CAS CS 591 B: Communication Complexity

Prof. Mark Bun — Fall 2019

Lecture Notes 12:

One-way communication, streaming

Reading.

- Rao-Yehudayoff Chapter 10, Roughgarden Chapters 1 & 2

We'll start looking in depth at an application of communication complexity to lower bounds for streaming algorithms. The data stream model is motivated by applications to analyzing massive data sets: ones which are too large to fit entirely in memory. For instance, we might be interested in monitoring traffic at a network switch and would like to be able to produce useful summary statistics about the traffic pattern. A streaming algorithm receives as input a sequence of elements $x_1, \dots, x_m$ from a domain $[n]$ one-by-one. With space $m \log n$ the algorithm can store all of the data. We'd like to be able to design algorithms which use far less space than – ideally $\text{polylog}(m, n)$ – as well as prove lower bounds on the memory requirement of such algorithms.

1 Frequency Moments

For a data stream $x_1, \dots, x_m \in [n]$ and universe item $j \in [n]$, let $f_j = |\{i : x_i = j\}|$ denote the frequency of item j in the stream. For $k \geq 0$, the k-th frequency moment of the stream is defined as

$$F_k = \sum_{j=1}^{n} f_j^k.$$

Interpreting $0^0 = 0$, the 0-th frequency moment F_0 is the number of distinct elements in the stream. F_1 is simply the number of elements in the stream m. F_2 is a natural measure of how far from uniform a data stream is. (A stream in which every element is distinct has $F_2 = m$, whereas a stream that is concentrated on a single element has $F_2 = m^2$.) And finally, we can define

$$F_\infty = \max_{j \in [n]} f_j$$

to be the maximum number of occurrences of any element.

If we allow for both randomization and approximation, then we can construct algorithms for estimating F_0 and F_2 with extremely low space.

Theorem 1 (Alon-Matias-Szegedy96)**.** *F_0 and F_2 can be approximated to multiplicative error $(1 \pm \varepsilon)$ with probability at least $1 - \delta$ using space $O((\log n + \log m) \cdot \log(1/\delta)/\varepsilon^2)$.*

The details of these algorithms are beyond the scope of this lecture, but you can find good expositions of them in the reading material.

To think about how to design an F_0 estimator, let $h : [n] \to [0, 1]$ be a random function. The streaming algorithm will simply keep track of the minimum value of $h(j)$ seen in the stream. To see why this is helpful, suppose we see k distinct elements in the stream. These k elements will be uniformly distributed over $[0, 1]$ and the expected minimum element will be $1/(k + 1)$. Hence if $u \in [0, 1]$ is the minimum value of $h(j)$ seen in the stream, a good estimator for k will be $1/u - 1$.

Similarly the idea for the F_2 algorithm is to start by designing a "basic" unbiased estimator for F_2, which can then be run many times in parallel to amplify its accuracy and its success probability. Let $h : [n] \to \{-1, 1\}$ be a random hash function. We initialize $Z = 0$, and every time an element $j \in [n]$ appears in the stream, we add $h(j)$ to Z. Finally, we output Z^2. To see that this is an unbiased estimator we compute

$$\begin{aligned} \mathbb{E}[Z^2] &= \mathbb{E}\left(\sum_{j=1}^{n} h(j) f_j\right)^2 \\ &= \mathbb{E}\left(\sum_{j=1}^{n} h(j)^2 f_j^2 + \sum_{j \neq k} h(j)h(k) f_j f_k\right) \\ &= F_2. \end{aligned}$$

Some details that need to be checked include: Showing that the estimator has low variance, showing that the hash function can be stored in small space (it actually needs to be derandomized using 4-wise independence), and showing that the error/failure probabilities can be amplified.

Can either of these algorithms be generalized to $k > 2$? It turns out the answer is no, and in general there is an $\Omega(n^{1-2/k})$ space lower bound for computing the k-th frequency moment for $k > 2$. The proofs go by way of reductions to communication complexity. To illustrate, let's see a lower bound of $\Omega(n)$ for the case of $k = \infty$.

Theorem 2. *Any streaming algorithm which, with probability at least* $2/3$ *estimates* F_∞ *to multiplicative error* (1 ± 0.2) *requires space* $\Omega(\min\{m, n\})$.

Proof. We show that a streaming algorithm for estimating F_∞ with space s on a stream of length $m = n$ would allow us to solve the Disjointness problem using communication s. The reduction is as follows. On input x, Alice initializes the streaming algorithm and feeds it every $i \in [n]$ such that $x_i = 1$. She then sends the state of the streaming algorithm to Bob using s bits of communication. Bob then feeds the algorithm every $i \in [n]$ for which $y_i = 1$. Observe that if $\mathrm{DISJ}_n(x, y) = 0$ then $F_\infty \leq 1$ since every i appears at most 1 time in the stream. On the other hand, if $\mathrm{DISJ}_n(x, y) = 1$ we have $F_\infty \leq 2$. Being able to approximate F_∞ with small multiplicative error allows us to distinguish these two cases and thus solve Disjointness. $\square$

2 One-Way Communication

The proof of Theorem 2 did not use the full power of the Disjointness lower bound. The protocol for Disjointness that we reduced to involved only a single message from Alice to Bob. Such "one-way" communication lower bounds are almost always sufficient to prove lower bounds in streaming.

