CS 229r Information Theory in Computer Science March 26, 2019

Lecture 15

Instructor: Madhu Sudan *Scribes: Nari Johnson, Duncan Rheingans-Yoo*

1 Overview

Today we will discuss:

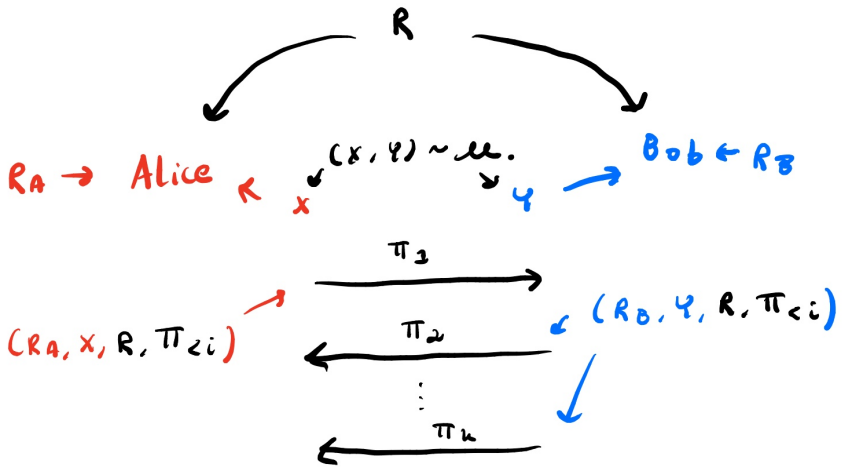
- Compressing Interactive Communication
- Correlated Sampling

2 Review of protocol

We'll begin by discussing single-shot communication, which occurs when some players Alice and Bob only have 1 conversation. In the future, we'll discuss amortized communication and examine what happens when Alice and Bob can have multiple conversations.

2.1 Setup

- Alice has input $X \sim \mu$ unknown to Bob.
- Bob has input $Y \sim \mu$ unknown to Alice.
- Alice has private randomness R_A unknown to Bob.
- Bob has private randomness R_B unknown to Alice.
- Both players observe public randomness R.



2.2 Protocol

The protocol π is the transcript of Alice and Bob's communication. The protocol begins by Alice sending a bit π_1. Bob sends the next bit π_2, and Alice and Bob continue to alternate sending bits until k total bits have been sent.

Each bit $(\pi_i : i \in [k])$ is a random variable. Because Alice and Bob only have access to their own inputs, their own private randomness, public randomness R, and the transcript so far, we can rewrite each π_i as a function of these variables:

$$\pi_i = \begin{cases} \pi_i(X, R, R_A, \pi_{<i}) & \text{i is odd} \\ \pi_i(Y, R, R_B, \pi_{<i}) & \text{i is even} \end{cases}$$

Because there are k π_i bits exchanged, $CC(\pi) = k$ (the communication complexity of the entire protocol is k).

3 Information Complexity

Definition 1 (Internal Information Complexity)**.** The internal information complexity of protocol π, denoted IC_μ^{int}, is given by

$$IC_\mu^{int}(\pi) = I(X; \pi | Y, R) + I(Y; \pi | X, R)$$

$IC_\mu^{int}(\pi)$ is the information the protocol π conveys to Alice and Bob about each others' inputs. In contrast,

Definition 2 (External Information Complexity)**.** The external information complexity of protocol π, denoted IC_μ^{ext}, is given by

$$IC_\mu^{ext}(\pi) = I(XY; \pi | R)$$

$IC_\mu^{ext}(\pi)$ is the information the protocol π conveys to an outside observer (without prior knowledge of X or Y) about X and Y. Because information is symmetric, this is also equal to the information that X and Y convey about transcript π.

We can rewrite $IC_\mu^{int}(\pi)$ as

$$IC_\mu^{int}(\pi) = \sum_{i=1}^{k} I(\pi_i; X | Y, R, \pi_{<i}) + I(\pi_i; Y | X, R, \pi_{<i})$$

Notice: $\forall i$, one of these two terms is always equal to 0. This is because when i is odd, π_i is not a function of Y and when i is even, π_i is not a function of X.

