COMS-E6998: Information Theory in Computer Science **Fall 2017**

Lecture 2: Direct Sums and Interactive Compression

Lecturer: Omri Weinstein *Scribes: scribe-name1,2,3*

2.1 Direct Sums and the Interactive Compression Problem

Direct sum and direct product theorems assert a lower bound on the complexity of solving n copies of a problem in parallel, in terms of the cost of a single copy. Let f^n denote the problem of computing n simultaneous instances of the function f (in some arbitrary computational model for now), and $C(f)$ denote the cost of solving a single copy of f. The obvious solution to f^n is to apply the single-copy optimal solution n times sequentially and independently to each coordinate, yielding a linear scaling of the resources, so clearly $C(f^n) \leq n \cdot C(f)$. The *strong direct sum* conjecture postulates that this naive solution is essentially optimal. In the context of randomized communication complexity, the strong direct sum conjecture informally asks whether it is true that for any function f and input distribution μ,

$$\mathsf{D}_{\mu^n}(f^n, \varepsilon) =^? \Omega(n) \cdot \mathsf{D}_\mu(f, \varepsilon). \tag{2.1}$$

More generally, direct sum theorems aim to give an (ideally linear in n, but possibly weaker) lower bound on the communication required for computing f^n with some *constant overall* error $\varepsilon > 0$ in terms of the cost of computing a single copy of f with the same (or comparable) fixed error.

A *direct product* theorem further asserts that unless sufficient resources are provided, the probability of successfully computing all n copies of f will be exponentially small, potentially as low as $(1-\varepsilon)^{\Omega(n)}$. This is intuitively plausible, since the naive solution which applies the best (ε-error) protocol for one copy of f independently to each of the n coordinates, would indeed succeed in solving f^n with probability $(1-\varepsilon)^n$. Is this naive solution optimal?

To make this more precise, let us denote by $\mathsf{suc}(\mu, f, C)$ the maximum success probability of a protocol with communication complexity $\leq C$ in computing f under input distribution μ. A direct product theorem asserts that any protocol attempting to solve f^n (under μ^n) using some number T of communication bits (ideally $T = \Omega(n \cdot C)$), will succeed only with exponentially small probability: $\mathsf{suc}(\mu^n, f^n, T) \lesssim (1-\varepsilon)^{\Omega(n)}$. Informally, the strong direct product question asks whether

$$\mathsf{suc}(\mu^n, f^n, o(n \cdot C)) \lesssim^? (\mathsf{suc}(\mu, f, C))^{\Omega(n)}. \tag{2.2}$$

Note that (2.2) in particular implies (2.1) when setting $C = \mathsf{D}_\mu(f, \varepsilon)$. Classic examples of direct product results in complexity theory are Raz's Parallel Repetition Theorem [Raz98, Rao08a] and Yao's XOR Lemma [Yao82] (For more examples and a broader overview of the rich history of direct sum and product theorems see [JainPP12] and references therein). The value of such results to computational complexity is clear: direct sum and product theorems, together with a lower bound on the (easier-to-reason-about) "primitive" problem, yield a lower bound on the composite problem in a "black-box" fashion (a method also known as *hardness amplification*). For example, the Karchmer-Raz-Wigderson approach for separating $\mathbf{P}$ from $\mathbf{NC}^1$ can be completed via a (still open) direct sum conjecture for Boolean formulas [KRW95] (after more than a decade, some progress on this conjecture was recently made using information-complexity machinery [GMWW14]). Other fields in which direct sums and products have played a central role in proving tight lower bounds are streaming [BaryossefJKS04, ST13, MWY13, GO13] and distributed computing [HRVZ13].

Can we always hope for such strong lower bounds to hold? It turns out that the validity of these conjectures highly depends on the underlying computational model, and the short answer is no.[1] In the communication complexity model, this question has had a long history and was answered positively for several restricted models of communication [Klauck10, Shaltiel03,LSS08,Sherstov12, JainPP12,MWY13,ParnafesRW97]. Interestingly, in the *determistic* communication complexity model, Feder et al. [FederKNN95] showed that

$$\mathsf{D}(f^n) \geq n \cdot \Omega\left(\sqrt{\mathsf{D}(f)}\right)$$

for any two-party Boolean function f (where $\mathsf{D}(f)$ stands for the deterministic communication complexity of f), but this proof completely breaks when protocols are allowed to err. Indeed, in the randomized communication model, there is a tight connection between the direct sum question for the function f and its information complexity. By now, this should come as no surprise: The "IC = Amortized CC" Theorem asserts that, for large enough n, the communication complexity of f^n scales linearly with the (single-copy) information cost of f, i.e. $\mathsf{D}_{\mu^n}(f^n,\varepsilon) = \Theta\left(n \cdot \mathsf{IC}_\mu(f,\varepsilon)\right)$, and hence the strong direct sum question (2.1) boils down to understanding the relationship between the single-copy measures $\mathsf{D}_\mu(f,\varepsilon)$ and $\mathsf{IC}_\mu(f,\varepsilon)$. Indeed, it can be formally shown ([BravermanR11]) that the direct sum problem is equivalent [2] to the following problem of "one-shot" compression of interactive protocols:

Problem 2.1.1 (Interactive compression problem, [BBCR])**.** *Given a protocol π over inputs $x, y \sim \mu$, with $\|\pi\| = C, \mathsf{IC}_\mu(\pi) = I$, what is the smallest amount of communication of a protocol τ which (approximately) simulates π (i.e., $\exists\ g$ s.t $|g(\tau(x,y)) - \pi(x,y)|_1 \leq \delta$ for a small constant δ)?*

In particular, if one could compress any protocol into $O(I)$ bits, this would have shown that $\mathsf{D}_\mu(f,\varepsilon) = O\left(\mathsf{IC}_\mu(f,\varepsilon)\right)$ which would in turn imply the strong direct sum conjecture. In fact, the additivity of information cost (Lemma **??** from Section **??**) implies the following general quantitative relationship between (possibly weaker) interactive compression results and direct sum theorems in communication complexity:

Proposition 2.1.2 (One-Shot Compression implies Direct Sum)**.** *Suppose that for any $\delta > 0$ and any given protocol π for which $\mathsf{IC}_\mu(\pi) = I$, $\|\pi\| = C$, there is a compression scheme that δ-simulates[3] π using $g_\delta(I,C)$ bits of communication. Then*

$$g_\delta\left(\frac{\mathsf{D}_{\mu^n}(f^n,\varepsilon)}{n}, \mathsf{D}_{\mu^n}(f^n,\varepsilon)\right) \geq \mathsf{D}_\mu(f,\varepsilon+\delta).$$

Proof. Let Π be an optimal n-fold protocol for f^n under μ^n with per-copy error ε, i.e., $\|\Pi\| = \mathsf{D}_{\mu^n}(f^n,\varepsilon) := C_n$. By Lemma **??** (equation (**??**)), there is a single-copy ε-error protocol θ for computing $f(x,y)$ under μ, whose information cost is at most $\mathsf{IC}_{\mu^n}(\Pi)/n \leq C_n/n$ (since communication always upper bounds information). By assumption of the claim, θ can now be δ-simulated using $g_\delta(C_n/n, C_n)$ communication, so as to produce a single-copy protocol with error $\leq \varepsilon + \delta$ for f, and therefore $\mathsf{D}_\mu(f,\varepsilon+\delta) \leq g_\delta(C_n/n\ ,\ C_n)$. □

The first general interactive compression result was proposed in the seminal work of Barak, Braverman, Chen and Rao [BBCR], who showed that any protocol π can be δ-simulated using $g_\delta(I,C) = \tilde{O}_\delta(\sqrt{C \cdot I})$

[1] In the context of circuit complexity, for example, this conjecture fails (at least in its strongest form): Multiplying an $n \times n$ matrix by a (worst case) n-dimensional vector requires n^2 operations, while (deterministic) multiplication of n different vectors by the same matrix amounts to matrix-multiplication of two $n \times n$ matrices, which can be done in $n^{2.37} \ll n^3$ operations [Williams12].

[2] The exact equivalence of the direct sum conjecture and Problem 2.1.1 holds for *relations* (Theorem 6.6 in [BravermanR11]). For total functions, one could argue that the requirement in Problem 2.1.1 is too harsh as it requires simulation of the entire transcript of the protocol, while in the direct sum context for functions we are merely interested in the output of f. However, all known compression protocols satisfy the stronger requirement and no separation is known between those techniques.

[3] The simulation here is in an internal sense, namely, Alice and Bob should be able to reconstruct the transcript of the original protocol (up to a small error), based on public randomness and their own private inputs. See [BRWY12] for the precise definition and the (subtle) role it plays in context of direct product theorems.

communication (we prove this result in Section 2.2.1). Plugging this compression result into Proposition 2.1.2, this yields the following weaker direct sum theorem:

Theorem 2.1.3 (Weak Direct Sum, [BBCR])**.** *For every Boolean function f, distribution μ, and any positive constant $\delta > 0$,*

$$\mathsf{D}_{\mu^n}(f^n, \varepsilon) \geq \tilde{\Omega}(\sqrt{n} \cdot \mathsf{D}_{\mu}(f, \varepsilon + \delta)).$$

Later, Braverman [Bra12] showed that it is always possible to simulate π using $2^{O_\delta(I)}$ bits of communication. This result is still far from ideal compression ($O(I)$ bits), but it is nevertheless appealing as it show that any protocol can be simulated using amount of communication which depends solely on its information cost, but *independent* of its original communication which may have been arbitrarily larger (we prove this result in Section 2.2.2). Notice that the last two compression results are indeed incomparable, since the communication of π could be much larger than its information complexity (e.g., $C \geq 2^{2^{2^I}}$). The current state of the art for the *general* interactive compression problem can be therefore summarized as follows: Any protocol with communication C and information cost I can be compressed to

$$g_\delta(I, C) \leq \min\left\{2^{O_\delta(I)} \ , \ \tilde{O}_\delta(\sqrt{I \cdot C})\right\} \tag{2.3}$$

bits of communication.

