ is equivalent to $D_\mu(F_\varepsilon) \leq C$. Somewhat informally phrased, the direct product question asks whether

$$\mathsf{suc}(F^n,\mu^n,o(n\cdot C)) < \mathsf{suc}(F,\mu,C)^{\Omega(n)}? \tag{3.13}$$

The examples showing that (3.6) fails also show that direct product (3.13) for communication is false. The direct sum discussion already suggests that for $\mathsf{suc}(F,\mu,C) = 1-\varepsilon$, the best scaling of the amount of communication one can hope for is $n \cdot I$, where $I = \mathsf{IC}(F_\varepsilon,\mu)$. This is because, as $n \longrightarrow \infty$, the per-copy communication cost of computing F with error ε scales as $n \cdot I$.

Thus, the "right" question is whether the direct product property holds when communication scales as the *information complexity* of the problem. If we denote by $\mathsf{suc}^{\mathsf{i}}(F,\mu,I) \geq \mathsf{suc}(F,\mu,I)$ the best success probability one can attain solving F while incurring an *information cost* of at most I, the direct product question for information asks whether

$$\mathsf{suc}(F^n,\mu^n,o(n\cdot I)) < \mathsf{suc}^{\mathsf{i}}(F,\mu,I)^{\Omega(n)}? \tag{3.14}$$

Note that the success probability on the left-hand-side is still with respect to communication. A statement such as this with respect to information cost is bound to be false: Information cost being an average-case quantity, one can attain an information-cost I_n protocol by doing nothing with probability $1-\delta$, and incurring an information cost of $I_n/\delta \gg n \cdot I$ with probability δ that can be taken only *polynomially* (and not exponentially) small.

This latter version of the direct product theorem was shown to be true up to polylogarithmic factors for boolean functions in [24,25]. To simplify parameters, suppose $\mathsf{suc}^{\mathsf{i}}(F,\mu,I) < 2/3$. Then there are constants c_1, c_2 such that

$$\text{if } T\log T < c_1 n \cdot I, \text{ then } \mathsf{suc}(F^n,\mu^n,T) < 2^{-c_2 n}. \tag{3.15}$$

The proof of (3.15) is quite involved and combines ideas from the proof of direct sum theorems and of parallel repetition theorems. The main idea is that an event that happens with probability $> 2^{-c_2 n}$ (namely, the event of succeeding on all coordinates) "confers" at most $\sim c_2$ bits of information onto each coordinate. If c_2 is a small constant, then this extra information is very small and can be ignored. The actual proof involves developing the right information-theoretic language to make this simple-sounding ideas rigorous.

We next turn our attention to an early application of information complexity: exact bounds on communication complexity. We briefly discuss additional applications in Section 3.4.

3.3. Exact communication complexity of set disjointness

One of the great successes of information theory as it applies to (classical, one-way) communication problems is its ability to give precise answers to fairly complicated asymptotic communication problems — ones involving complicated dependencies between terminals or complicated channels. Using combinatorial techniques (in most cases) such precision is inaccessible in the two-party setting, since the techniques often lose constant factors by design. In contrast, information complexity extends the precision benefits of one-way information theory to the interactive setting.

We give one specific example of an exact communication complexity bound. Recall that the disjointness problem $Disj_n(X,Y)$ takes two n-bit vectors X,Y and checks whether there is a location with $X_i = Y_i = 1$. Thus $Disj_n$ is just a disjunction of n independent copies of the two bit $AND(X_i,Y_i)$ function. Using techniques similar to the proof of Theorem 3.1, one can show that the communication complexity of disjointness is tightly linked with the information complexity of AND. Note that disjointness becomes trivial if many coordinates (X_j,Y_j) of the input are $(1,1)$. However, any distribution of inputs where $\mu((X_j,Y_j) = (1,1)) \sim 1/n \to 0$ will not be trivial. More formally, denote by 0^+ a function $f(n)$ of n such that $f(n) = o(1)$ and $f(n) \gg 2^{-O(n)}$. For example, one can take $f(n) = 1/n$. Denote by AND_0 the task of computing AND correctly on all four possible inputs. Then with some work one shows [18] that

$$R_{0^+}(Disj_n) = \left(\inf_{\mu:\mu(1,1)=0} \mathsf{IC}(AND_0,\mu)\right) \cdot n \pm o(n). \tag{3.16}$$

Thus, understanding the precise asymptotics of the communication complexity of $Disj_n$ boils down to understanding the (0-error) information complexity of the two-bit AND function[23].

The information-theoretically optimal protocol for the two-bit AND function (and for any other function) depends on the prior distribution of the inputs. The protocol attaining the optimal information complexity for the two-bit AND function for symmetric prior distributions (where $\mu(0,1) = \mu(1,0)$) is given in Figure 1[24]

Observe that the "protocol" in Figure 1 is not an actual communication protocol: it involves a continuous-time clock, and not a finite sequence of discrete messages. The protocol can be approximated by a discrete protocol by sampling N^A and N^B from the discrete set $\{0, \frac{1}{r}, \frac{2}{r}, \ldots, \frac{r-1}{r}\}$ instead of $[0,1)$, and then having r iterations of the clock going over multiples of $\frac{1}{r}$.