Exercise 3. *Show that $IC_\mu^{int}(\pi) \leq IC_\mu^{ext}(\pi)$. Hint: Use the above sum expansion of $IC_\mu^{int}(\pi)$ and the property that only one term is nonzero for each π_i.*

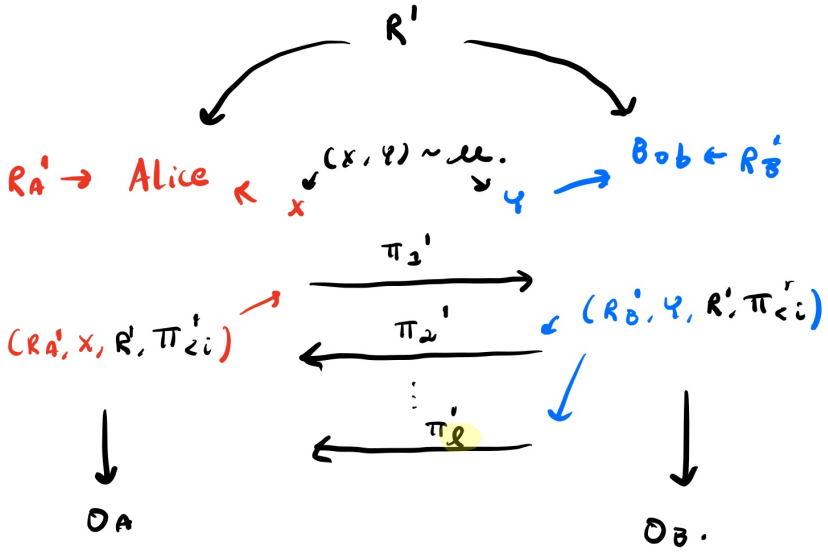
Exercise 4. *Show that $I(X; \pi, R | Y) + I(Y; \pi, R | X) = I(X; \pi | Y, R) + I(Y; \pi | X, R)$.*

3.1 Protocol Simulation

Generally, if a protocol reveals very little information, or if the entropy of a protocol is small, then the protocol itself can be compressed. What does it mean to "compress" a communication? Consider the following modifications to the above interaction:

- We will now consider 1-way communication. Alice will send messages to Bob, but Bob will not send messages to Alice.
- Alice will now send a compressed message to Bob. Intuitively, we will show that Bob can reconstruct Alice's original message (before compression!) from the compressed message.
- The output of our protocol will be this reconstructed transcript.
- Both players have access to the entire transcript, which is synonymous to observing the entire interaction.

More generally, when Alice and Bob can communicate (i.e. when we no longer have the constraint of 1-way communication), we can define our protocol as follows:



Notice:

- Alice and Bob can make use of public randomness R' and private randomness R'_A, R'_B which do not need to be the same as R, R_A, R_B.
- Our compressed protocol π' only requires $l < k$ bits.
- O_A, Alice's output transcript, and O_B, Bob's output transcript, are random variables. They are both functions of (X, Y, R, π').

Definition 5 (Protocol Simulation)**.** π' simulates π if $O_A = O_B = (R, \pi)$ and the distributions of x, y, π, R are exactly preserved.

4 BBCR

BBCR is a compression theorem published by Barak, Braverman, Chen, and Rao in 2007.

Theorem 6 (BBCR)**.** $\forall \pi$ *with* $CC(\pi) = k$*,* $IC(\pi) = I$*, then* $\exists\, \pi'$ *simulating* π *s.t.* $CC(\pi') = O(\sqrt{I \cdot k}\log(k))$*.*

BBCR says that by using simulation, we can achieve a communication complexity that's roughly the geometric mean of the information and communication cost. BBCR is the best known compression algorithm for this problem.