Definition 3. Let $f : X \times Y \to \{0, 1\}$ and $\varepsilon > 0$. The randomized one-way communication complexity, denoted $R_\varepsilon^{A \to B}(f)$ is the least cost of a public-coin randomized protocol computing f to error ε in which the only communication consists of a single message from Alice to Bob.

2.1 Lower Bound for Indexing

In the Indexing problem IND_n, Alice is given a string $x \in \{0,1\}^n$ and Bob is given an index $i \in [n]$. The goal is for Bob to compute x_i given one-way communication from Alice. Note that if we allow communication from Bob to Alice, this problem is easy since he can just send the $\log n$ bit index i. Nevertheless, when communication only goes from Alice to Bob, this problem requires communication $\Omega(n)$.

To state the result, it will be helpful for us to define the binary entropy function $h(\varepsilon) = \varepsilon \log(1/\varepsilon) + (1-\varepsilon)\log(1/(1-\varepsilon))$. This is simply the entropy of a binary random variable B which takes value 1 with probability ε.

Theorem 4. *For every $\varepsilon > 0$, $R_\varepsilon^{A\to B}(\mathrm{IND}_n) \geq (1-h(\varepsilon))n$.*

Proof. As one should expect from the statement, the proof goes by way of information theory. Let Π be a one-way protocol computing IND_n with error ε. Let A be uniformly random over $\{0,1\}^n$ and let B be uniform over $[n]$. Our goal will be to show that $I(A;\Pi(A,B)) \geq (1-h(\varepsilon))n$. To do this, let's write

$$I(A;\Pi) = H(A) - H(A|\Pi) = n - H(A|\Pi)$$

so the remaining goal is to show that $H(A|\Pi) \leq h(\varepsilon)n$. To show this we use the chain rule to obtain

$$H(A|\Pi) = \sum_{i=1}^{n} H(A_i|\Pi A_{\leq i}) \leq \sum_{i=1}^{n} H(A_i|\Pi).$$

Now for every $i \in [n]$, we have $H(A_i|\Pi) = H(A_i|\Pi, B=i)$ since (A_i,Π) is independent of B. So it now suffices to show that $H(A_i|\Pi, B=i) \leq h(\varepsilon)$ for every i. It's intuitive that we should be able to upper bound this quantity by a constant. If Π lets us predict A_i with high probability, then conditioning on Π should leave little remaining uncertainty about A_i. The technical tool we need is (a special case of) Fano's Inequality:

Lemma 5 (Fano)**.** *Let $X \in \{0,1\}$ be a binary random variable, let M be a random variable jointly distributed with X, and let g be a function of M. Let E be the event that $g(M) \neq X$. Then*

$$H(X|M) \leq H(E).$$

To use Lemma 5 to complete the proof, let E be the event that Bob's output disagrees with A_i. Then by correctness of the protocol, $E=0$ with probability $1-\delta$ and $E=1$ with probability δ for some $0 < \delta \leq \varepsilon$. Hence

$$H(A_i|\Pi, B=i) \leq H(E|B=i) = h(\delta) \leq h(\varepsilon)$$

as we wanted. ∎

For completeness, let's now give a proof of the special case of Fano's Inequality we used.

Proof of Lemma 5. Since E is a function of M and X, we have $H(E|M,X) = 0$. Therefore,

$$\begin{aligned} H(X|M) &= H(X|M) + H(E|M,X) \\ &= H(X,E|M) \\ &= H(E|M) + H(X|M,E) \\ &\leq H(E), \end{aligned}$$

where the last inequality follows because M, E completely determines X. ∎

The general statement of Fano's Inequality is as follows.

Theorem 6 (Fano's Inequality)**.** *Let $X \in \mathcal{X}$ be a random variable, let M be jointly distributed with X, and let g be a function of M. Let E be the event that $g(M) \neq X$. Then*

$$H(X|M) \leq H(E) + \Pr[E] \log(|\mathcal{X}| - 1).$$

2.2 Lower Bound for Gap Hamming

In the Gap Hamming problem GH_n, Alice is given a string $x \in \{0,1\}^n$ and Bob is given a string $y \in \{0,1\}^n$ and the goal is to distinguish between the following two cases: 1) $|x-y| \leq n/2 - c\sqrt{n}$ and 2) $|x-y| \geq n/2 + c\sqrt{n}$. Here $c > 0$ is a parameter of the problem which is independent of n (chosen to make proofs work).