Interestingly, even in the case of such a simple function as two-bit AND, the information complexity is not attained by any particular protocol, but rather by an infinite family of communication protocols! Moreover, if we denote by $IC_r(AND_0)$ the information complexity of AND_0 where the infimum in (3.2) is only taken over protocols of length r, then it turns out that $IC_r(AND_0) = IC(AND_0) + \Theta(1/r^2)$, implying that an asymptotically optimal protocol is only achieved with a super-constant number of rounds [18]. We do not yet know how general this $1/r^2$ gap phenomenon is, and which communication tasks admit a minimum in (3.2).

[23] Note that even when $\mu(1,1) = 0$ and thus $AND(X,Y) = 0$ on $\mathrm{supp}(\mu)$, the task AND_0 requires the protocol to *always* be correct – even on the $(1,1)$ input. Otherwise, $\mathsf{IC}(AND_0,\mu)$ would trivially be 0.

[24] The protocol for general μ is an extension of the protocol in Figure 1, and can be found in [18]

Alice receives $x \in \{0,1\}$; Bob receives $y \in \{0,1\}$.
Goal: to compute $AND(x,y) = x \wedge y$ with no error.

(1) *If $x = 0$ then Alice samples $N^A \in_R [0,1)$ uniformly at random.*
If $x = 1$ then Alice sets $N^A = 1$.

(2) *If $y = 0$ then Bob samples $N^B \in_R [0,1)$ uniformly at random.*
If $y = 1$ then Bob sets $N^B = 1$.

(3) *Alice and Bob monitor the clock C, which starts at value* 0.

(4) *The clock continuously increases to* 1. *If* $\min(N^A, N^B) < 1$, *when the clock reaches* $\min(N^A, N^B)$ *the corresponding player sends* 0 *to the other player, the protocol ends, the players output 0. If* $\min(N^A, N^B) = 1$, *once the clock reaches* 1, *Alice sends* 1 *to Bob, the protocol ends, and the players output* 1.

Figure 1
The information-theoretically optimal protocol for $AND(x,y)$ under prior distribution μ with $\mu(0,1) = \mu(1,0)$

By calculating the information cost of the optimal protocol for AND, and maximizing it over all possible distributions μ with $\mu(1,1) = 0$, we obtain from (3.16):

$$R_{0^+}(Disj_n) = C_{DISJ} \cdot n \pm o(n), \quad \text{where} \;\; C_{DISJ} \approx 0.4827. \tag{3.17}$$

Small set Disjointness. An interesting special case of the set-disjointness problem, is the *small set disjointness* case. In this setting, only at most $k \ll n$ of the X_i's are 1 and at most k of the Y_i's are 1. In other words, Alice and Bob each have a set of k elements over a universe of $n \gg k$ elements, and they wish to determine whether they have an element in common. Denote this problem by $Disj_{n,k}$.

The naïve upper bound in this case is $O(k \log n)$, since it takes $O(\log n)$ bits to transmit a single element from the set $\{1,\dots,n\}$[25]. Somewhat surprisingly, Håstad and Wigderson **[52]** showed that small set disjointness can be solved using communication linear in k:

$$R_{0^+}(Disj_{n,k}) = O(k). \tag{3.18}$$

Note that the $\Omega(n)$ lower bound for $Disj_n$ immediately translates into an $\Omega(k)$ lower bound for $Disj_{n,k}$, leading to

$$R_{0^+}(Disj_{n,k}) = \Theta(k). \tag{3.19}$$

[25] The precise bound is $O(k\log(n/k))$, but this becomes $O(k \log n)$ whenever $n > k^{1+c}$

It turns out that the precise bound follows from the optimality of the protocol in Figure 1 almost immediately. The relevant distribution for the single AND instance is one where the probability of $X = 1$ is $\frac{k}{n}$. Calculating the information cost of the protocol with prior $\mu(1,0) = \mu(0,1) = \frac{k}{n}$, $\mu(0,0) = 1 - \frac{2k}{n}$ yields [18]:

$$R_{0^+}(Disj_{n,k}) = \frac{2}{\ln 2} \cdot k \pm o(k). \tag{3.20}$$

3.4. Some other connections

Let us briefly mention some recent connections between information complexity and other sub-areas of theoretical computer science.

Streaming: do we need numbers to approximately count? Beyond answering questions such as the direct sum for randomized communication complexity, the main advantage of information complexity is that it allows us to phrase intuitive statements about computation and communication in a rigorous way. We will illustrate it with a sketch of a recent result about the streaming complexity of approximate majority [19].

In the *streaming* setup, inputs $X_1, \ldots, X_n$ arrive one-by-one, and the state of the computation is updated based on the input and the previous state. Thus, the computation can be represented as the following diagram:

$$M_0 \xrightarrow{X_1} M_1(M_0, X_1) \xrightarrow{X_2} M_2(M_1, X_2) \xrightarrow{X_3} \ldots \xrightarrow{X_n} M_n(M_{n-1}, X_n)$$

The answer is then computed from the final state M_n. Typically we are interested in either the average memory used by the algorithm $\bar{m} = \frac{1}{n}\sum_i |M_i|$, or the maximum amount of memory $m_{max} := \max |M_i|$[26].

Consider the following problem: *Given n i.i.d. coin tosses of $X_i \sim B_{1/2}$, compute* $\mathbf{MAJ}(X_1, \ldots, X_n)$ *while allowing a 1% error probability*[27].