Theorem 7 (Braverman)**.** *Using the same assumptions as BBCR,*
$\forall \pi$ *with* $CC(\pi) = k$*,* $IC(\pi) = I$*, then* $\exists\, \pi'$ *simulating* π *s.t.* $CC(\pi') = 2^{O(I)}$*.*

Therefore you can use Braverman's to find an expression for the communication complexity of π' depending only on the information complexity of π.

Theorem 8 (Ganor, Kol, Raz)**.** *The above are tight.*

One motivation of BBCR is understanding what happens when we engage in multiple conversations simultaneously. In this scenario,

- We have many inputs $(X_1, ..., X_n)$ and $(Y_1, ..., X_n) \sim \mu$.
- We carry on n conversations, each with input (X_i, Y_i), in parallel.

How can we compress the protocols π of these n parallel conversations? We will formalize this more next time, but we can use BBCR to prove a lower bound of the communication complexity:

Theorem 9. $CC(\pi^{\otimes n}) \geq CC(\pi)\sqrt{n}$

(The above notation $\pi^{\otimes n}$ is expressed in terms of the direct product, which we will define in a future lecture.)

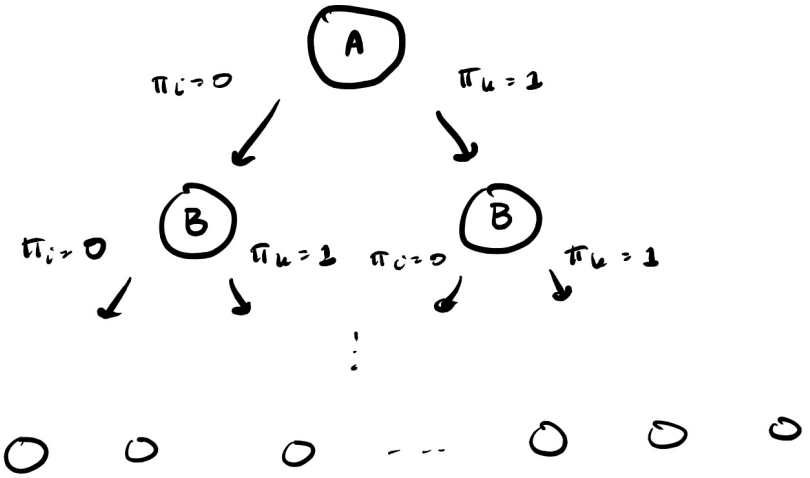
Theorem 10. $CC(\pi^{\otimes n}) = nIC(\pi)(1 + O(1))$

Theorem 10 can be rewritten as $CC(\pi^{\otimes n})/n \approx IC(\pi)$, or that the amortized communication complexity (i.e. communication complexity "per copy" of the problem, as number of copies n approaches infinity) is equal to the information cost.

Today we will begin to prove BBCR as stated in Theorem 6. In order to do this, we'll introduce the notion of a protocol tree.

5 Protocols, Priors, Information Cost

Assume that π has no common randomness. We'll define a protocol tree as follows:



Here, at each node, the path taken (whether or not we traverse right or left down the tree) represents the bit of transcript that the player whose turn it is sends. At each node of the tree, we need to keep track of who "owns" that node - whether it's Alice or Bob's turn to communicate. Because there are k rounds of interaction, the tree is of depth k. At the final level of the tree, note that each individual leaf represents a distinct "path" taken through the tree. Therefore each leaf represents a distinct transcript of Alice and Bob's communication.

At any level of the tree, what determines if we should go right or left? The public randomness R, the private randomness R_A or R_B, and the input of the player who owns that node. Therefore we're interested in $(\pi_i|\pi_{<i}, X)$ and $(\pi_i|\pi_{<i}, Y)$. For each node U at level i in the tree, define:

- $(P_U^A = \pi_i|\pi_{<i}, X)$, where P_U^A represents Alice's belief (distribution) about what direction the path will continue in.
- $(P_U^B = \pi_i|\pi_{<i}, Y)$, where P_U^B represents Bob's belief (distribution) about what direction the path will continue in.

