COMP760, SUMMARY OF LECTURE 17.

HAMED HATAMI

1. How to do compression?

In this lecture we will discuss how to compress a protocol with low information cost, but possibly very high communication cost, to a protocol with low communication cost. It will be important to understand how much information is revealed at every step of the protocol. Thus we will take a closer look at the protocol tree, and we shall try to break the information cost into an expression in terms of the amounts of the information that are revealed nodes of the tree.

Consider a protocol π, and fix the public randomness to $R = r$ (recall that we denoted this protocol by π_r) so that there is no public randomness anymore. Note that every internal node v of the protocol tree at distance j from the root corresponds to a value for the partial transcript $\Pi_{\leq j}$. Let L_j denote the set of the internal nodes at distance j from the root, V_{int} denote the internal nodes of the tree, and let ℓ be the number of rounds of the protocol. Using the formula $I(A;B|C) = \mathbb{E}_{c\sim C} I(A;B|C=c)$, we have

$$\begin{aligned} I(X;\Pi|Y) &= I(X;\Pi_1|Y) + I(X;\Pi_2|Y\Pi_1) + \ldots + I(X;\Pi_\ell|Y\Pi_{\leq \ell-1}) \\ &= I(X;\Pi_1|Y) + \sum_{v\in L_1} \Pr[\pi_r \text{ reaches } v] I(X;\Pi_2|Yv) + \ldots + \sum_{v\in L_{\ell-1}} \Pr[\pi_r \text{ reaches } v] I(X;\Pi_\ell|Yv) \\ &= \sum_{v\in V_{\text{int}}} \Pr[\pi_r \text{ reaches } v] I(X;\Pi_v|Yv), \end{aligned}$$

where Π_v is the message sent at the node v. Repeating the same argument for $I(Y;\Pi|X)$, we conclude that

$$\text{IC}(\pi_r) = \sum_{v\in V_{\text{int}}} \Pr[\pi_r \text{ reaches } v] \left(I(X;\Pi_v|Yv) + I(Y;\Pi_v|Xv) \right),$$

If Bob is the owner of the node v, then $I(X;\Pi_v|Yv) = 0$, and if Alice is the owner, then $I(Y;\Pi_v|Xv) = 0$. Hence we have the following proposition.

Proposition 1. *Let π be a protocol, and π_r be the protocol obtained from π by setting the public randomness to $R = r$. If V_{int} denotes the internal nodes of the protocol tree of π_r, then*

$$\text{IC}(\pi_r) = \sum_{v\in V_{\text{int}}} \Pr[\pi_r \textit{ reaches } v] I(P;\Pi_v|Qv), \tag{1}$$

where $P = X$ and $Q = Y$ if Alice owns v, and $P = Y$ and $Q = X$ otherwise. Similarly

$$\text{IC}_{\text{ext}}(\pi_r) = \sum_{v\in V_{\text{int}}} \Pr[\pi_r \textit{ reaches } v] I(XY;\Pi_v|v), \tag{2}$$

Proposition 1 breaks the information cost to a sum over the terms that correspond to the amount of information that is revealed at node v.

As we saw in Lecture 15, given an input (x, y), every internal node u, defines a probability distribution on the set of its children. This probability distribution is *only known to the owner* of

the node u. If u is owned by Alice, then she knows the probability distribution $p_u^x(\cdot)$, and if v is owned by Bob, then he knows the probability distribution $p_v^y(\cdot)$. Given a leaf t of the protocol tree, the probability that t will be reached by the protocol for a given pair of inputs x, y is given by

$$\Pr_{R_A,R_B}[\Pi_{xy}(t)] = \left(\prod_{\substack{i\in[\ell-1] \\ \text{Alice owns } v_i}} p_{v_i}^x(v_{i+1}) \right) \times \left(\prod_{\substack{i\in[\ell-1] \\ \text{Bob owns } v_i}} p_{v_i}^y(v_{i+1}) \right) = p_A^x(t) p_B^y(t),$$