The simplest possible algorithm would just count the bits: set $M_0 = 0$ and

$$M_i(M_{i-1}, X_i) := M_{i-1} + X_i,$$

so that $M_n = \sum X_i$, from which one can compute $\mathbf{MAJ}(X_1, \ldots, X_n)$ *with no error*. This solution requires $\bar{m} \sim \log n$ memory. It is not hard to show that producing an exact count requires this much memory. What about approximate counting? Can we avoid storing numbers if we only wish to count the numbers approximately? It turns out that the answer is 'no': indeed $\bar{m} = \Omega(\log n)$ is necessary.

A key step of the construction is to correctly define the information cost of this streaming setup[28]:

$$\mathsf{IC}(M) := \sum_{i=1}^{n} \sum_{j=1}^{i} I(M_i; X_j | M_{j-1}) \tag{3.21}$$

[26] Here $|M|$ is the length of M in bits

[27] That is, the algorithm needs to be correct at least 99% of the time

[28] An important benefit of an information-theoretic lower bound – as opposed to a combinatorial one – is that it can be used in the context of a direct sum theorem to lower bound the cost of doing multiple copies of a problem in parallel. Indeed, this is how it was used in [19]

Each term of the sum captures how much information the i-th message still retains about input X_j that appeared earlier. As in many other cases, information here is a lower bound on $\sum |M_i|$. It turns out that for a typical pair we must have

$$I(M_i; X_j | M_{j-1}) \gtrsim \frac{1}{i-j+1}, \tag{3.22}$$

and therefore

$$\bar{m} = \frac{1}{n} \cdot \sum_i |M_i| \geq \frac{1}{n} \cdot \mathsf{IC}(M) \gtrsim \frac{1}{n} \cdot \sum_{i=1}^{n} \sum_{j=1}^{i} \frac{1}{i-j+1} = \Theta(\log n) \tag{3.23}$$

The main inequality (3.22) is proved by rephrasing the following intuition in information-theoretic terms. If we break the stream into $k = 2^r$ blocks $B_1, \ldots, B_k$ of length n/k each, then at the end of each block B_i, the message $M_{i \cdot n/k}$ should contain at least 1 bit of information about the approximate count in the previous block. This translates into containing at least $\frac{k}{n}$ bits of information about a typical X_j in that block, leading to (3.22). This proof also gives intuition for the need to have $\Omega(\log n)$ bits of information in the streaming algorithm: ~ 1 bit of information needs to be dedicated to each of $\log n$ “scales” of the stream.

Distributed learning. All large-scale machine learning today is performed using a large number of processing cores. As a result, communication costs and delays often dominate the overall execution time. This motivates efforts to minimize communication between worker cores, and to understand the fundamental limits of communication needed to complete basic tasks — such as distributed parameter estimation [108]. Information complexity (and its ability to bring in tools from information theory, such as strong data processing inequalities) has led to tight results about problems such as distributed sparse parameter estimation [17, 46].

Parallel repetition. Parallel repetition first appeared in the context of Probabilistically Checkable Proofs (PCP) and hardness amplification. Hardness amplification is accomplished here by taking a task T (e.g. a verification procedure that allows authorized provers to pass the test, while unauthorized provers pass with probability at most $1 - \varepsilon$), and creating a task T^n by taking n independent instances of T. It has been shown [37, 54, 84, 86] that as n grows, the success probability of unauthorized provers goes to 0. Unfortunately, it does not go to 0 as $(1 - \varepsilon)^n$. Indeed, as shown by a counterexample constructed by Raz [87], the best rate one can hope for is $(1 - \varepsilon^2)^n$. The reason for this, pointed out by an earlier example by Feige and Verbitsky [41], is that the answers can be arranged to align errors together, so that when the provers fail, they fail on a lot more than εn coordinates at the same time. This is possible when answers are allowed to be correlated.

It should not be surprising that the parallel repetition problem shares some similarities with the *direct product* problem in communication complexity. In both cases, the concern is that correlations between coordinates will lead to an unexpectedly high success probability — much higher than $(1 - \varepsilon)^n$. Indeed, the proof of the direct product theorem for information complexity (3.15) can be combined with “standard” parallel repetition machinery to obtain the most general parallel-repetition theorem to-date [15].

In turn, parallel repetition has interesting connections to foams — low surface area tiling of $\mathbb{R}^n$ by $\mathbb{Z}^n$ [40], leading to new geometric constructions that implicitly use information theory [21,65].

Quantum information complexity. Information theory and its quantum extensions have been used to obtain key results in quantum communication complexity [58, 66]. The basic notions of information complexity as discussed in the previous section can be adapted to the quantum setting [98]. Unlike classical information complexity, the quantum information complexity of the two-bit AND as in (3.16) actually vanishes as the number of rounds goes to ∞. This is consistent with the fact that the quantum communication complexity of disjointness is $O(\sqrt{n}) = o(n)$ [1,28], and an earlier result by Elitzur and Vaidman on quantum bomb testing [38]. Nonetheless, it is possible to use the information complexity machinery to get a near tight bound on the information complexity of AND in terms of the number of rounds (the dependence is $\mathsf{IC}(AND_0, \mu) = \tilde{\Theta}(\frac{1}{r})$ for the best r-round protocol). This gives the tight bound of $\tilde{\Omega}(\frac{n}{r} + r)$ on the r-round quantum communication complexity of $Disj_n$ [16].