The above results may seem as a plausible evidence that it is in fact possible to compress general interactive protocols all the way down to $O(I)$ bits. Unfortunately, this task turns out to be too ambitious: In a recent breakthrough result, Ganor, Kol and Raz [GKR14] proved the following lower bound on the communication of any compression scheme:

$$g_\delta(I, C) \geq \max\left\{2^{\Omega(I)} \ , \ \tilde{\Omega}(I \cdot \log C)\right\}. \tag{2.4}$$

More specifically, they exhibit a Boolean function f which can be solved using a protocol with information cost I, but cannot be simulated by a protocol π' with communication cost $< 2^{\Omega(I)}$ (a simplified construction and proof was very recently obtained by Rao and Sinha [RaoS15]). Since the *communication* of the low information protocol they exhibit is $\sim 2^{2^I}$, this also rules out a compression to $I \cdot o(\log C)$, or else such compression would have produced a too good to be true ($2^{o(I)}$ communication) protocol. The margin of this text is too narrow to contain the proof of this separation result, but it is noteworthy that proving it was particularly challenging: It was shown that essentially all previously known techniques for proving communication lower bounds apply to information complexity as well [BW12,KLL], and hence could not be used to separate information complexity and communication complexity. Using (the reverse direction of) Proposition 2.1.2 (see Theorem 6.6 in [BravermanR11]), the compression lower bound in (2.4) refutes the strongest possible direct sum (2.1), but leaves open the following gap

$$\tilde{\Omega}_\delta\left(\sqrt{n}\right) \ \leq \ \min_f \frac{\mathsf{D}_{\mu^n}(f^n, \varepsilon)}{\mathsf{D}_{\mu}(f, \varepsilon + \delta)} \ \leq \ O\left(\frac{n}{\log n}\right). \tag{2.5}$$

Notice that this still leaves the direct sum conjecture for randomized communication complexity wide open: It is still conceivable that improved compression to $g_\delta(I, C) = I \cdot C^{o(1)}$ is in fact possible, and the quest to beat the compression scheme of [BBCR] remains unsettled.[4]

Despite the lack of progress in the general regime, several works showed that it is in fact possible to obtain near-optimal compression results in restricted models of communication: When the input distribution μ is a *product distribution* (x and y are independent), [BBCR] show a near-optimal compression result, namely

[4] Ramamoorthy and Rao [RR15] recently showed that BBCR's compression scheme can be improved when the underlying communication protocol is *asymmetric*, i,e., when Alice reveals much more information than Bob.

that π can be compressed into $O(I \cdot polylog(C))$ bits.[5] Once again, using Proposition 2.1.2 this yields the following direct sum theorem:

Theorem 2.1.4 ([BBCR]). *For every product distribution μ and any $\delta > 0$,*

$$\mathsf{D}_{\mu^n}(f^n, \varepsilon) = \tilde{\Omega}(n \cdot \mathsf{D}_\mu(f, \varepsilon + \delta)).$$

Improved compression results were also proven for *public-coin protocols* (under arbitrary distributions) [BBKLSV13, BMY14], and for bounded-round protocols, leading to near-optimal direct sum theorems in corresponding communication models. We summarize these results in Table 2.1.

Reference	Regime	Communication Complexity
[HJMR07]	r-round protocols, product distributions	$I + O(r)$
[BravermanR11, BRWY13b]	r-round protocols	$I + O\left(\sqrt{r \cdot I}\right) + O(r \log 1/\delta)$
[BMY14] (improved [BBKLSV13])	Public coin protocols	$O(I^2 \cdot \log\log(C)/\delta^2)$
[BBCR]	Product distributions	$O(I \cdot poly \log(C)/\delta)$
[Bra12, BBCR]	**General protocols**	$\min\{2^{O(I/\delta)}\ ,\ O(\sqrt{I \cdot C} \cdot \log(C)/\delta)\}$
[GKR14,RaoS15]	**Best lower bound**	$\max\{2^{\Omega(I)}\ ,\ \Omega(I \cdot \log(C))$

Table 2.1: Best to date compression schemes, for various regimes. Notice that in the general regime (last two columns), in terms of dependence on the original communication C, the gap is still very large ($\Omega(\log C)$ vs. $\tilde{O}(C^{1/2})$).