We can use a reduction to Indexing to show that the Gap Hamming problem requires linear one-way communication. Note that a linear lower bound holds for two-way randomized communication as well (but the proof is more involved).

Theorem 7. $R_{1/3}^{A \to B}(\mathrm{GH}_n) \geq \Omega(n)$.

Proof. We give a randomized reduction from Indexing to Gap Hamming. Given an instance (x, i) of the Indexing problem for m odd, $x \in \{0,1\}^m$ and $i \in [m]$, Alice and Bob will (with public randomness but zero communication) construct an instance $(x', y') \in \{0,1\}^n \times \{0,1\}^n$ for $n = O(m)$ such that $\mathrm{GH}_n(x', y') = \mathrm{IND}_n(x, i)$ with high probability.

The inputs x', y' will each be constructed one bit at a time independently. To construct a single pair of bits (a, b) Alice and Bob will look at a fresh m-bit portion of the public randomness $r \in \{0,1\}^m$. Bob will set $b = r_i$, i.e., the i-th bit of the random string. Alice will set $a = 1$ if $|x - r| < m/2$ and $x'_j = 1$ if $|x - r| > m/2$. The idea is that if $x_i = 1$, then the bits a, b are correlated, but if $x_i = 0$ then the bits are anti-correlated. More precisely, let E be the event that $|x_{-i} - r_{-i}| = m/2$. If E occurs, then $a = 1$ iff $x_i = r_i$. Hence conditioned on E we have that $x_i = 1 \implies a = b$ and $x_i = 0 \implies a \neq b$. On the other hand, conditioned on $\bar{E}$, the bit a is determined by r_{-i} and is a coin flip independent of b. So for some constant $c = \Omega(1)$,

$$\begin{aligned}
\Pr[a = b] &= \Pr[a = b|E]\Pr[E] + \Pr[a = b|\bar{E}]\Pr[\bar{E}] \\
&= \Pr[a = b|E] \cdot \frac{2c'}{\sqrt{m}} + \frac{1}{2} \cdot \left(1 - \frac{2c'}{\sqrt{m}}\right) \\
&= \begin{cases} \frac{1}{2} - \frac{c'}{\sqrt{m}} & \text{if } x_i = 0 \\ \frac{1}{2} + \frac{c'}{\sqrt{m}} & \text{if } x_i = 1. \end{cases}
\end{aligned}$$

Hence for every j independently we have $\Pr[x'_j = y'_j] \leq 1/2 - c'/\sqrt{m}$ if $x_i = 0$ and $\Pr[x'_j = y'_j] \geq 1/2 + c'/\sqrt{m}$ if $x_i = 1$. Repeating $n = O(m)$ times, we get that there is a constant c such that with high probability (say 8/9) $|x' - y'| \geq n/2 + c\sqrt{n}$ if $x_i = 0$ and $|x' - y'| \leq n/2 - c\sqrt{n}$ if $x_i = 1$. So if we can solve GH_n to error 1/3 with sublinear communication, we can solve IND_m to error 4/9 with sublinear communication. □

3 Frequency Moment Lower Bound from Gap Hamming

Theorem 8. *A randomized algorithm for computing F_0 to within a multiplicative $(1 \pm c/\sqrt{n})$ requires space $\Omega(n)$.*

Proof. Suppose we have a streaming algorithm which computes F_0 to within a multiplicative $(1 \pm c/\sqrt{n})$ factor using space s. We construct a one-way communication protocol which computes Hamming distance up to an additive $\pm c\sqrt{n}$ error using space $s + \log n$ as follows. Alice and Bob interpret their inputs x, y as the indicator vectors for sets $A, B \subseteq [n]$. Alice loads A into the streaming algorithm and sends the state to Bob, who then loads B. She also sends him $|A|$ using an additional $\log n$ bits of communication. Observe that $|x - y| = |A \Delta B| = |A \setminus B| + |B \setminus A| = (|A \cup B| - |B|) + (|A \cup B| - |A|)$. Noting that $F_0 = |A \cup B|$ we have $|x - y| = F_0 - |x| - |y|$. Since $F_0 \in [0, n]$, a multiplicative $(1 \pm c/\sqrt{n})$ approximation translates into an additive $\pm c\sqrt{n}$ approximation to $|x - y|$. □

A padding argument can be used to show that for $\varepsilon > c'/\sqrt{n}$, a multiplicative $(1 \pm \varepsilon)$ approximation to F_0 requires space $\Omega(1/\varepsilon^2)$. Specifically, we can let $k = 1/\varepsilon^2$ and reduce from GH_k as follows. On inputs x, y to GH_k, construct x', y' by appending $n - k$ zeroes to the end of each. Estimating $F_0 \in [0, k]$ to error $(1 \pm \varepsilon)$ on the resulting stream translates into additive error $k \cdot \varepsilon = O(\sqrt{k})$, which is ruled out by the lower bound of $\Omega(k)$ for computing GH_k.