Interactive error-correcting codes. Most of the discussion so far focused on developing (two-party) information complexity as a tool for studying communication complexity and related models of computation. In other words, the motivation has been mostly complexity-theoretic.

The main aim of the original information theory project, starting from the work of Shannon in the 1940s was to further coding theory and practice. Coding theory is concerned with developing efficient codes that are robust to errors for data storage and transmission — information theory has become a tool for giving bounds (that are sometimes tight) on what codes are possibly attainable.

In the context of interactive communication one can view interactive information complexity (and even communication complexity) as one aspect of coding for interactive communication (one dealing with noiseless coding). Another important aspect of coding theory is dealing with *noisy communication*.

In the interactive setting, this gives rise to questions about interactive error-correcting codes: given a noisy channel[29], encode the entire interactive computation in a way that is robust to noise. The problem was first studied by Schulman in the 1990s [91], who showed that it is possible to protect an interactive protocol against a small amount of adversarial noise. Note that "standard" techniques of encoding each message separately cannot work here, since in such an encoding an adversary would be able to derail an entire protocol by completely replacing one of the messages.

The area has seen a resurgence of activity since the work by the author with Rao [23], which showed that it is possible to encode an interactive protocol in a way that protects it against $\frac{1}{4} - \varepsilon$ adversarial error rate. Since then there has been much activity dealing with

[29] As in the one-way communication case, the various models of noise include adversarial and various forms of random noise

making the constructions more efficient, more error resilient, and apply in a broader set of regimes. A survey on the developments in the field as of 2017 can be found in [47].

In addition to developing interactive coding schemes, some of the fundamental questions such about interactive channel capacity (as the analogue of non-interactive Shannon's channel capacity) need to be revisited in the interactive setting [20, 67].

4. Challenges and next steps

As we have seen in the last section, information complexity has been a useful tool (and the right "language") in a variety of settings involving communication. We have also briefly seen in Section 1.4 that there are several attack routes for obtaining strong (and currently apparently unreachable) separations between complexity classes using communication complexity. This raises the natural question of whether information complexity can be helpful with these communication complexity bounds.

There are several settings where information complexity (and, more broadly, information-theoretic reasoning) appears to get "stuck". Specific examples include:

- Extending tight communication lower bounds to 3 or more parties in the number-on-the-forehead model (with $(\log n)^{O(1)}$ parties this would imply difficult circuit lower bounds [10]).
- Pătraşcu's multiphase conjecture [83] — a lower bound conjecture against a specific model of computation with 3 parties. The conjecture implies strong dynamic data structures lower bounds.
- The Arthur-Merlin model in communication complexity. This is a particularly challenging model for communication complexity lower bounds. It is the communication-complexity analogue of the corresponding Arthur-Merlin AM class in computational complexity [5]. We will not define it here, only mention that it can be thought of as a communication protocol with $2 + \varepsilon$ players. "ε" here is Merlin, who can provide Alice and Bob with an untrusted hint, but then cannot participate in the protocol. There is evidence that the Arthur-Merlin communication model is resistant to information complexity techniques [49].
- Extending parallel repetition results from the setting with two provers to settings with three or more provers. While tight bounds are known in the two-prover case, there is an exponential gap between the best upper and lower bounds even in some of the simplest settings with three provers [48].

There appears to be a common theme in terms of what makes these examples difficult — namely, the existence of *secure computation* in the relevant models.

Secure computation. Throughout most of this note information complexity was presented as the interactive extension of Shannon's entropy (emphasizing connections to amortized communication cost). Historically, the fist appearance of information complexity within theoretical computer science was in the context of privacy of communication protocols [7,90][30]. The formula (3.2) for information complexity *exactly* quantifies the smallest possible information-theoretic privacy loss[31] that Alice and Bob can experience while successfully completing task T. It is important that the model here is *information-theoretic* security: it is possible to attain cryptographic security based on computational hardness assumptions [107].

In contrast with the cryptographic results, we now know that information-theoretic privacy in the honest-but-curious model is unattainable. Many of the communication complexity bounds, such as results (3.17) and (3.20) actually apply to information complexity as well, which means that for these problems there is (asymptotically) no gap between information and communication, and the shortest possible protocol is also the one that reveals the least information to the participants about the inputs. In other words, *information-theoretically secure two-party computation is impossible*.

Surprisingly, with three or more players information-theoretically secure computation becomes possible [11,32]: if Alice, Bob and Charlie have inputs X, Y, Z, respectively, and have pairwise private channels[32], then any function $F(X,Y,Z)$ can be computed in such a way that the only thing Alice learns about (Y,Z) is the value of $F(X,Y,Z)$ (and similarly for the other two players).

The result above means that while one can write the natural expression for 3-party information complexity, and even prove a direct sum result about it, the result will be vacuous: $n \times 0 = 0$, since the information complexity of any function in this model is zero.

This pattern repeats itself when one tries to prove Arthur-Merlin lower bounds using information complexity. Here, the relevant result about secure computation has to do with the channel used. Communication so far was defined over the binary channel where Alice and Bob send individual bits. A different kind of channel would take in input from both Alice and Bob, and then distribute an output to them. The simplest channel of this kind is the Shannon-Blackwell Binary Multiplying channel [93]: Alice and Bob each send a bit $a \in \{0,1\}$, $b \in \{0,1\}$, respectively into the channel, and the channel sends to both of them the value of $a \wedge b \in \{0,1\}$. Note that in this channel, if Alice sends $a = 0$ into the channel, she does not learn anything about the value of b.