2.2 State of the Art Interactive Compression Schemes

In this section we present the two state-of-the-art compression schemes for unbounded-round communication protocols, the first due to Barak et al., and the second due to Braverman [BBCR, Bra12]. As mentioned in the introduction, a natural idea for compressing a multi-round protocol is to try and compress each round separately, using ideas from the transmission (one-way) setup [huffman1952method, HJMR07, BravermanR11]. Such compression suffers from one fatal flaw: It would inevitably require sending at least 1 bit of communication at each round, while the information revealed in each round may be $\ll 1$ (an instructive example is the protocol in which Alice sends Bob, at each round of the protocol, an independent coin flip which is ε-biased towards her input $X \sim Ber(1/2)$, for $\varepsilon \ll 1$). Thus any attempt to implement the compression on a round- by-round basis is hopeful only when the number of rounds is bounded but is doomed to fail in general (indeed, this is the essence of the bounded-round compression schemes of [BravermanR11, BRWY13b]).

The main feature of the compression results we present below is that they do not depend on the number of rounds of the underlying protocol, but only on the overall communication and information cost.

2.2.1 Barak et al.'s compression scheme

Theorem 2.2.1 ([BBCR]). *Let π be a protocol executed over inputs $x, y \sim \mu$, and suppose $\mathsf{IC}_\mu(\pi) = I$ and $\|\pi\| = C$. Then for every $\varepsilon > 0$, there is a protocol τ which ε-simulates π, where*

$$\|\tau\| = O\left(\sqrt{C \cdot I} \cdot (\log(C/\varepsilon)/\varepsilon)\right). \tag{2.6}$$

[5] These compression results in fact hold for general (non-product) distributions as well, when compression is with respect to I^{ext}, the external information cost of the original protocol π (which may be significantly larger than I).

Proof. The conceptual idea underlying this compression result is using public randomness to *avoid communication by trying to guess what the other player is about to say.* Informally speaking, the players will use shared randomness to sample (correlated) *full paths* of the protocol tree, according to their private knowledge: Alice has the "correct" distribution on nodes that she owns in the tree (since conditioned on reaching these nodes, the next messages only depend on her input x), and will use her "best guess" (i.e., her prior distribution on Bob's next message, under μ, her input x and the history of messages) to sample messages at nodes owned by Bob. Bob will do the same on nodes owned by Alice. This "guessing" is done in a correlated way using public randomness (and no communication whatsoever (!)), in a way that guarantees that if the player's guesses are close to the correct distribution, then the probability that they sample the same bit is large.

The above step gives rise to two paths, P_A and P_B respectively. In the the next step, the players will use (mild) communication to find all inconsistencies among P_A and P_B and correct them one by one (according to the "correct" speaker). By the end of this process, the players obtain a consistent path which has the correct distribution $\Pi(x, y)$. Therefore, the overall communication of the simulating protocol would be comparable to the number of mistakes between P_A and P_B (times the communication cost of fixing each mistake). Intuitively, the fact that π has low information will imply that the number of inconsistencies is small, as inconsistent samples on a given node typically occur when the "receiver's" prior distribution is far from the "speaker's" correct distributions, which will in turn imply that this bit conveyed a lot of information to the receiver (Alas, we will see that if the information revealed by the i'th bit of π is ε, then the probability of making a mistake on the i'th node is $\approx \sqrt{\varepsilon}$, and this is the source of sub-optimality of the above result. We discuss this bottleneck at the end of the proof).

We now sketch the proof more formally (yet still leaving out some minor technicalities). Let $\Pi = M_1, \dots, M_C$ denote the transcript of π. Each node w at depth i of the protocol tree of π is associated with two numbers, $p_{x,w}$ and $p_{y,w}$, describing the probability (according to each player's respective "belief") that conditioned on reaching w, the next bit sent in π is "1" (the right child of w). That is,

$$p_{x,w} := \Pr[M_i = 1 \mid xrM_{<i} = w] \quad \text{, and} \quad p_{y,w} := \Pr[M_i = 1 \mid yr, M_{<i} = w]. \tag{2.7}$$

Note that if w is owned by the Alice, then $p_{x,w}$ is exactly the correct probability with which the i-th bit is transmitted in π, conditioned that π has reached w.

In the simulating protocol τ, the players first sample, without communication and using public randomness, a uniformly random number ρ_w in the interval $[0, 1]$, for every node w of the protocol tree[6]. For simplicity of analysis, in the rest of the proof we assume the public randomness is fixed to the vale $R = r$. Alice and Bob now privately construct the following respective trees $\mathcal{T}_A, \mathcal{T}_B$: For each node w, Alice includes the right child of w in $\mathcal{T}_A$ iff $p_{w,x} < \rho_w$, and the left child ("0") otherwise. Bob does the same by including the right child of w in $\mathcal{T}_B$ iff $p_{w,y} < \rho_w$.

