CSE 522: Sublinear (and Streaming) Algorithms **Spring 2014**

Lecture 12: Communication Complexity, Streaming Lower Bounds

May 7, 2014

Lecturer: Paul Beame *Scribe: Paul Beame*

We recall communication complexity definitions, $D(f)$, $D^{A\to B}(f)$, $R_\delta(f)$, $R_\delta^{A\to B}(f)$, $D_t(f)$, $R_{t,\delta}(f)$.

Last time we proved that $D(EQ) = n + 1$ and showed how this implies that any deterministic protocol that exactly computes F_0 requires space $\Omega(n)$. We did this by mapping $x, y \in \{0,1\}^n$ to $\overline{x}, \overline{y} \subseteq [2n]$ such that $F_0(\overline{x}\,\overline{y}) = n + \Delta(x, y)$ where Δ is the Hamming distance between x and y. We now derive essentially the same lower bound for approximately computing F_0.

Lemma 0.1. *Any deterministic streaming algorithm that approximates F_0 within 7.5% error requires space $\Omega(n)$.*

Proof. We follow a similar pattern to the reduction we showed from computing EQ to exactly computing F_0 but instead we first encode x and y in such a way that if $x \neq y$ then the Hamming distance between their encodings is large. In particular, we will define an *encoding* function $E : \{0,1\}^n \to \{0,1\}^{3n}$ such that $x \neq y$ implies that $\Delta(E(x), E(y)) \geq n/2$. Suppose for now that we have such a function E. On inputs x and y, respectively, Alice and Bob will simulate the streaming algorithm for approximating F_0 on input stream $\overline{E(x)}\,\overline{E(y)}$ where again $\overline{z}$ is an encoding of z that represents z_i by the number $2i - z_i$. Since $|E(x)| = 3n$, $\overline{E(x)}$ will consist of $3n$ elements from $[4n]$. Alice will send Bob the contents of the storage after $\overline{E(x)}$ has been read. We have $F_0(\overline{E(x)}\,\overline{E(y)}) = 3n + \Delta(E(x), E(y))$ so $F_0(\overline{E(x)}\,\overline{E(y)}) = \begin{cases} 3n & \text{if } x = y \\ 3n + n/2 = 7n/2 & \text{if } x \neq y. \end{cases}$ Given an approximation to F_0 within 7.5% then if $x = y$ the output must be at most $3.225n$, which is not within 7.5% of $7n/2$, so that protocol could determine whether or not $x = y$ given the approximation to F_0 and hence the storage must be at least $n + 1$.

It remains to show how E can be constructed. We do this by showing that there is a subset C of size at least 2^n inside $\{0,1\}^{3n}$, no two of which are with Hamming distance $n/2$ of each other. Alice and Bob can agree ahread of time on this subset and on some fixed map E from the elements of $\{0,1\}^n$ to this set. (The properties of the set C we want are those of a good *error-correcting code*. In other applications it is important that such codes be explicitly constructed and have good computational properties but in our application this only has to be an existential argument since Alice and Bob are computationally unbounded.)

Write $N = 3n$. The way we find this subset will be via a greedy argument: We will maintain a set $W \subseteq \{0,1\}^N$ of candidate elements. At each step we choose some arbitrary element of W, add it

to C and then remove all strings in W within Hamming distance $n/2 = N/6$ of this string. It is immediate that no two strings in W are within Hamming distance $n/2$.

There are precisely $\binom{N}{k}$ strings in $\{0,1\}^N$ at Hamming distance precisely k from any fixed string. Therefore there are at most $B(N,k) = \binom{N}{0} + \binom{n}{1} + \cdots + \binom{N}{k}$ strings at distance at most $N/6$ from any given string where $k = \lfloor N/6 \rfloor$. We can bound this in terms of the *binary entropy function* $H_2(\delta) = \delta \log_2(1/\delta) + (1-\delta)\log_2(1/(1-\delta))$.

Proposition 0.2. *For $k \leq \delta N \leq N/2$, $B(N,k) \leq 2^{H_2(\delta)N}$.*

Proof. By the binomial theorem, for any $\delta \leq 1/2$,

$$\begin{aligned} 1 &\geq \sum_{i \leq k} \binom{N}{i} \delta^i (1-\delta)^{N-i} \\ &\geq \sum_{i \leq k} \binom{N}{i} \delta^{\delta N} (1-\delta)^{N(1-\delta)} \qquad \text{since } \delta \leq 1/2 \\ &\geq \sum_{i \leq k} \binom{N}{i} [\delta^{\delta} (1-\delta)^{(1-\delta)}]^N \\ &= B(N,k) 2^{-H_2(\delta)N} \end{aligned}$$

which immediately implies the claim. □

Using this bound we see that one can obtain a set C of size at least $2^{(1-H_2(\delta))N}$ since at most $2^{H_2(\delta)N}$ elements are removed from W for each element of C chosen. In our case $\delta = 1/6$ and $H_2(1/6) = 0.650022... < 2/3$. Therefore $|C| \geq 2^{N/3} = 2^n$ as required. □

For randomized protocols we can see that the equality function EQ has protocols with small communication complexity. This should not be surprising since we can efficiently approximate F_0 using randomized protocols, but it is a particularly clean argument.