It turns out that over the Binary Multiplying channel (BMC) one can implement secure *two-party* computation [64]. Once again, one can write expressions for information complexity over the BMC, and obtain direct sum results similar to Theorem 3.1 above, but the result would be vacuous of the form $0 + 0 = 0$.

Analytic techniques to bypass the secure computation barrier? It remains to be seen whether the barrier to using information-theoretic techniques (or any techniques for that matter) for the problems discussed above is merely a technical one, or is related to something deeper.

[30] And, more recently, in the context of differential privacy [76]

[31] In the "honest-but-curious" model of privacy, where participants do not actively deviate from the protocol to learn information they are not supposed to learn.

[32] that is, Alice can talk with Bob without Charlie listening

It is worth noting that in the two-party case (for both communication and parallel repetition) it is possible to rephrase most proofs in analytic terms, in terms of values of relevant semi-definite programs on the function's value matrices [37, 69, 72, 95]. In fact, in cases where both an analytic and an information-theoretic proof exists, the analytic proof often pre-dated the information-theoretic one. A notable example of a problem for which we had a number of analytic proofs [30,94] before an information-theoretic one [50] is for the Gap Hamming Distance. In other cases, such as exact communication bounds (3.17) and (3.20), information complexity appears to be the right tool.

When moving from two to three or more parties, in the analytic setup, the main object of consideration becomes tensors instead of matrices (see e.g. [26]). They are much more difficult to deal with, both because some of the nicer aspects of linear algebra are missing, and because the theory as a whole is much less developed. A promising strategy for pinning down the exact difficulty in the examples above would be to trace it to a statement about 3-dimensional tensors.

If that statement is true, the proof might be useful in communication and parallel repetition applications (as has been the case with the analytic tools in the two-party setting [37, 69, 72, 95]). Moreover, using 2-party information complexity as a guiding map, it might lead to new "information-like" definitions that don't currently exist.

If that statement is false, or turns out to be very difficult to prove even in its analytic form, then we might have discovered a mathematical obstacle to computational complexity lower bounds that would guide future lower bound efforts.

In either case, we can look forward to exciting results on the quest towards unconditional lower bounds in various computation models.

Acknowledgments

I would like to thank Ankit Garg, Sumegha Garg, and Omri Weinstein for their comments on earlier versions of this paper.

This note is based on many works with students, postdocs and colleagues. Working with them is one of the biggest privileges of my career and I am deeply grateful for everything I have learned from them and everything we have done and will continue to do in the future.

Funding

Work on this paper was supported in part by the NSF Alan T. Waterman Award, Grant No. 1933331, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry.

Work described here has been generously supported over the years by NSERC, the NSF, the Alfred P. Sloan Foundation, the University of Toronto, Princeton University, the Institute for Advanced Study, and the John Templeton Foundation, through a Turing Centenary Fellowship.

References

[1] S. Aaronson and A. Ambainis, Quantum search of spatial regions. In *44th annual ieee symposium on foundations of computer science, 2003. proceedings.*, pp. 200–209, IEEE, 2003

[2] S. Arora and B. Barak, *Computational complexity: a modern approach*. Cambridge University Press, 2009

[3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)* **45** (1998), no. 3, 501–555

[4] S. Arora and S. Safra, Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM (JACM)* **45** (1998), no. 1, 70–122

[5] L. Babai, P. Frankl, and J. Simon, Complexity classes in communication complexity theory. In *27th annual symposium on foundations of computer science (focs 1986)*, pp. 337–347, IEEE, 1986

[6] T. Baker, J. Gill, and R. Solovay, Relativizations of the p=?np question. *SIAM Journal on computing* **4** (1975), no. 4, 431–442

[7] R. Bar-Yehuda, B. Chor, E. Kushilevitz, and A. Orlitsky, Privacy, additional information and communication. *IEEE Transactions on Information Theory* **39** (1993), no. 6, 1930–1943

[8] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar, An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences* **68** (2004), no. 4, 702–732

[9] B. Barak, M. Braverman, X. Chen, and A. Rao, How to compress interactive communication. *SIAM Journal on Computing* **42** (2013), no. 3, 1327–1363

[10] R. Beigel and J. Tarui, On acc. *Computational Complexity* **4** (1994), no. 4, 350–366

[11] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson, Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the 20th annual acm symposium on theory of computing*, pp. 113–131, 1988

[12] M. Braverman, Interactive information and coding theory. In *Proceeding of the international congress of mathematicians, icm 2014*, 2014

[13] M. Braverman, Interactive information complexity. *SIAM Journal on Computing* **44** (2015), no. 6, 1698–1739

[14] M. Braverman, A. Ganor, G. Kol, and R. Raz, A candidate for a strong separation of information and communication. In *Innovations in theoretical computer science conference (itcs 2018)*, 2018

[15] M. Braverman and A. Garg, Small value parallel repetition for general games. In *Proceedings of the forty-seventh annual acm symposium on theory of computing*, pp. 335–340, 2015