where $(v_1, v_2, \ldots, v_\ell (= t))$ is the unique path from the root to the leaf t. The reason that Alice and Bob need to communicate is that one of them knows some of the terms in the product and the other one knows the rest. Indeed if they knew the probabilities $\Pr_{R_A,R_B}[\Pi_{xy}(t)]$, then they could use the public randomness to mutually sample a leaf according to this probability distribution, and that would be a simulation of π_r with no communication. The idea behind the compression is that Alice and Bob will try to communicate in order to be able to choose a leaf with the right probability distribution $\Pr_{R_A,R_B}[\Pi_{xy}(t)]$, and they will try to save the amount of the communication using the assumption that the information cost of the original protocol is low.

Consider π_r and an input (x, y). Let u be a node owned by Alice, and thus she knows the probability distribution $p_u^x(\cdot)$. Bob, in general, does not know what this probability distribution is, as he does not know x, but he can have an estimate of this probability distribution. Bob knows y, r_B and the partial transcript u, and considering them, he has a posterior probability distribution for w (and prior to knowing x). More precisely

$$q_u^y(w) = \Pr_{XY,R_A}[a_u(x, R_A, r) = w | u, Y = y] = \mathbb{E}_{x\sim\mu|_{u,Y=y}}[p_u^x(w)]$$

is Bob's best estimate for p_u^x, and similarly if v is an internal node owned by Bob then

$$q_v^x(w) = \Pr_{XY,R_B}[b_v(y, R_B, r) = w | v, X = x] = \mathbb{E}_{y\sim\mu|_{v,X=x}}[p_v^y(w)]$$

is Alice's best estimate for p_v^y. We will think of the probabilities p_u^x and p_v^y as the "correct" probabilities and of q_u^y and q_v^x as the estimate of the other party of these probabilities. Now with this notation we can state Proposition 1 in terms of divergence.

Theorem 2. *Let π be a protocol, and let V_{int} denote the internal nodes of the protocol tree. Then*

$$\text{IC}_\mu(\pi) = \mathbb{E}_r \mathbb{E}_{xy\sim\mu} \left[\sum_{v\in V_{\text{int}}} \Pr[\pi_r \text{ reaches } v] D(p_v \| q_v) \right],$$

where $p_v = p_v^x$ and $q_v = q_v^y$ if v is owned by Alice, and $p_v = p_v^y$ and $q_v = q_v^x$ otherwise.

Proof. The proof is by applying the formula

$$I(A; B|C) = \mathbb{E}_{ac\sim AC}\left[D\left(B|_{A=a,C=c} \parallel B|_{C=c} \right) \right]$$

from Lecture 13 to each term in (1). □

1.1. **Braverman-Rao's correlated sampling.** Theorem 2 shows that if the information cost of a protocol tree is low, then for many nodes on the tree, the divergence of the correct distribution (known to the owner) from the other party's prior of that distribution is small. Let us look at an internal node u of of the tree, and let us assume that it is owned by Alice. Suppose that u has many children, and thus at this point Alice is likely to send a long message. Note that for example in a protocol with bounded number of rounds, most nodes of the tree will have many children. Now Alice is going to pick a child according to the distribution p_u^x, which is known to her. If p_u^x is

close to Bob's estimate q_u^y, then can they use a small amount of communication to mutually pick a child according to the probability distribution p_u^x, rather than first Alice picking the child and then sending the possibly long message to Bob? Note for example if $p_u^x = q_u^y$ and if they both know this fact, then they do not need to communicate at all. Instead they can use public randomness to choose a child randomly according to p_u^x (which is known to both in this case). Note also the difficultly here that even if $p_u^x = q_u^y$, Alice and Bob might not be aware of it. Braverman and Rao [BR14] discovered a protocol whose communication is bounded by a function of the divergence $D(p_u^x \| q_u^x)$ and allows Alice and Bob to do this sampling mutually.