The trees $\mathcal{T}_A$ and $\mathcal{T}_B$ define a unique path $\ell = m_1, \dots, m_C$ of π, by combining outgoing edges from $\mathcal{T}_A$ in nodes owned by Alice, and edges from $\mathcal{T}_B$ in nodes owned by Bob. Note that ℓ has precisely the desired distribution of $\Pi(X, Y)$. To identify ℓ, the players will now find the inconsistencies among $\mathcal{T}_A$ and $\mathcal{T}_B$ and correct them one by one.

We say that a *mistake* occurs in level i if the outgoing edges of m_{i-1} in $\mathcal{T}_A$ and $\mathcal{T}_B$ are inconsistent. Finding the (first) mistake of τ amounts to finding the first differing index among two C-bit strings (corresponding to the paths P_A and P_B induced by $\mathcal{T}_A$ and $\mathcal{T}_B$). Luckily, there is a randomized protocol which accomplishes this task with high probability $(1-\gamma)$ using only $O(log(C/\gamma))$ bits of communication, using a clever "noisy" binary search due to Feige et al. [FeigePRU94]. Since errors accumulate over C rounds and we are aiming for an overall simulation error of ε, we will set $\gamma \approx \varepsilon/C$, thus the cost of fixing each inconsistency remains

[6] Note that there are exponentially many nodes, but the communication model does not charge for local computations or the amount of shared randomness, so these resources are indeed "for free".

$O(\log(C/\varepsilon))$ bits. The expected communication complexity of τ (over X, Y, R) is therefore

$$\mathbb{E}[\|\tau\|] = \mathbb{E}[\#\text{ mistakes of } \tau] \cdot O(\log(C/\varepsilon)). \tag{2.8}$$

Though we are not quite done, one should appreciate the simplicity of analysis of the cost of this protocol. The next lemma completes the proof, asserting that the expected number of mistakes τ makes is not too large:

Lemma 2.2.2. $\mathbb{E}[\#$ *mistakes of* $\tau] \le \sqrt{C \cdot I}$.

Indeed, substituting the assertion of Lemma 2.2.2 into (2.8), we conclude that the expected communication complexity of τ is $O(\sqrt{C \cdot I} \cdot poly \log(C/\varepsilon))$, and a standard Markov bound yields the bound in (2.6) and therefore finishes the proof of Theorem 2.2.1.

Proof of Lemma 2.2.2. Let $\mathcal{E}_i$ be the indicator random variable denoting whether a mistake has occurred in step i of the protocol tree of π. Hence the expected number of mistakes is $\sum_{i=1}^{C} \mathcal{E}_i$. We shall bound each term $\mathbb{E}[\mathcal{E}_i]$ separately. By construction, a mistake at node w in level i occurs exactly when either $p_{x,w} < \rho_w < p_{y,w}$ or $p_{y,w} < \rho_w < p_{x,w}$. Since ρ_w was uniform in $[0,1]$, the probability of a mistake is

$$|p_{x,w} - p_{y,w}| = |(M_i|x, r, M_{<i} = w) - (M_i|y, r, M_{<i} = w)|,$$

where the last transition is by definition of $p_{x,w}$ and $p_{y,w}$. Note that, by definition of a protocol, if $w := m_{<i}$ is owned by Alice, then $M_i|xyrm_{<i}] = M_i|xyrm_{<i}$ and if it is owned by Bob, then $M_i|y, r, m_{<i} = M_i|x, y, r, m_{<i}$. We therefore have

$$\begin{align}
\mathbb{E}[\mathcal{E}_i] &= \mathbb{E}_{xym_{<i}\sim\pi}[|(M_i|xrm_{<i}) - (M_i|yrm_{<i})|] \notag\\
&\le \mathbb{E}_{xym_{<i}\sim\pi}\left[\max\{|(M_i|xyrm_{<i}) - (M_i|xrm_{<i})|\ ,\ |(M_i|xyrm_{<i}) - (M_i|yrm_{<i})|\}\right] \notag\\
&\le \mathbb{E}_{xym_{<i}\sim\pi}\left[\sqrt{\mathbb{D}\left(M_i|xyrm_{<i}\|M_i|xrm_{<i}\right) + \mathbb{D}\left(M_i|xyrm_{<i}\|M_i|yrm_{<i}\right)}\right] \tag{2.9}\\
&\le \sqrt{\mathbb{E}_{xym_{<i}\sim\pi}\left[\mathbb{D}\left(M_i|xyrm_{<i}\|M_i|xrm_{<i}\right) + \mathbb{D}\left(M_i|xyrm_{<i}\|M_i|yrm_{<i}\right)\right]} \tag{2.10}\\
&= \sqrt{I(M_i; X|M_{<i}RY) + I(M_i; Y|M_{<i}RX)} \tag{2.11}
\end{align}$$

where transition (2.9) follows from Pinsker's inequality (Lemma **??**), transition (2.10) follows from the convexity of $\sqrt{\cdot}$, and the last transition is by Proposition **??**.