[16] M. Braverman, A. Garg, Y. K. Ko, J. Mao, and D. Touchette, Near-optimal bounds on the bounded-round quantum communication complexity of disjointness. *SIAM Journal on Computing* **47** (2018), no. 6, 2277–2314

[17] M. Braverman, A. Garg, T. Ma, H. L. Nguyen, and D. P. Woodruff, Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the forty-eighth annual acm symposium on theory of computing*, pp. 1011–1020, 2016

[18] M. Braverman, A. Garg, D. Pankratov, and O. Weinstein, From information to exact communication. In *Proceedings of the 45th annual acm symposium on symposium on theory of computing*, pp. 151–160, ACM, 2013

[19] M. Braverman, S. Garg, and D. P. Woodruff, The coin problem with applications to data streams. In *2020 ieee 61st annual symposium on foundations of computer science (focs)*, pp. 318–329, IEEE, 2020

[20] M. Braverman and J. Mao, Simulating noisy channel interaction. In *Proceedings of the 2015 conference on innovations in theoretical computer science*, pp. 21–30, 2015

[21] M. Braverman and D. Minzer, Optimal tiling of the euclidean space using permutation-symmetric bodies. In *36th computational complexity conference (ccc 2021)*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021

[22] M. Braverman and A. Rao, Information equals amortized communication. In *52nd annual symposium on foundations of computer science (focs)*, pp. 748–757, IEEE, 2011

[23] M. Braverman and A. Rao, Towards coding for maximum errors in interactive communication. In *Stoc*, pp. 159–166, 2011

[24] M. Braverman, A. Rao, O. Weinstein, and A. Yehudayoff, Direct products in communication complexity. In *Foundations of computer science (focs), 2013 ieee 54th annual symposium on*, pp. 746–755, IEEE, 2013

[25] M. Braverman and O. Weinstein, An interactive information odometer with applications. In *Electronic colloquium on computational complexity (eccc)*, 2014

[26] J. Briët and T. Vidick, Explicit lower and upper bounds on the entangled value of multiplayer xor games. *Communications in Mathematical Physics* **321** (2013), no. 1, 181–207

[27] J. Brody, H. Buhrman, M. Kouckỳ, B. Loff, F. Speelman, and N. Vereshchagin, Towards a reverse newman?s theorem in interactive information complexity. *Algorithmica* **76** (2016), no. 3, 749–781

[28] H. Buhrman, R. Cleve, and A. Wigderson, Quantum vs. classical communication and computation. In *Proceedings of the thirtieth annual acm symposium on theory of computing*, pp. 63–68, 1998

[29] A. Buttari, J. Langou, J. Kurzak, and J. Dongarra, A class of parallel tiled linear algebra algorithms for multicore architectures. *Parallel Computing* **35** (2009), no. 1, 38–53

[30] A. Chakrabarti and O. Regev, An optimal lower bound on the communication complexity of gap-hamming-distance. *SIAM Journal on Computing* **41** (2012), no. 5, 1299–1317

[31] A. Chakrabarti, Y. Shi, A. Wirth, and A. Yao, Informational complexity and the direct sum problem for simultaneous message complexity. In *Proceedings of the 42nd annual ieee symposium on foundations of computer science*, edited by B. Werner, pp. 270–278, IEEE Computer Society, Los Alamitos, CA, 2001

[32] D. Chaum, C. Crépeau, and I. Damgard, Multiparty unconditionally secure protocols. In *Proceedings of the twentieth annual acm symposium on theory of computing*, pp. 11–19, 1988

[33] J. Chen, X. Huang, I. A. Kanj, and G. Xia, Linear fpt reductions and computational lower bounds. In *Proceedings of the thirty-sixth annual acm symposium on theory of computing*, pp. 212–221, 2004

[34] S. A. Cook, A taxonomy of problems with fast parallel algorithms. *Information and control* **64** (1985), no. 1-3, 2–22

[35] T. M. Cover and J. A. Thomas, *Elements of information theory, 2nd edition*. J. Wiley and Sons, New York, 2006

[36] J. W. Demmel, M. T. Heath, and H. A. Van Der Vorst, Parallel numerical linear algebra. *Acta numerica* **2** (1993), 111–197

[37] I. Dinur and D. Steurer, Analytical approach to parallel repetition. In *Proceedings of the forty-sixth annual acm symposium on theory of computing*, pp. 624–633, 2014

[38] A. C. Elitzur and L. Vaidman, Quantum mechanical interaction-free measurements. *Foundations of Physics* **23** (1993), no. 7, 987–997

[39] T. Feder, E. Kushilevitz, M. Naor, and N. Nisan, Amortized communication complexity. *SIAM Journal on Computing* **24** (1995), no. 4, 736–750

[40] U. Feige, G. Kindler, and R. O'Donnell, Understanding parallel repetition requires understanding foams. In *Proceedings of the 48th annual ieee symposium on foundations of computer science*, pp. 179–192, 2007

[41] U. Feige and O. Verbitsky, Error reduction by parallel repetition – A negative result. *Combinatorica* **22** (2002), no. 4, 461–478

[42] M. Furst, J. B. Saxe, and M. Sipser, Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory* **17** (1984), no. 1, 13–27

[43] A. Ganor, G. Kol, and R. Raz, Exponential separation of information and communication. In *2014 ieee 55th annual symposium on foundations of computer science*, pp. 176–185, IEEE, 2014