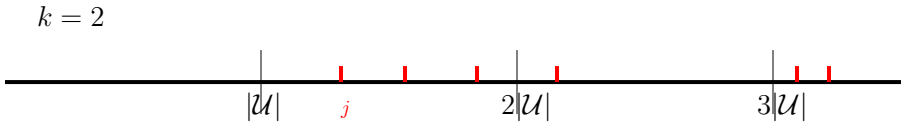
1.1.1. *The correlated sampling protocol.* Braverman-Rao's correlated sampling is very fundamental, and it is interesting even outside the realm of information complexity. Consider two distributions p and q over a universe $\mathcal{U}$. Suppose that p is known to Alice and q is known to Bob. They want to use as little communication as possible to mutually pick a random element from $\mathcal{U}$ according to the distribution p.

Alice and Bob will use the public randomness to generate a random sequence $(a_1, a_2, \ldots)$ where each $a_i = [x_i, \alpha_i]$ is sampled independently by picking $x_i \in \mathcal{U}$ and $\alpha_i \in [0,1]$ both uniformly at random (and independently of each other). Let $a_j = [x_j, \alpha_j]$ be the first element in the sequence with $\alpha_j \le p(x_j)$. Obviously Alice is the only one who knows what a_j is, as Bob does not know p. Note that $\Pr[x_j = x] = p(x)$, which shows that x_j has the right distribution p. So what remains is that Alice has to give enough hints to Bob, so that he can figure out what j is. First note that with high probability $j = O(|\mathcal{U}|)$. Indeed note that $1_{[\alpha_i \le p(x_i)]}$ are independent Bernoulli random variable with expected values $\frac{1}{|\mathcal{U}|}$. Hence

$$\Pr[j > m|\mathcal{U}|] = \left(1 - \frac{1}{|\mathcal{U}|}\right)^{m|\mathcal{U}|} < e^{-m}.$$

So the first thing Alice will do is that she will send the value of $k = \left\lceil \frac{j}{|\mathcal{U}|} \right\rceil$ to Bob so that he will know that j belongs to the window $k|\mathcal{U}| \le j < (k+1)|\mathcal{U}|$ of length $|\mathcal{U}|$. Note that this requires $O(\log(k))$ bits of communication which with high probability is $O(1)$ as we discussed above.

FIGURE 1. The red points correspond to the values of i with $\alpha_i \le p(x_i)$. Here $k = 2$ is the first window that contains such a point, and the first red element in this window is j.



Now Bob knows that j belongs to this possibly large window (denoted by W), but he needs some way to identify it. To this end, Alice also sends him the values of $s = 2 + \log(1/\epsilon)$ (publicly chosen) hash[1] functions $h_1(x_j), \ldots, h_s(x_j)$.

Now Bob will use his distribution q (his guess of what p is) to sieve out the potential a_j's from the window that Alice has told him. So he computes

$$\mathcal{Q}_0 = \{i \in W : \alpha_i \le q(x_i)\},$$

[1]For example, in the case $\mathcal{U} = \{0,1\}^n$, they can publicly choose $r_1, \ldots, r_s \in \{0,1\}^n$ at random, and then Alice can send the values of $h_i(x_j) = \langle x_j, r_i \rangle_{\mathbf{F}_2}$ to Bob.

pretending that his estimate q is prefect and is precisely equal to p. Note that typically $\mathcal{Q}_0$ will have very few elements (in expectation it has only one element). Now if $j \in \mathcal{Q}_0$ then Bob will identify it using the hash values that Alice has sent to him. Indeed since $\mathcal{Q}_0$ is very small it will be very unlikely that some element in $\mathcal{Q}_0$ other than x_j will match these hash functions.