Notice that at each step of communication, it's not the public or private randomness, but instead private inputs (X, Y) that bind Alice and Bob to go one way or another. Therefore these two distributions P_U^A and P_U^B will not be identical. The divergence between P_U^A and P_U^B is the information cost I.

First, let's consider the setting where the information cost is very low: when $I = 0$. This means that at any level of the tree, the players will learn nothing about the other player's input based on the path that was taken. In order for this to occur, the distributions P_U^A and P_U^B must be identical at every node.

Claim 11. *$I = 0$ only if $\forall U, P_U^A = P_U^B$.*

This means that at every node, the distribution of going left or going right, regardless of whether it's conditioned on Alice or Bob's prior knowledge, is identical. Now we can simulate the entire path throughout the protocol tree using 0 communication. We can do so by using shared randomness R. First sample a real number R between 0 and 1. At each given node U, Alice and Bob's beliefs P_U^A and P_U^B are Bernoulli distributions with parameter p of going right at that node. If R is less than p, then go right, and if R is greater than p, then go left. Then restrict attention to the portion of the 0-1 line that R is in (on either side of p) and relabel the ends to be 0 and 1. Then evaluate the next node. Therefore our π' to simulate π will have common randomness.

Our π' when $I = 0$ is the "easy" case. Now we will examine the communication complexity when I is very small. What is our distribution on the leaves?

6 Correlated Sampling

We will define a third distribution on each node of the tree, P_U. P_U will model what we will call "the right distribution" on the probability of going right or left at that node.

Definition 12. $P_U = (\pi_i | \pi_{<i}, X, Y)$.

For nodes owned for Alice, $P_U = P_U^A$, and for nodes owned by Bob, $P_U = P_U^B$.

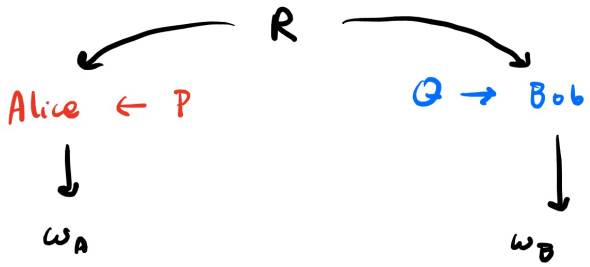
Suppose some player has knowledge of both inputs (X, Y) and wants to know the probability of going right at some node U. Then the player can simulate either Alice or Bob using P_U.

Consider the situation where P_U^A and P_U^B are very close to each other $\forall U$ but not identical. In this case, we can't just use common randomness R as we did when $I = 0$.

Goal: Sample the root to leaf path of the protocol tree according to the $\{P_U\}_U$, or according to the distribution P_U at each node U.

One solution to this problem comes from correlated sampling. In this problem,

- Alice gets as input a distribution P.
- Bob gets as input a distribution Q.
- Alice and Bob have common randomness R.
- Alice and Bob engage in 0 communication.
- Alice produces output $\omega_A \sim P$.
- Bob produces output $\omega_B \sim Q$.



Example 13. *P, Q are distributions on the leaves of the tree. Then when P and Q are close to each other (formalization coming), whp $\omega_A = \omega_B$.*

Objective: Maximize $Pr[\omega_A = \omega_B]$. Or: Minimize $Pr[\omega_A \neq \omega_B]$.

Suppose P and Q have disjoint support. Then $Pr[\omega_A \neq \omega_B] = 1$. On the other hand, as distributions P and Q become closer, then this probability increases. Consider the case when $P = Q$. Then $Pr[\omega_A \neq \omega_B] = 0$. We want something that interpolates nicely between these two extremes.