[44] A. Ganor, G. Kol, and R. Raz, Exponential separation of information and communication for boolean functions. *Journal of the ACM (JACM)* **63** (2016), no. 5, 1–31

[45] A. Ganor, G. Kol, and R. Raz, Exponential separation of communication and external information. *SIAM Journal on Computing* **50** (2019), no. 3, STOC16–236

[46] A. Garg, T. Ma, and H. Nguyen, On communication cost of distributed statistical estimation and dimensionality. *Advances in Neural Information Processing Systems* **27** (2014)

[47] R. Gelles, Coding for interactive communication: A survey. *Foundations and Trends® in Theoretical Computer Science* **13** (2017), no. 1–2, 1–157

[48] U. Girish, K. Mittal, R. Raz, and W. Zhan, Polynomial bounds on parallel repetition for all 3-player games with binary inputs. *arXiv preprint arXiv:2204.00858* (2022)

[49] M. Göös, T. Pitassi, and T. Watson, Zero-information protocols and unambiguity in arthur–merlin communication. *Algorithmica* **76** (2016), no. 3, 684–719

[50] U. Hadar, J. Liu, Y. Polyanskiy, and O. Shayevitz, Communication complexity of estimating correlations. In *Proceedings of the 51st annual acm sigact symposium on theory of computing*, pp. 792–803, 2019

[51] J. Hastad, Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual acm symposium on theory of computing*, pp. 6–20, 1986

[52] J. Håstad and A. Wigderson, The randomized communication complexity of set disjointness. *Theory of Computing* **3** (2007), no. 1, 211–219

[53] W. Hesse, E. Allender, and D. A. M. Barrington, Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences* **65** (2002), no. 4, 695–716

[54] T. Holenstein, Parallel repetition: simplifications and the no-signaling case. In *Proceedings of the thirty-ninth annual acm symposium on theory of computing*, pp. 411–419, 2007

[55] D. A. Huffman, A method for the construction of minimum redundancy codes. *Proceedings of the IRE* **40** (1952), no. 9, 1098–1101

[56] R. Impagliazzo and R. Paturi, On the complexity of k-sat. *Journal of Computer and System Sciences* **62** (2001), no. 2, 367–375

[57] R. Impagliazzo, R. Paturi, and F. Zane, Which problems have strongly exponential complexity? *Journal of Computer and System Sciences* **63** (2001), no. 4, 512–530

[58] R. Jain, J. Radhakrishnan, and P. Sen, A lower bound for the bounded round quantum communication complexity of set disjointness. In *44th annual ieee symposium on foundations of computer science, 2003. proceedings.*, pp. 220–229, IEEE, 2003

[59] S. Jukna, *Boolean function complexity: advances and frontiers*. 5, Springer, 2012

[60] B. Kalyanasundaram and G. Schnitger, The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics* **5** (1992), no. 4, 545–557

[61] M. Karchmer, Communication complexity a new approach to circuit depth. 1989

[62] M. Karchmer, R. Raz, and A. Wigderson, Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity* **5** (1995), no. 3/4, 191–204

[63] R. M. Karp, Reducibility among combinatorial problems. In *Complexity of computer computations*, pp. 85–103, Springer, 1972

[64] J. Kilian, A general completeness theorem for two party games. In *Proceedings of the twenty-third annual acm symposium on theory of computing*, pp. 553–560, 1991

[65] G. Kindler, R. O'Donnell, A. Rao, and A. Wigderson, Spherical cubes and rounding in high dimensions. In *2008 49th annual ieee symposium on foundations of computer science*, pp. 189–198, IEEE, 2008

[66] H. Klauck, A. Nayak, A. Ta-Shma, and D. Zuckerman, Interaction in quantum communication. *IEEE Transactions on Information Theory* **53** (2007), no. 6, 1970–1982

[67] G. Kol and R. Raz, Interactive channel capacity. In *Proceedings of the forty-fifth annual acm symposium on theory of computing*, pp. 715–724, 2013

[68] E. Kushilevitz and N. Nisan, *Communication complexity*. Cambridge University Press, Cambridge, 1997

[69] T. Lee and A. Shraibman, Disjointness is hard in the multiparty number-on-the-forehead model. *Computational Complexity* **18** (2009), no. 2, 309–336

[70] T. Lee and A. Shraibman, *Lower bounds in communication complexity*. Now Publishers Inc, 2009

[71] N. Linial, Y. Mansour, and N. Nisan, Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)* **40** (1993), no. 3, 607–620

[72] N. Linial, S. Mendelson, G. Schechtman, and A. Shraibman, Complexity measures of sign matrices. *Combinatorica* **27** (2007), no. 4, 439–463

[73] N. Ma and P. Ishwar, Some results on distributed source coding for interactive function computation. *Information Theory, IEEE Transactions on* **57** (2011), no. 9, 6180–6195

[74] N. Ma and P. Ishwar, The infinite-message limit of two-terminal interactive source coding. *Information Theory, IEEE Transactions on* **59** (2013), no. 7, 4071–4094

[75] A. Madry, Continuous optimization: The ?right? language for graph algorithms?(invited talk). In *36th iarcs annual conference on foundations of software technology and theoretical computer science (fsttcs 2016)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016

