COMP760, SUMMARY OF LECTURE 18.

HAMED HATAMI

1. Divergence and the total variation distance

The analysis of Barak, Braverman, Chen, and Rao's compression protocol [BBCR10] uses an inequality that relates the total variation distance to the divergence. The total variation distance is probably the most natural notion of a distance between two probability distributions p and q on a universe $\mathcal{U}$. It is defined as

$$|p-q| = \max_E |p(E)-q(E)|.$$

Note that the maximum[1] in the above formula is achieved for $E = \{x \in \mathcal{U} \ : \ p(x) \le q(x)\}$. This shows that the total variation distance is equivalent to the L_1 distance

$$2|p-q| = \|p-q\|_1 = \sum_{x \in \mathcal{U}} |p(x)-q(x)|.$$

Divergence is another notion of distance between probability measures. The following inequality shows that small divergence implies small total variation distance.

Theorem 1 (Pinsker's inequality)**.** *Let p and q be two probability distributions defined on a universe $\mathcal{U}$. Then*

$$\|p-q\| \le \sqrt{D(p\|q)/2}.$$

Note that the opposite direction is not true. One can obviously have two measures which are very close in total variation distance however with $D(p\|q)=\infty$. Indeed it suffices to have one point in the support of p (with however small probability mass) that does not belong to the support of q.

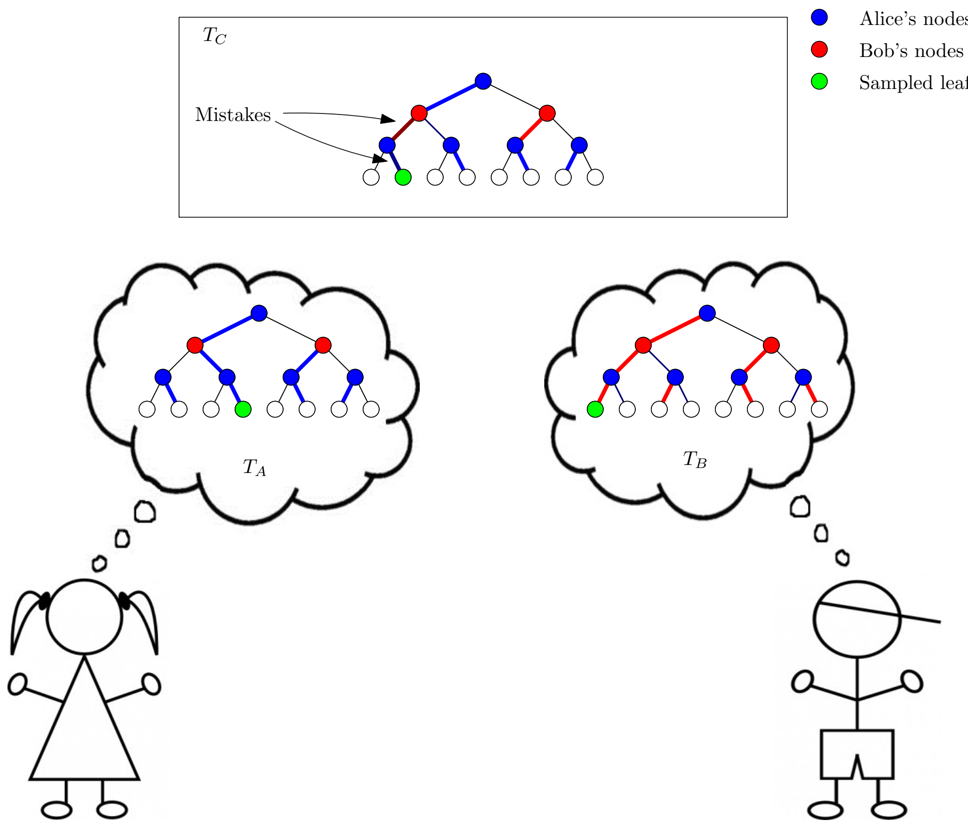
2. Path fixing compression [BBCR10]

In this section we study Barak, Braverman, Chen, and Rao's compression protocol [BBCR10]. The idea behind this protocol is that first Alice and Bob will try to use public randomness to mutually sample a child of each internal node of the tree according to the correct distribution at that node. Since the correct distribution is known only to the owner of each node, the other party can only use his estimate of this distribution and thus might choose a different child by mistake. Then in the second phase, they will communicate to find the differences so that they can both agree on the same path from the root to a leaf. Since in this compression we only care about the order of the magnitude of the compression, we will assume that the protocol is a binary tree, and thus at every round the owner of a node will send one bit.