Exercise 14. *Use our discussion of the $I = D(P||Q) = 0$ case above to solve this problem with common randomness when P and Q have the same distribution.*

Total variation distance $\delta(P, Q)$ parametrizes how far apart P and Q are in terms of where their support is. The following exercise shows that TVD is a lower bound on how often the outputs can agree:

Exercise 15. *Show that $\forall P, Q$,*

$$\min Pr[\omega_A \neq \omega_B] \geq \delta(P, Q)$$

where total variation distance $\delta(P, Q)$ is defined as

$$\delta(P, Q) = \frac{1}{2} \sum_{\omega \in \Omega} |P(\omega) - Q(\omega)|$$

Lemma 16 (Broder, Kleinberg, Tardos, Holenstein)**.** $\exists$ *a protocol which achieves*

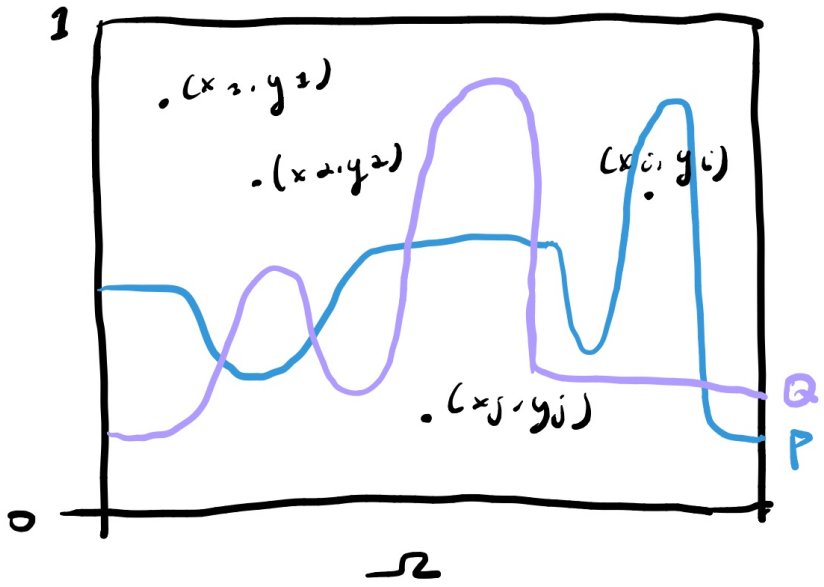
$$Pr[\omega_A \neq \omega_B] \leq \frac{2\delta(P, Q)}{1 + \delta(P, Q)} \leq 2\delta(P, Q)$$

Note: When $\delta(P, Q) = 1$, we can upper-bound the probability exactly using $\delta(P, Q)$. When $\delta(P, Q)$ is small, then our bound is much closer to $2\delta(P, Q)$.

Note: Broder used a protocol of this form to compare files. Correlated sampling has many applications to search engine hash functions. His solution to this problem is called the min-hash protocol.

6.1 Holenstein's Protocol

How can we assess the similarity between two distributions P and Q?



The x-axis of the above chart represents all of the numbers in alphabet Ω, and the y-axis represents all of the real numbers between 0 and 1. The above lines represent the density functions of distributions P and Q. Bob wants to produce a sample according to the purple distribution (Q), and Alice wants to produce a sample according to the blue distribution (P). We also want the samples to equal each other whp. Here, a "sample" represents an x-coordinate, or a letter drawn from Ω.

We can use the following random protocol:

- Using common randomness R, begin sampling randomly generated sequence of points $\{(x_1, y_1), (x_2, y_2), ...\}$ in the rectangular region.
- Alice and Bob will each stop when they observe the first point (x_i, y_i) that's under their respective distribution curves on the chart. Therefore we maintain that Alice and Bob output ω_A, ω_B according to the distributions P and Q.
- After each point (x_i, y_i) is sampled, check to see if (x_i, y_i) is (1) under the purple curve, (2) under the blue curve, or (3) under both the blue and purple curves. If (x_i, y_i) is only under the blue curve, then output $\omega_A = x_i$, but continue sampling until we get a point under the blue curve to output Bob's (x_j, y_j). Similarly, if (x_i, y_i) is under the purple curve, but not under the blue curve, then Bob will output $\omega_B = x_i$ but Alice must keep sampling until she outputs a point under her curve. However, if our (x_i, y_i) is under both curves, then Bob and Alice will both output $\omega_A = \omega_B = x_i$.