[76] A. McGregor, I. Mironov, T. Pitassi, O. Reingold, K. Talwar, and S. Vadhan, The limits of two-party differential privacy. In *2010 ieee 51st annual symposium on foundations of computer science*, pp. 81–90, IEEE, 2010

[77] M. Naor, A. Orlitsky, and P. W. Shor, Three results on interactive communication. *IEEE Trans. Inf. Theory* **39** (1993), no. 5, 1608–1615

[78] I. Newman, Private vs. common random bits in communication complexity. *Information Processing Letters* **39** (1991), no. 2, 67–71

[79] R. O'Donnell, *Analysis of boolean functions*. Cambridge University Press, 2014

[80] A. Orlitsky, Average-case interactive communication. *IEEE Transactions on Information Theory* **38** (1992), no. 5, 1534–1547

[81] A. Orlitsky and A. El Gamal, Communication complexity. In *Complexity in information theory*, pp. 16–61, Springer, 1988

[82] A. Orlitsky and J. R. Roche, Coding for computing. In *Information theory, 1995. proceedings., 1995 ieee international symposium on*, p. 451, IEEE, 1995

[83] M. Patrascu, Towards polynomial lower bounds for dynamic problems. In *Proceedings of the forty-second acm symposium on theory of computing*, pp. 603–610, 2010

[84] A. Rao, Parallel repetition in projection games and a concentration bound. *SIAM Journal on Computing* **40** (2011), no. 6, 1871–1891

[85] A. Rao and A. Yehudayoff, *Communication complexity: and applications*. Cambridge University Press, 2020

[86] R. Raz, A parallel repetition theorem. *SIAM Journal on Computing* **27** (1998), no. 3, 763–803

[87] R. Raz, A counterexample to strong parallel repetition. *SIAM Journal on Computing* **40** (2011), no. 3, 771–777

[88] A. Razborov, On the distributed complexity of disjointness. *TCS: Theoretical Computer Science* **106** (1992)

[89] A. A. Razborov, Lower bounds for the size of circuits of bounded depth with basis $\vee, \oplus$. *Math. notes of the Academy of Sciences of the USSR* **41** (1987), no. 4, 333–338

[90] T. Satoh, K. Kurosawa, and S. Tsujii, Privacy for multi-party protocols. In *International workshop on the theory and application of cryptographic techniques*, pp. 252–260, Springer, 1992

[91] L. J. Schulman, Coding for interactive communication. *IEEE Transactions on Information Theory* **42** (1996), no. 6, 1745–1756

[92] R. Sedgewick and K. Wayne, *Introduction to programming in java: an interdisciplinary approach*. Addison-Wesley Professional, 2017

[93] C. E. Shannon, Two-way communication channels. *Proc. 4th Berkeley Symp. Math. Stat. Prob* **1** (1961), no. 3, 611–644

[94] A. A. Sherstov, The communication complexity of gap hamming distance. *Theory of Computing* **8** (2012), no. 1, 197–208

[95] A. A. Sherstov, The multiparty communication complexity of set disjointness. In *Proceedings of the forty-fourth annual acm symposium on theory of computing*, pp. 525–548, 2012

[96] A. Shpilka and A. Yehudayoff, Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science* **5** (2010), no. 3–4, 207–388

[97] R. Smolensky, Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the nineteenth annual acm symposium on theory of computing*, pp. 77–82, 1987

[98] D. Touchette, Quantum information complexity. In *Proceedings of the forty-seventh annual acm symposium on theory of computing*, pp. 317–326, 2015

[99] A. M. Turing, On computable numbers, with an application to the entscheidungsproblem. *J. of Math* **58** (1936), no. 345-363, 5

[100] H. Tyagi, S. B. Venkatakrishnan, P. Viswanath, and S. Watanabe, Information complexity density and simulation of protocols. *IEEE Transactions on Information Theory* **63** (2017), no. 11, 6979–7002

[101] J. von Neumann, The general and logical theory of automata. In *John von Neumann, collected works*, chap. 9, pp. 288–328, Pergamon Press, 1951

[102] O. Weinstein, Information complexity and the quest for interactive compression. *ACM SIGACT News* **46** (2015), no. 2, 41–64

[103] V. V. Williams, On some fine-grained questions in algorithms and complexity. In *Proceedings of the international congress of mathematicians: Rio de janeiro 2018*, pp. 3447–3487, World Scientific, 2018

[104] A. Wyner and J. Ziv, The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on information Theory* **22** (1976), no. 1, 1–10

[105] A. C. C. Yao, Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the eleventh annual acm symposium on theory of computing*, pp. 209–213, ACM, 1979

[106] A. C. C. Yao, Lower bounds by probabilistic arguments. In *Foundations of computer science, 1983., 24th annual symposium on*, pp. 420–428, IEEE, 1983

[107] A. C.-C. Yao, How to generate and exchange secrets. In *27th annual symposium on foundations of computer science (FOCS 1986)*, pp. 162–167, IEEE, 1986

[108] Y. Zhang, J. Duchi, M. I. Jordan, and M. J. Wainwright, Information-theoretic lower bounds for distributed statistical estimation with communication constraints. *Advances in Neural Information Processing Systems* **26** (2013)

Mark Braverman

Department of Computer Science, Princeton University, Princeton, NJ 08540, USA,
mbraverm@cs.princeton.edu