Now what if j does not belong to $\mathcal{Q}_0$? Then with high probability none of the elements in $\mathcal{Q}_0$ will match the hash values and Bob will know that his estimate q was not good enough and as a result he has sieved out j. Thus he will revise his set $\mathcal{Q}_0$ to include more elements (and hopefully j):

$$\mathcal{Q}_1 = \{i \in W : \alpha_i \leq 2q(x_i)\}.$$

But including more elements in this way means possibly doubling the size of $\mathcal{Q}$. Thus he might need more hash values to identify x_j correctly, and because of this he will ask Alice to send him one more hash function. Alice will send him the value of $h_{s+1}(x_j)$, and then Bob will look in $\mathcal{Q}_1$ to see if any element matches all the hash values. If yes, he finds x_j and otherwise it means that he will need to consider even more elements. So they repeat the above process. Let us formally state the protocol now:

- Alice finds the first j with the first $\alpha_j \leq p(x_j)$.
- Alice sends Bob $k = \left\lceil \frac{j}{|\mathcal{U}|} \right\rceil$.
- Alice sends Bob the hash values $h_i(x_j)$ for $i = 1, \ldots, s$, where $s = 2 + \lceil \log(1/\epsilon) \rceil$.
- Repeat until Bob produces an output, beginning with $t = 0$:
  - Bob defines
    $$\mathcal{Q}_t = \{i \in W : \alpha_i \leq 2^t q(x_i)\}.$$
  - If any element in $\mathcal{Q}_t$ matches the hash values $h_1(x_j), \ldots, h_{s+t}(x_j)$ Bob responds "success" and outputs that value.
  - Otherwise he responds "fail", and Alice sends him another hash value $h_{s+t+1}(x_j)$.
  - Set $t = t + 1$

Now let us formally analyze the above protocol π. First note that π is guaranteed to terminate at some time $t \leq T := \log \frac{p(x_j)}{q(x_j)}$ as obviously $j \in \mathcal{Q}_T$. Hence as x_j has distribution p, the expected number of rounds is bounded by

$$\mathbb{E}_{x \sim p} \log \frac{p(x)}{q(x)} = D(p \| q).$$

Now since every round requires two bits of communication, we have

$$\mathrm{CC}^{\mathrm{avg}}(\pi) \leq \mathbb{E}\left(\lceil \log(k) \rceil + 2 + \lceil \log(1/\epsilon) \rceil + 2|T|\right) \leq 5 + \log(1/\epsilon) + 2D(p\|q).$$

Next we need to bound the probability of error, that is if Bob by mistake would take another element x_ℓ for x_j.

Claim 3. *Consider a fixed t. Then*

$$\Pr[\exists \ell \in \mathcal{Q}_t, x_\ell \neq x_j, (h_i(x_\ell) = h_i(x_j)\ \forall i = 1, \ldots, s+t) \mid j] \leq \epsilon.$$

Proof. Note that conditioned on the value of j, the elements $a_1, \ldots, a_{j-1}$ are uniformly and independently distributed in $\{(u, \alpha) : u \in \mathcal{U}, \alpha > p(u)\}$. On other hand this conditioning does not affect the distribution of $(a_{j+1}, a_{j+1}, \ldots)$. Note that in either case $\ell < j$, or $\ell > j$,

$$\Pr[\ell \in \mathcal{Q}_t | j] \leq \Pr[\ell \in \mathcal{Q}_t] = \frac{2^t}{|\mathcal{U}|}.$$

Moreover by the property of the hash functions

$$\Pr[h_i(x_\ell) = h_i(x_j)\ \forall i = 1, \ldots, s+t \mid j, \ell, x_\ell \neq x_j] \leq 2^{-s-t}.$$

Hence

$$\Pr[\ell \in \mathcal{Q}_t, x_\ell \neq x_j, (h_i(x_\ell) = h_i(x_j)\ \forall i = 1, \ldots, s+t) \mid j] \leq \frac{2^{-s}}{|\mathcal{U}|} \leq \frac{\epsilon}{|\mathcal{U}|}.$$

Applying the union bound completes the proof. □

Setting t to be terminating round, shows that the protocol π performs the required task, has error probability at most ϵ, and its average communication complexity is at most $5 + \log(1/\epsilon) + 2D(p\|q)$. Unfortunately the price $2D(p\|q)$ is too much to pay for applications such as information equals amortized communication. The coefficient 2 comes from the fact that at each round Bob has to say "fail" so that Alice will send him another hash value. This can be easily remedied. They can combine many rounds together. In other words at each round Alice would send many new hash values to Bob.