Finally, by linearity of expectation and the Cauchy-Schwartz inequality, we conclude that

$$\begin{aligned}
\mathbb{E}\left[\sum_{i=1}^{C} \mathcal{E}_i\right] &\le \sum_{i=1}^{C} \sqrt{I(M_i; X|M_{<i}RY) + I(M_i; Y|M_{<i}RX)}\\
&\le \sqrt{\left(\sum_{i=1}^{C} 1\right) \cdot \left(\sum_{i=1}^{C} I(M_i; X|M_{<i}RY) + I(M_i; Y|M_{<i}RX)\right)}\\
&= \sqrt{C \cdot I}
\end{aligned}$$

where the last transition is by the chain rule for mutual information. □

□

A natural question arising from the above compression scheme is whether the analysis in Lemma 2.2.2 is tight. Unfortunately, the answer is yes, as demonstrated by the following example: Suppose Alice has a uniform C-bit string $X_1 \dots X_C$ where $X_i \sim Ber(1/2)$, and consider the C-bit protocol in which Alice sends, at each round i, an independent sample M_i such that

$$M_i \sim \begin{cases} Ber(1/2+\varepsilon) & \text{if } X_i = 1 \\ Ber(1/2-\varepsilon) & \text{if } X_i = 0 \end{cases}$$

for $\varepsilon = 1/\sqrt{C}$. Since Bob has a perfectly uniform prior on X, a direct calculation shows that in this case $I(M_i; X|M_{<i}) = I(M_i; X) = \mathbb{D}\left(Ber(1/2+\varepsilon)\|Ber(1/2)\right) = O(\varepsilon^2)$, so the total information cost of the protocol is $O(C \cdot \varepsilon^2) = O(1)$. On the other hand, the probability of making a "mistake" at step i of the simulation above is the total variation distance $|Ber(1/2+\varepsilon) - Ber(1/2)| \approx \varepsilon$. Therefore, the expected number of mistakes conditioned on, say, $X_1 = \ldots = X_C = 1$, is $\approx C \cdot \varepsilon = \sqrt{C}$, by choice of $\varepsilon = 1/\sqrt{C}$. I.e., this example shows that both Pinsker's and the Cauchy-Schwartz inequalities are tight in the extreme case where each of the C bit of π reveals $\approx I/C$ bits of information. In the next section we present a different compression scheme which can do better in this regime, at least when I is much smaller than C.

2.2.2 Braverman's compression scheme

Theorem 2.2.3 ([Bra12]). *Let π be a protocol executed over inputs $x, y \sim \mu$, and suppose $\mathsf{IC}_\mu(\pi) = I$. Then for every $\varepsilon > 0$, there is a protocol τ which ε-simulates π, where $\|\tau\| = 2^{O(I/\varepsilon)}$.*

Proof. To understand this result, it will be useful to view the interactive compression problem as the following correlated sampling task: Denote by π_{xy} the distribution of the transcript $\Pi(x, y)$, and by π_x (resp. π_y) the conditional marginal distribution $\Pi|x$ ($\Pi|y$) of the transcript from Alice's (Bob's) point of view (for notational ease, the conditioning on the public randomness r of the protocol is included here implicitly. Note that in general π is still randomized even conditioned on x, y, since it may have private randomness). By the product structure of communication protocols, the probability of reaching a leaf (path) $\ell \in \{0,1\}^C$ of π is

$$\pi_{xy}(\ell) = p_x(\ell) \cdot p_y(\ell) \tag{2.12}$$

where $p_x(\ell) = \prod_{w \subseteq \ell, w \text{ odd}} p_{x,w}$ is the product of the transition probabilities defined in (2.7) on the nodes owned by Alice along the path from the root to ℓ, and $\pi_y(\ell)$ is analogously defined on the even nodes. Thus, the desirable distribution from which the players wish to jointly sample, decomposes to a natural product distribution [7]. Similarly,

$$\pi_x(\ell) = p_x(\ell) \cdot q_x(\ell) \qquad \text{and} \qquad \pi_y(\ell) = q_y(\ell) \cdot p_y(\ell) \tag{2.13}$$

where $q_x(\ell) = \prod_{w \subseteq \ell, w \text{ even}} p_{x,w}$ is Alice's prior "belief" on the *even nodes* owned by Bob along the path to ℓ (see (2.7)), and $q_y(\ell) = \prod_{w \subseteq \ell, w \text{ odd}} p_{y,w}$ is Bob's prior belief on the odd nodes owned by Alice. Thus, the player's goal is to sample $\ell \sim \pi_{x,y}$, where Alice has the correct distribution on odd nodes (and only an estimate on the odd ones), and Bob has the correct distribution on even nodes (and an estimate on the even ones).

