CS 15-859: Algorithms for Big Data Fall 2019

Lecture Lecture 9b — October 31

Prof. David Woodruff *Scribe: Justin Jia*

1 Distribution Distances

For the following distance measures we assume that the two distributions being compared share the same support.

1.1 Total Variation Distance

Definition. Total variation distance can be defined as

$$D_{TV}(p,q) = \frac{1}{2}\sum_i |p_i - q_i|$$

in other words half the L1 norm of the difference between the two distributions. An equivalent definition is

$$D_{TV}(p,q) = \max_{events\ E} |p(E) - q(E)|$$

We can see this by separately analyzing $a = \max_{events\ E} p(E) - q(E)$ and $b = \max_{events\ E'} q(E) - p(E)$. In the first scenario, we choose the event $E = \{j \mid p_j > q_j\}$ to be the set of outcomes that are more likely to occur under p and similarly in the second scenario we choose the event $E' = \{j \mid q_j > p_j\}$ to be the set of outcomes that are more likely to occur under q. It should be clear that these events yield the corresponding maximal expressions. Then $a = \sum_{p_i > q_i} p_i - q_i$ and $b = \sum_{p_i < q_i} q_i - p_i$, but these values should be equal because if we add $\sum_{p_i < q_i} p_i + \sum_{p_i > q_i} q_i$ to both values they become one. Finally, $a + b = |p - q|_1$ meaning this other definition of total variation distance is the same as $\frac{1}{2}\sum_i |p_i - q_i|$. This equivalent expression for total variation distance lets us interpret the measure as the largest possible difference the two distributions p and q exhibit on the same event E.

1.2 Hellinger Distance

Definition. If p and q are distributions that we want compare, let $\sqrt{p}$ and $\sqrt{q}$ denote the unit vectors $(\sqrt{p_1}, \ldots, \sqrt{p_n})$ and $(\sqrt{q_1}, \ldots, \sqrt{q_n})$ respectively. Whereas total variation distance resembles the L1 norm, the Hellinger distance corresponds to the L2 norm, being defined as follows.

$$h(p,q) = \frac{1}{\sqrt{2}}|\sqrt{p} - \sqrt{q}|_2$$

We can rewrite the squared Hellinger distance as

$$\begin{aligned} h^2(p,q) &= \frac{1}{2}|\sqrt{p} - \sqrt{q}|_2^2 \\ &= \frac{1}{2}(1 - 2\langle\sqrt{p}, \sqrt{q}\rangle + 1) \\ &= 1 - \langle\sqrt{p}, \sqrt{q}\rangle \end{aligned}$$

which we can use to demonstrate the product property the Hellinger distance has on independent distributions.

Lemma 1. *Let distributions p and q be independent and also let distributions p' and q' be independent. Then product property gives*

$$\begin{aligned} h^2((p,q),(p',q')) &= 1 - \langle\sqrt{p,q}, \sqrt{p',q'}\rangle \\ &= 1 - \sum_{i,j} \sqrt{p_i}\sqrt{q_i}\sqrt{p'_i}\sqrt{q'_i} \\ &= 1 - (\sum_i \sqrt{p_i}\sqrt{p'_i})(\sum_i \sqrt{q_i}\sqrt{q'_i}) \\ &= 1 - \langle\sqrt{p}, \sqrt{p'}\rangle\langle\sqrt{q}, \sqrt{q'}\rangle \\ &= 1 - (1 - h^2(p,p'))(1 - h^2(q,q')) \end{aligned}$$

Note that $\sqrt{p,q}$ and $\sqrt{p',q'}$ have a support size of n^2 if p and q have a support size of n.

Both the Hellinger distance and the total variation distance satisfy the triangle inequality, which we can see because the L1 and L2 norms already exhibit the triangle inequality.

1.3 Jensen-Shannon Distance

Definition. The Kullback-Leibler divergence for two distributions p and q is

$$KL(p,q) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right)$$

The issue here is that the Kullback-Leibler divergence can be infinite when there exists i such that p_i is nonzero but $q_i = 0$. Another distance measure addresses this issue.

Definition. The Jensen-Shannon distance between two distributions p and q, where $r = (p+q)/2$ is the average of these distributions, is

$$JS(p,q) = \frac{1}{2}KL(p,r) + \frac{1}{2}KL(q,r)$$

By comparing both distributions to their average, we avoid the scenario of an infinite value since the denominator in the log is always nonzero whenever the numerator is nonzero. The Jensen-Shannon distance also lower bounds information.

Lemma 2. *Let X, B be possibly dependent random variables and B be a uniform bit. Then*

$$I(X;B) \geq JS(X \mid B = 1, X \mid B = 0)$$

Intuitively, this bound tells us that the information gained about X by B is at least the difference between the distributions of X conditioned on B.