- Alice finds the first j with the first $\alpha_j \leq p(x_j)$.
- Alice sends Bob $k = \left\lceil \frac{j}{|\mathcal{U}|} \right\rceil$.
- Alice sends Bob the hash values $h_i(x_j)$ for $i = 1, \ldots, s$, where $s = 2 + \lceil \log(1/\epsilon) \rceil$.
- Repeat until Bob produces an output, beginning with $t = 0$:
  - Bob defines
    $$\mathcal{Q}_t = \{i \in W : \alpha_i \leq 2^{t^2} q(x_i)\}.$$
  - If any element in $\mathcal{Q}_t$ matches the hash value Bob responds "success" and outputs that value.
  - Otherwise he responds "fail", and Alice sends him another $2t+3$ new hash values.
  - Set $t = t + 1$

This new protocol terminates after fewer rounds as Q_t grows much faster. Indeed it will terminate after at most $T := \left\lceil \sqrt{\log \frac{p(x_j)}{q(x_j)}} \right\rceil$ rounds. The analysis of the error probability is exactly as the previous protocol. The number $2t + 3$ is chosen for the convenience so that after t rounds the total number of hash functions is $s + (t+1)^2 - 1$. Hence using $T \leq \sqrt{\log \frac{p(x_j)}{q(x_j)}} + 1$ the average communication is at most

$$\begin{aligned}
\mathrm{CC}^{\mathrm{avg}}(\pi) &\leq \mathbb{E}\left[\lceil \log(k) \rceil + s - 1 + (T+1)^2 + 2T\right] \\
&\leq 5 + \log(1/\epsilon) + \mathbb{E}_{x\sim p}\left[\log \frac{p(x)}{q(x)} + 5\sqrt{\log \frac{p(x)}{q(x)}}\right] \\
&= D(p\|q) + 5 + \log(1/\epsilon) + 5\mathbb{E}_{x\sim p}\left[\sqrt{\log \frac{p(x)}{q(x)}}\right] \\
&\leq D(p\|q) + 5 + \log(1/\epsilon) + 5\sqrt{\mathbb{E}_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right]} \\
&= D(p\|q) + \log(1/\epsilon) + O(\sqrt{D(p\|q)} + 1).
\end{aligned}$$

This finally finishes the analysis of the correlated sampling, and proves the following theorem.

Theorem 4. *Given probability distributions p and q over the same universe, known to Alice and Bob respectively, there exists a protocol with expected communication at most*

$$D(p\|q) + \log(1/\epsilon) + O(\sqrt{D(p\|q)})$$

such that with probability at least $1-\epsilon$ both parties output the same output x which is distributed exactly according to p.

Next we show that how one can use this sampling to compress a protocol that has bounded number of rounds.

1.2. **Compression using this sampling.** The idea is that Alice and Bob start from the root and use Theorem 4 at every node to sample a child according to the correct distribution. If no error occurs, then they will reach a mutual leaf which will have the same distribution as in the original protocol, and furthermore Theorem 2 shows that the expected number of communicated bits is bounded by a function of the information cost. However there is one complication. What if an error occurs and Alice and Bob lose synchronization and they end up on different nodes, and thus traverse different paths down the tree. The problem here is that in this case we do not have any control on the amount of communication anymore. In this case, Alice and Bob will do correlated sampling with two completely irrelevant distribution as one thinks that they are on one node of the tree and the other thinks they are on some other node. That is why in the following theorem, there is a conditioning on an event E. Here E is basically the event that all the correlation samplings will go right, and Alice and Bob stay synchronized.