We claim that the information cost of π being low (I bits) implies that Alice's prior "belief" q_x on the even nodes owned by Bob, is "close" to the true distribution p_y on these nodes (and vice versa for q_y and

[7] As we shall see, the rejection sampling approach of the compression protocol below crucially exploits this product structure of the target distribution, and it is curious to note this simplifying feature of interactive compression as opposed to general correlated sampling tasks.

p_x on the odd nodes). To see this, recall the equivalent interpretation of mutual information in terms of KL-divergence:

$$\begin{aligned} I = I(\Pi; X|Y) + I(\Pi; Y|X) &= \mathbb{E}_{(x,y)\sim\mu}\left[\mathbb{D}\left(\pi_{xy}\|\pi_y\right) + \mathbb{D}\left(\pi_{xy}\|\pi_x\right)\right] \\ = \mathbb{E}_{x,y,\ell\sim\pi_{x,y}}\left[\log\frac{\pi_{xy}(\ell)}{\pi_y(\ell)} + \log\frac{\pi_{xy}(\ell)}{\pi_x(\ell)}\right] &= \mathbb{E}_{x,y,\ell\sim\pi_{x,y}}\left[\log\frac{p_x(\ell)}{q_y(\ell)} + \log\frac{p_y(\ell)}{q_x(\ell)}\right], \end{aligned} \tag{2.14}$$

where the last transition follows from substituting the terms according to (2.12) and (2.13). The above equation asserts that the typical log-ratio p_x/q_y is at most I, and the same holds for p_y/q_x. The following simple corollary essentially follows from Markov's inequality[8], so we state it without a proof.

Corollary 2.2.4. *Define the set of transcripts* $B_\varepsilon := \{\ell : p_x(\ell) > 2^{(I+1)/\varepsilon} \cdot q_y(\ell) \;\; \textit{or} \;\; p_y(\ell) > 2^{(I+1)/\varepsilon} \cdot q_x(\ell)\ \}$. *Then* $\pi_{x,y}(B_\varepsilon) < \varepsilon$.

The intuitive operational interpretation of the above claim is that, for almost all transcripts ℓ, the following holds: If a *uniformly random* point $\in [0,1]$ falls below $p_y(\ell)$, then the probability it falls below q_x as well is $\gtrsim 2^{-I}$. This intuition gives rise to the following rejection sampling approach: The players interpret the public random tape as a sequence of points $(\ell_i, \alpha_i, \beta_i)$, uniformly distributed in $\mathcal{U} \times [0,1] \times [0,1]$, where $\mathcal{U} = \{0,1\}^C$ is the set of all possible transcripts of π. Their goal will be to discover the first index i^* of a transcript ℓ_i that satisfies $\alpha_{i^*} \le p_x(\ell_{i^*})$ and $\beta_{i^*} \le p_y(\ell_{i^*})$. Note that, by design, the probability that a randomly sampled transcript ℓ_i satisfies these conditions is precisely $p_x(\ell_i) \cdot p_y(\ell_i) = \pi_{xy}(\ell_i)$, and therefore ℓ_{i^*} has the correct distribution.

The players consider only the first $t := 2|\mathcal{U}|\ln(1/\varepsilon)$ points of the public tape, as the probability that a single node satisfies the desirable condition is exactly $1/|\mathcal{U}|$, and thus by independence of the points, the probability that $i^* > t$ is at most $(1 - 1/|\mathcal{U}|)^t = \varepsilon^2 < \varepsilon/16)$.

In order to discover the index of the first "legal" transcript (i^*), each player defines his own set of "potential candidates" for the index i^*. Alice defines the set

$$\mathcal{A} := \{i < T \;:\; \alpha_i \le p_x(\ell_i) \;\text{ and }\; \beta_i \le 2^{8I/\varepsilon} \cdot q_x(\ell_i)\}.$$

Thus $\mathcal{A}$ is the set of transcript which have the correct distribution on the odd nodes (which Alice can verify by herself), and "approximately" satisfies the desirable condition on the even nodes, on which Alice only has a prior estimate (q_x). Similarly, Bob defines

$$\mathcal{B} := \{i < t \;:\; \beta_i \le p_y(\ell_i) \;\text{ and }\; \alpha_i \le 2^{8I/\varepsilon} \cdot q_y(\ell_i)\}.$$