1.4 Relationships Between Distance Measures

Lemma 3. *Squared Hellinger distance lower bounds Jensen-Shannon distance*

$$JS(p,q) \geq h^2(p,q)$$

Lemma 4. *Squared total variation distance lower bounds squared Hellinger distance*

$$h^2(p,q) \geq D_{TV}(p,q)$$

Lemma 5. *If one can distinguish from a sample between distributions p and q with probability $1/2 + \delta/2$, where $\delta/2$ is called the advantage, then δ lower bounds the total variation distance*

$$D_{TV}(p,q) \geq \delta$$

The proposition above actually goes both ways if δ is a constant. For small, non-constant δ then this if and only if does not necessarily hold. The Hellinger distance though captures this for non-constant advantage.

2 Communication Lower Bounds

2.1 The Index Problem

The Index problem involves two players, Alice and Bob. Alice receives a bit vector $x \in \{0,1\}^n$ whereas Bob receives an index $j \in [n]$. Alice is allowed to send a message M to Bob, but Bob is not allowed to send anything to Alice. The goal is to have Bob predict x_j with probability at least 2/3.

Theorem 1. *There exists a distribution over the inputs x and j such that M must necessarily be $\Omega(n)$ bits long. In other words, we want to lower bound the worst case input for Alice and Bob by a linear dependence on n.*

Proof. Let $P_e \geq 2/3$ be the probability that Bob's guess for x_j is correct. We will consider a uniform distribution over X. By Fano's inequality, we have that

$$\begin{aligned} H(X_j \mid M) &\leq H(P_e) + (1 - P_e)(\log_2 |X_j| - 1) \\ &\leq H\Big(\frac{2}{3}\Big) + \frac{1}{3}(\log_2 2 - 1) \\ &= H\Big(\frac{1}{3}\Big) \end{aligned}$$

Note that since $P_e \geq 2/3$, then $H(P_e) \leq H(2/3)$ since entropy for a binary random variable increases as the distribution approaches uniformity. Now consider the mutual information between X and M

$$\begin{aligned} I(X; M) &= \sum_i I(X_i; M \mid X_{<i}) \\ &= \sum_i H(X_i \mid X_{<i}) - H(X_i \mid M, X_{<i}) \\ &\geq \sum_i H(X_i) - H(X_i \mid M) \\ &= \sum_i 1 - H\Big(\frac{1}{3}\Big) \\ &= n(1 - H\Big(\frac{1}{3}\Big)) \end{aligned}$$

Where we used the fact that $H(X_i \mid X_{<i}) = H(X_i)$ since each bit is independent and also the fact that conditioning does not increase entropy to get that $H(X_i \mid M, X_{<i}) \leq H(X_i \mid M)$. Finally, let M have b bits. We know that $H(M)$ is at most $\log_2 |M|$ and the support size here is 2^b. Thus,

$$|M| \geq H(M) \geq I(M; X) = I(X; M) = \Omega(n)$$

■

2.2 Communication Reduction Example

Communication bounds in general can be used to demonstrate lower bounds on the memory for streaming algorithms through reductions in the following fashion.

1. Alice generates her own stream $s(a)$ and runs the streaming algorithm on it, then transmits the state of the algorithm to Bob.
2. Bob uses that memory to compute the algorithm on the concatenation of Alice's stream $s(a)$ with his own stream $s(b)$.
3. If Bob is able to compute the appropriate function $g(a, b)$ for the communication problem, then the space complexity of the streaming algorithm is lower bounded by the 1-way communication complexity.

Example 1. Consider the streaming problem where, given $a_1, a_2 \ldots a_m \in [n]$, the goal is to output the number of distinct numbers. We can construct a reduction with the following steps.

1. Alice's stream $s(a)$ consists of all indices i such that $x_i = 1$. She sends the state of this algorithm to Bob.
2. Bob's stream consists of just his index j.
3. If he notices that the number of distinct elements, which is described by the output of the streaming algorithm, increases, then he knows that $x_j = 0$ since Alice never streamed j. On the other hand, if the number of distinct indices does not increase, then we know $x_j = 1$ since Alice streamed j.

As a result, the memory of the streaming algorithm must be $\Omega(n)$ since the Index problem itself is lower bounded by $\Omega(n)$.

2.3 Augmented Indexing

As we've seen, communication problems seem useful to proving lower bounds for the space complexity of some algorithms. Thus, we will take a look at a more complex version of the Index problem.

Definition. Alice is again given some bit vector $x \in \{0,1\}^n$. However, Bob, in addition to some index $i \in [n]$, is also given $x_1, x_2, \ldots, x_i$, essentially all the bits that precede x_i. Bob's goal again is to predict x_i while Alice is allowed to send a message M to him. This problem setting is called Augmented Indexing.

Because the index Bob is concerned with can again come from anywhere, it seems as if M will have to be just as long as before. We can do a similar proof to the Index problem to see that this is true.