[1] Here p and q are discrete probability measures.

2.1. **Phase I, Correlated sampling:** For every internal node $w \in V_{\text{int}}$, Alice and Bob sample publicly a number $\rho_w \in [0, 1]$ uniformly at random. Now for every node w, Alice includes the left child in her tree if her best estimate for the probability of going left is at least ρ_w, and she choses the right child otherwise. Here Alice's best estimate is determined by p_w^x if she is the owner of w, and q_w^x otherwise. Bob does the same thing. Now let T_A be the chosen edges by Alice, and T_B be the chosen edges by Bob, and let T_C be the hybrid of T_A and T_B by choosing the child of each node w according to the owner of that node. Here we think of T_C as the correct one. Indeed the leaf t that is reached by following the path from the root to a leaf in T_C has the correct distribution. However the players do not know T_C. Alice only knows T_A and Bob only knows T_B. See Figure 1.

FIGURE 1. The selected edges T_A, T_B and T_C.



2.2. **Phase II, correcting the path:** Each player finds his/her unique path from the root to a leaf, and then they communicate $O(\log(C/\beta))$ bits to find the first vertex on which they disagree (they succeed with probability at least $1 - \beta$). They correct this coordinate and then they repeat this phase until they find no disagreements.

- For every internal node w, Alice and Bob publicly choose $\rho_w \in [0, 1]$ uniformly and independently.

- Alice includes the left child in her tree if her best estimate for the probability of going left is at least ρ_w, and she choses the right child otherwise. Here Alice's best estimate is p_w^x if she is the owner of w, and it is q_w^x otherwise. Bob does the same thing.
- Repeat until they agree on a path from the root to a leaf:
  - Each player computes her/his unique path from the root to a leaf.
  - Use $O(\log C/\beta)$ bits of communication to find the first vertex on which they disagree.
  - They correct this vertex by both agreeing on the child chosen by the owner of tree.
- Output the leaf.

2.3. **Analysis.** Let v_i be the i-th vertex reached on the root-leaf path in T_C (this is the correct path that they want to find). Then

$$\begin{aligned}
\mathbb{E}[\#\text{ mistakes}] &= \sum_{i=1}^{C} \Pr_{XY,R}[\text{mistake in the } i\text{-th step}] \\
&= \sum_{i=1}^{C} \mathbb{E}_{XY,R}\left[|p_{v_i} - q_{v_i}|\right] \leq \sum_{i=1}^{C} \mathbb{E}_{XY,R}\sqrt{D(p_{v_i}\|q_{v_i})} \qquad \text{(Pinsker)} \\
&\leq \sum_{i=1}^{C} \sqrt{\mathbb{E}_{XY,R} D(p_{v_i}\|q_{v_i})} \qquad \text{(concavity of } \sqrt{\cdot}) \\
&\leq \sqrt{C}\sqrt{\sum_{i=1}^{C} \mathbb{E}_{XY,R} D(p_{v_i}\|q_{v_i})} = \sqrt{CI}. \qquad \text{(Cauchy-Shwarz)}
\end{aligned}$$

In order to be able to analyze the error probability of the above protocol, we truncate it after $\sqrt{CI}/\gamma$ rounds where $\gamma = \epsilon/2$. Also we let $\beta = \gamma^2/C$. Then the communication complexity of the truncated version is

$$\mathrm{CC} = O\left(\frac{\sqrt{CI}}{\gamma}\log\frac{C^2}{\gamma^2}\right) = O(\sqrt{CI}\log(C/\epsilon)/\epsilon).$$

It remains to analyze the probability of error. An error occurs if either the number of mistakes is larger than $\sqrt{CI}/\gamma$, or if the $O(\log C/\beta)$-bit protocol for finding the disagreement makes an error. The probability of the former is at most γ by Markov's inequality, and the probability of the latter is at most $\beta\sqrt{CI}/\gamma$ as there are at most $\sqrt{CI}/\gamma$ rounds. Hence the probability of error is bounded by

$$\gamma + \beta\sqrt{CI}/\gamma \leq \gamma + \gamma = \epsilon.$$

References

[BBCR10] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao, *How to compress interactive communication [extended abstract]*, STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing, ACM, New York, 2010, pp. 67–76. MR 2743255

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