Exercise 17. *Show that Holenstein's Protocol achieves*

$$Pr[\omega_A \neq \omega_B] \leq \frac{2\delta(P,Q)}{1+\delta(P,Q)}$$

6.2 Back to our goal...

Recall that our goal is to sample the root-to-leaf path of the protocol tree according to $\{P_U\}_U$. If the distributions P_U^A and P_U^B are very close to each other $\forall U$, then there is a relatively high probability that they will output the same leaf. If we can apply the correlated sampling solution to the distributions P^A and P^B only on the leaf nodes, then we can simulate the path.

A simple example: Suppose that our protocol only involves Alice communicating, and Bob listening. Define

$$P_U^B \sim Bern(\frac{1}{2})$$

$$P_U^A \sim Bern(\frac{1}{2} - \delta)$$

where P_U^B is Bob's prior probability of going right at node U, P_U^A is Alice's prior probability of going right at node U, for small δ and for all nodes U.

Can we upper-bound the variation distance between leaves? To construct the distribution over the leaves, consider that we begin at the root node and traverse down the tree. At any level of the tree, if Alice and Bob disagree, then our distributions will produce different leaves. At any node, the probability that Alice and Bob will disagree is $O(\delta)$. There are k levels, so there are k chances for Alice and Bob to disagree. Therefore

$$TVD(leaf^A, leaf^B) = O(k\delta)$$

where TVD is the total variation distance, $leaf^A$ is Alice's distribution over the leaves, and $leaf^B$ is Bob's distribution over the leaves. Therefore the probability that Alice and Bob don't get the same leaf is

$O(k\delta)$.

Assumption: $k\delta$ is tiny.

Now how do we relate this problem to information cost? We can define V_i as

$$V_i = I(\pi_i; X|Y, \pi_{<i}) + I(\pi_i; Y|X, \pi_{<i})$$

In our case, the divergence between these two distributions is δ^2, so we can simplify:

$$V_i = \delta^2$$

This implies that $I = k\delta^2$, as $I = \sum_{i=1}^{k} V_i$, where k is the communication complexity of the original protocol.

We can rewrite $O(k\delta) = O(\sqrt{k}\sqrt{k\delta^2})$. This is equal to $O(\sqrt{I}\sqrt{k})$. While this looks similar to the communication complexity bound that we're trying to prove for BBCR, we aren't quite there yet. We've just shown an upper bound for the error in our zero-communication protocol as defined in the Correlated Sampling section (below Definition 12), but this is different from the BBCR protocol, which we want to have some exponentially small probability of error ε. In the next lecture we will use this bound to construct the correct BBCR protocol.

7 Solutions

Solution to Exercise 14: Let R have the same distribution as P, Q. Then Alice produces output $\omega_A = R$ and Bob produces output $\omega_B = R$. Because $\omega_A \sim R, \omega_B \sim R$ and $R \sim P \sim Q$, we have that $\omega_A \sim P, \omega_B \sim Q$. We also have that $\omega_A = \omega_B$ always.

Solution to Exercise 15: (inspired by MIT 6.896[1])

By definition, ω_A is drawn randomly from distribution P, and ω_B is drawn randomly from distribution Q. Consider first that

$$\begin{aligned} P(z) &= Pr[\omega_A = z] \\ &= Pr[\omega_A = z \cap \omega_B = \omega_A] + Pr[\omega_A = z \cap \omega_B \neq \omega_A] \\ &\leq Pr[\omega_B = z] + Pr[\omega_A = z \cap \omega_B \neq z] \\ &\leq Q(z) + Pr[\omega_A = z \cap \omega_B \neq \omega_A] \end{aligned}$$