Proof. The proof is almost the same as before, except instead of using the fact that conditioning does not increase entropy to upper bound $H(X_i \mid M, X_{<i})$ by $H(X_i \mid M)$, we can instead directly upper bound it by $H(\delta)$, where δ is Bob's error rate, using Fano's inequality since $M, X_{<i}$ is exactly the information Bob receives. Thus, the Augmented Indexing problem also has a $\Omega(n)$ lower bound. ■

2.4 Lower Bounds for Estimating Norms

Theorem 2. *Any stream algorithm that estimates norms must have space complexity* $\Omega(\log n)$.

Proof. We now describe a protocol for Alice and Bob to accomplish the Augmented Indexing task given an algorithm $\mathcal{A}$ that can estimate norms. Note that the bits of x that are known to Bob are actually flipped here for convenience, so he is aware of x_j for all $j > i$. It should be clear that symmetrically this is the same as the setting described previously.

1. Alice has a bit vector $x \in \{0,1\}^{\log n}$ and creates a vector with a single coordinate equivalent to $\sum_j 10^j x_j$. She then sends Bob the state of the norm estimating algorithm after using her vector as input.

2. Bob then creates his own vector with a single coordinate whose value is $-\sum_{j>i} 10^j x_j$ and uses it as input into $\mathcal{A}$.

3. Now $\mathcal{A}$ should be estimating the norm of a vector with a single coordinate whose value is $\sum_{j\leq i} 10^j x_j$, and the norm of this vector is exactly $\sum_{j\leq i} 10^j x_j$. Finally, Bob guesses $x_i = 1$ if the output of the algorithm is at least $10^i/2$ and $x_i = 0$ otherwise.

In the case that $x_i = 0$ then the norm of vector is upper bounded by

$$\sum_{j=0}^{i-1} 10^j \leq \frac{10^i - 1}{10 - 1} \leq \frac{10^i}{9}$$

On the other hand, if $x_i = 1$ then the norm of vector is lower bounded by 10^i. Thus, using an algorithm that adequately approximates the norm means we can correctly identify x_i with a threshold at $10^i/2$.

■

Definition. In the Gap-Hamming problem, two individuals each have a bit vector in $\{0,1\}^n$, with the guarantee that the hamming distance is at least $n/2 + 2\varepsilon n$ or at most $n/2 + \varepsilon n$. The goal is to have them distinguish between these two scenarios.

Theorem 3. *Any stream algorithm that estimates norms must have space complexity* $\Omega(1/\epsilon^2)$.

Proof. It has been shown [2, 4, 3] that a lower bound of $\Theta(n)$ for the Gap-Hamming problem in the 1-way communication setting implies a $\Theta(1/\varepsilon^2)$ lower bound for any streaming algorithm estimating a norm. It also happens to be true for the 2-way communication case [1]. Then it suffices to demonstrate the linear lower bound for the Gap-Hamming problem by using an algorithm solving it to implement a sufficient protocol for the Index problem

1. Have Alice and Bob share some randomness by making $r^1, \ldots, r^t \in \{0,1\}^t$ available to both.

2. From $x \in \{0,1\}^t$, Alice generates $a \in \{0,1\}^t$ where

$$a_k = \text{Majority}_{j,x_J=1} r_j^k$$

3. Bob generates $b \in \{0,1\}^t$ where

$$b_k = r_i^k$$

4. Invoke the algorithm for the Gap-Hamming problem with a and b. If the result is at least $t/2 + \sqrt{t}$ then Bob predicts $x_i = 0$, otherwise he predicts $x_i = 1$.

In the case where $x_i = 0$, then $\mathbf{P}[a_k = b_k] = 1/2$ since a_k and b_k are independent. However, in the case where $x_j = 1$, then b_k participates in the vote for a_k, meaning the probability that a_k is the same as b_k is slightly more than the probability that they are not equal. More precisely, if we

approximate the distribution as binomial then the probability they are the same gains the middle value where the rest of the votes are splits. Using Stirling's approximation then gives us

$$\mathbf{P}[\text{Majority}(r_i^k, \dots) = r_i^k] \geq \frac{1}{2} + \binom{t}{t/2} 2^{-t} = \frac{1}{2} + \Theta\Big(\frac{1}{\sqrt{t}}\Big)$$

Computing the expected value of the hamming distance over all the bits gives $t/2 + x_i\sqrt{t}$. Thus, an appropriate algorithm that solves the Gap-Hamming problem can be used to resolve Augmented Indexing, implying that the lower bound for the space complexity of the Gap-Hamming problem is $\Omega(n)$.

■

References

[1] Amit Chakrabarti and Oded Regev. An optimal lower bound on the communication complexity of gap-hamming-distance. *SIAM Journal on Computing*, 41(5):1299–1317, 2012.

[2] Piotr Indyk and David Woodruff. Tight lower bounds for the distinct elements problem. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 283–288. IEEE, 2003.

[3] Thathachar S Jayram, Ravi Kumar, and D Sivakumar. The one-way communication complexity of hamming distance. *Theory of Computing*, 4(1):129–135, 2008.

[4] David Woodruff. Optimal space lower bounds for all frequency moments. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 167–175. Society for Industrial and Applied Mathematics, 2004.