By Corollary 2.2.4, $\Pr[\ell^* \notin \mathcal{A} \cap \mathcal{B}] \le \varepsilon/8$, so for the rest of the proof we assume that $\ell^* \in \mathcal{A} \cap \mathcal{B}$. In fact, ℓ^* is the first element of $\mathcal{A} \cap \mathcal{B}$. Note that for each point $(\ell_i, \alpha_i, \beta_i)$, $\Pr[\ell_i \in \mathcal{A} \cap \mathcal{B}] \le 2^{8I/\varepsilon}/|\mathcal{U}|$. Since we consider only the first $t = 2|\mathcal{U}|\ln(1/\varepsilon)$ points, this implies $\mathbb{E}[|\mathcal{A}|] \le 2^{8I/\varepsilon} \cdot 2\ln(1/\varepsilon)$, and Chernoff bound further asserts that

$$\Pr[|\mathcal{A}| > 2^{10I/\varepsilon}] \ll \varepsilon/16.$$

Thus, if we let $\mathcal{E}_1$ denote the event that $\ell^* \notin \mathcal{A} \cap \mathcal{B}$, and $\mathcal{E}_2 := \{i^* > t \text{ or } |\mathcal{A}| > 2^{10I/\varepsilon} \text{ or} |\mathcal{B}| > 2^{10I/\varepsilon}\ \}$, then by a union bound $\Pr[\mathcal{E}_1 \cup \mathcal{E}_2] \le 2\varepsilon/8 + 3\varepsilon/16 < \varepsilon/2$. Thus, letting $\tau_{x,y}$ denote the distribution of $\ell_{i^*}|\neg(\mathcal{E}_1 \cup \mathcal{E}_2)$, the above implies

$$|\tau_{x,y} - \pi_{x,y}| \le \varepsilon/2,$$

as desired. We will now show a (2-round) protocol τ in which Alice and Bob output a leaf $\ell \sim \tau_{x,y}$, thereby completing the proof. To this end, note we have reduced the simulation task to the problem of finding and

[8] One needs to be slightly careful, since the log ratios can in fact be negative, while Markov's inequality applies only to non-negative random variables. However, it is well known that the contribution of the negative summands is bounded, see [Bra12] for a complete proof.

outputting the first element in $\mathcal{A}\cap\mathcal{B}$, where $|\mathcal{A}| \le 2^{10I/\varepsilon}$ and $|\mathcal{B}| \le 2^{10I/\varepsilon}$. The idea is simple: Alice wishes to send her entire set $\mathcal{A}$ to Bob, who can then check for intersection with his set $\mathcal{B}$. Alas, explicitly sending each element $\ell \in \mathcal{A}$ may be too expensive (requires $\log|\mathcal{U}|$ bits), so instead Alice will send Bob sufficiently many ($O(I/\varepsilon)$) random hashes of the elements in $\mathcal{A}$, using a publicly chosen sequence of hash functions. Since for $a \in \mathcal{A}$ and $b \in \mathcal{B}$ such that $a \neq b$, the probability (over the choice of the hash functions) that $h_j(a) = h_j(b)$ for all $j \in O(I/\varepsilon)$ is bounded by $2^{-O(I/\varepsilon)} < \frac{\varepsilon}{4|\mathcal{A}|\cdot|\mathcal{B}|}$, a union bound ensures that the probability there is an $a \in \mathcal{A}$, $b \in \mathcal{B}$ such that $a \neq b$ but the hashes happen to match, is bounded by $\varepsilon/4$, which completes the proof. For completeness, the protocol τ is described in Figure 2.1.

The simulation protocol τ

1. Alice computes the set $\mathcal{A}$. If $|\mathcal{A}| > 2^{10I/\varepsilon}$ the protocol fails.
2. Bob computes the set $\mathcal{B}$. If $|\mathcal{B}| > 2^{10I/\varepsilon}$ the protocol fails.
3. For each $a \in \mathcal{A}$, Alice computes $d = \lceil 20I/\varepsilon + \log 1/\varepsilon + 2\rceil$ random hash values $h_1(a), \ldots, h_d(a)$, where the hash functions are evaluated using public randomness.
4. Alice sends the values $\{h_j(a_i)\}_{a_i\in\mathcal{A},\ 1\le j\le d}$ to Bob.
5. Bob finds the first index i such that there is a $b \in \mathcal{B}$ for which $h_j(b) = h_j(a_i)$ for $j = 1..d$ (if such an i exists). Bob outputs ℓ_b and sends the index i to Alice.
6. Alice outputs ℓ_i.

Figure 2.1: A simulating protocol for sampling a transcript of $\pi(x, y)$ using $2^{O(I/\varepsilon)}$ communication.

□

References