Theorem 5. *Consider a k-round protocol π, and $\epsilon > 0$. There is a* public coin *protocol τ and and event $E(x, y, r_\pi, r_\tau)$ such that*

$$\Pr[E|x, y, r_\pi] \geq 1 - \epsilon \quad \forall x, y, r_\pi,$$

where r_π is the public randomness of π and r_τ is the public randomness of τ. Moreover conditioned on E the following two statements hold:

- *$\tau(x, y)$ has the same distribution as $\pi_r(x, y)$ for every x, y, r.*
- *The expected number of communicated bits is at most $I + O(k \log(k/\epsilon) + \sqrt{Ik})$, where I is the information cost of π.*

Proof. Let $\delta = \epsilon/k$. Alice and Bob use Theorem 4 (with error parameter δ) at every node to sample a child according to the correct distribution. At a node v, this requires the expected communication at most

$$D(p_v\|q_v) + \log(1/\delta) + O(\sqrt{D(p_v\|q_v)}).$$

While they mutually sampled a child they move to that child and continue this process until they reach a leaf. Let E be the event that no error occurs in these correlated samplings, and let $v_1, \ldots, v_{k+1}$ be the path from the root to the leaf transversed by them. Since there only k rounds, the probability that an error happens in the correlated samplings is at most $k\delta = \epsilon$. Hence $\Pr[E|x, y, r_\pi] \geq 1-\epsilon$. On the other hand conditioned on E, the expected communication is at most

$$\sum_{i=1}^{k} D(p_{v_i}\|q_{v_i}) + \log(1/\delta) + O(\sqrt{D(p_{v_i}\|q_{v_i})}) \leq k \log(1/\delta) + \sum_{i=1}^{k} D(p_{v_i}\|q_{v_i}) + O\left(\sqrt{k \sum_{i=1}^{k} D(p_{v_i}\|q_{v_i})}\right),$$

where we used the Cauchy-Schwarz inequality. Taking the expected value with respect to $xy \sim \mu$, and the public randomness r_π, we conclude that conditioned on E, the expected number of

communicated bits is at most

$$\mathbb{E}_{xy\sim\mu,r}[\text{communication}|E] \leq k\log(1/\delta)+\mathbb{E}_{xyr_\pi}\left[\sum_{i=1}^{k} D(p_{v_i}\|q_{v_i})\right]+O\left(\sqrt{k\mathbb{E}_{xyr_\pi}\left[\sum_{i=1}^{k} D(p_{v_i}\|q_{v_i})\right]}\right),$$

where we used concavity of $\sqrt{x}$ to take the expected value inside the square root. Now the proof is completed as

$$\mathbb{E}_{xyr}\left[\sum_{i=1}^{k} D(p_{v_i}\|q_{v_i})\right] = \mathbb{E}_{xyr}\left[\sum_{v\in V_{\text{int}}} \Pr[\pi \text{ reaches } v]D(p_v\|q_v)\right] = \text{IC}_\mu(\pi),$$

using Theorem 2. □

Remark 6. Here basically E is the event that things go right in the simulation, and thus conditioned on E, τ will be a perfect simulation of π and furthermore its expected communication will be bounded by $K := I + k\log(k/\epsilon) + O(\sqrt{Ik})$. Note that the gain is that I is outside $O(\cdot)$, and thus we know the exact constant in front of I. On the other hand when E does not happen we have no control on the communication. Obviously we can always truncate the protocol and terminate after K/ϵ bits of communication. Since conditioned on E the expected communication is K, by Markov's inequality, the probability that it will be larger than K/ϵ will be bounded by ϵ. Hence this way we will get a simulation of π with error at most 2ϵ and worst case communication

$$K/\epsilon = O_\epsilon(I + k\log k + \sqrt{kI}),$$

as it was discussed in the previous lecture. ■

References

[BR14] Mark Braverman and Anup Rao, *Information equals amortized communication*, IEEE Trans. Inform. Theory **60** (2014), no. 10, 6058–6069. MR 3265014

School of Computer Science, McGill University, Montréal, Canada
E-mail address: hatami@cs.mcgill.ca