Therefore it follows that

$$P(z) - Q(z) \leq Pr[\omega_A = z \cap \omega_B \neq \omega_A]$$

Similarly, it also follows that

$$Q(z) - P(z) \leq Pr[\omega_B = z \cap \omega_A \neq \omega_B]$$

[1] http://people.csail.mit.edu/costis/6896sp11/lec3s.pdf

We're given that

$$\begin{aligned}
\delta(P,Q) =& \frac{1}{2}\sum_{\omega\in\Omega}|P(\omega)-Q(\omega)| \\
2\delta(P,Q) =& \sum_{z\in\Omega,P(z)\geq Q(z)}(P(z)-Q(z)) + \sum_{z\in\Omega,P(z)<Q(z)}(Q(z)-P(z)) \\
\leq& \sum_{z\in\Omega,P(z)\geq Q(z)} Pr[\omega_A = z \cap \omega_B \neq \omega_A] + \sum_{z\in\Omega,P(z)<Q(z)} Pr[\omega_B = z \cap \omega_A \neq \omega_B]
\end{aligned}$$

$$\begin{aligned}
&\leq Pr[\omega_A \neq \omega_B] + Pr[\omega_B \neq \omega_A] \\
2\delta(P,Q) &\leq 2Pr[\omega_A \neq \omega_B] \\
\min Pr[\omega_A \neq \omega_B] &\geq \delta(P,Q)
\end{aligned}$$

Solution to Exercise 17:

By construction, Alice and Bob choose the same tuple, (i.e $(x_i, y_i) = (x_j, y_j)$) iff. the first sampled point under one of the curves is actually under both curves. So because we are sampling uniformly, $Pr[(x_i, y_i) \neq (x_j, y_j)]$ is just the area under exactly one curve divided by the area under either curve. Recall the definition of total variation distance:

$$\delta(P,Q) = \frac{1}{2}\sum_{\omega\in\Omega}|P(\omega)-Q(\omega)|$$

By this definition, $\delta(P,Q)$ is the area under P but not Q (and equivalently the area under Q but not P). Also, the total area under P must be equal to 1. Therefore the total area under either of the curves is $1+\delta(P,Q)$ and the area under exactly one curve is $2\delta(P,Q)$. Therefore,

$$Pr[(x_i, y_i) \neq (x_j, y_j)] = \frac{2\delta(P,Q)}{1+\delta(P,Q)}$$

Because Alice outputs $\omega_A = x_i$ and Bob outputs $\omega_B = x_j$, we know that $\omega_A \neq \omega_B \implies (x_i, y_i) \neq (x_j, y_j)$ (the converse is not necessarily true). So

$$Pr[\omega_A \neq \omega_B] \leq Pr[(x_i, y_i) \neq (x_j, y_j)] = \frac{2\delta(P,Q)}{1+\delta(P,Q)}$$

8 Ideas for Exercises

Exercise 18. *Find an example of protocol π' simulating some π where the length of the simulated transcript π' is less than the entropy of the transcript $H(\pi)$. Morally, why is this possible? Hint: Consider the case when $I = D(P||Q) = 0$.*

Solution: As established earlier, when $I = 0$ the length of π' can be 0 (no communication), so any non-trivial π will give us this result. Morally, the shared randomness R' is doing all our "communication" for us, so we're not really compressing the transcript to sub-entropy levels.

Exercise 19. *Consider Lemma 16 for the special case when P, Q are distributions over $\{0,1\}$. Come up with a protocol that is as simple as possible, in that it only requires sampling one real number from the public randomness R.*

Solution: Suppose $P \sim Bern(p), Q \sim Bern(q)$. Sample $R \sim Unif([0,1])$. Have Alice output 1 if $R < p$ and 0 otherwise, and Bob output 1 if $R < q$ and 0 otherwise. Certainly, Alice's output $\omega_A \sim P$ and Bob's output $\omega_B \sim Q$. Also, $Pr[\omega_A \neq \omega_B] = |p - q| = \delta(P, Q)$. Because $p, q \in [0,1]$, $|p - q| \leq 1$, so $\delta(P,Q) \leq \delta(P,Q)\frac{2}{1+\delta(P,Q)}$, so $Pr[\omega_A \neq \omega_B] \leq \frac{2\delta(P,Q)}{1+\delta(P,Q)}$.