Theorem 0.3. *$R_\delta^{A \to B}(EQ)$ is $O(\log(1/\delta))$.*

Proof. Let $k = \lceil \log_2(1/\delta) \rceil$. Alice and Bob use the shared random string to represent a random $k \times n$ binary matrix A. Alice computes $A \cdot x \bmod 2$ and sends this length k binary vector to Bob. Bob computes $A \cdot y \bmod 2$ and outputs 1 if and only if it agrees with what Alice sent.

$Ax \bmod 2 = Ay \bmod 2$ if and only if $A(x-y) = 0 \pmod 2$. If $x = y$ then this always holds. Write $z = x - y$. Each entry of $Az \pmod 2$ is a random linear combination $r_1 z_1 + \ldots + r_n z_n$ taken modulo 2. Let z_i be some fixed non-zero entry in z, which must be ± 1 and hence equivalent to 1 modulo 2. For each choice of $r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_n$, there is one choice of r_i that will make the total even and one choice that will make it odd, hence the probability that the i-th entry of Az

is $0 \bmod 2$ is precisely 1/2. Since the rows are independent $Az \bmod 2$ is all 0 with probability $1/2^k \leq \delta$ as required. □

In order to obtain lower bounds for $R_\delta(f)$ for various other functions, it is useful to add one more measure of the computational complexity of functions:

For any probability distribution μ on $X \times Y$ and error δ, define $D^\mu_\delta(f)$ to be the minimum number of bits required by any deterministic protocol to compute f correctly for all but a δ fraction of inputs (x, y) under distribution μ. This is called the δ-error *distributional* communication complexity of f. We also have the analogous definition for one-way protocols.

The following lemma, the easy half of a lemma due to Yao, relates randomized and distributional communication complexity.

Lemma 0.4. *$R_\delta(f) \geq D^\mu_\delta(f)$ and $R^{A\to B}_\delta(f) \geq D^{\mu,A\to B}_\delta(f)$.*

Proof. A randomized protocol with a given complexity C is a distribution over deterministic protocols with that complexity. For each input, the correctness of the randomized protocol implies that the average error of these deterministic protocols is at most δ. Therefore if we choose an input according to μ and a random one of these deterministic protocols, the error will be at most δ. The order of these choices is not important, so there must be at least one protocol whose average error under μ is at most δ, which is what we needed. □

We now describe a few functions whose communication complexity is important for showing the limitations of streaming algorithms:

Define $Index_n : \{0,1\}^n \times [n] \to \{0,1\}$ by $Index(u, j) = u_j$.

If Bob could speak first, computing $Index$ would require at most $\log_2 n$ bits since Bob could simply send j to Alice and she could output u_j.

However, it is easy to see that $D^{A\to B}(Index) \geq n$ since Alice's message on input u must be the same independent of Bob's input and with the different inputs j, we would be able to reconstruct Alice's input from the different possible answers Bob would have to be able to produce from her message.

It requires a more work to show,

Theorem 0.5. *$R^{A\to B}_\delta(Index_n) \geq (1 - H_2(\delta))n$.*

Define $GapHamming : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ by

$$GapHamming(x, y) = \begin{cases} 1 & \text{if } \Delta(x, y) \geq n/2 + \sqrt{n} \\ 0 & \text{if } \Delta(x, y) \leq n/2 - \sqrt{n} \\ \text{don't care} & \text{otherwise} \end{cases}.$$

Theorem 0.6 (Chakrabarti,Regev)**.** $R_{1/3}(GapHamming)$ *is* $\Omega(n)$.

We will later prove the one-way lower bound for $Index_n$ and use this to derive a one-way lower bound for $GapHamming$. The general argument would take too long to develop.

The final problem we will consider is Unique Disjointness function $UDISJ_n^t$ which is for t-party communication. We identify its inputs $x_1, \ldots, x_t \in \{0,1\}^n$ as characteristic vectors of subsets $A_1, \ldots, A_t \subseteq [n]$.

$$UDISJ_n^t(A_1, \ldots, A_t) = \begin{cases} 0 & \text{for all } i \neq j,\ A_i \cap A_j = \emptyset \\ 1 & \text{there is a } k \in [n] \text{ s.t. for all } i \neq j,\ A_i \cap A_j = \{k\} \\ \text{don't care} & \text{otherwise.} \end{cases}$$

Theorem 0.7. $R_{t,1/3}(UDISJ_n^t)$ *is* $\Omega(n/t)$

The first bound for $t \geq 3$ was an $\Omega(n/t^3)$ bound by Alon, Matias, and Szegedy and this was improved by several authors until Gronemeier gave the above bound, which is optimal; earlier Chakrabart, Khot, and Sun had shown the same lower bound for t-party one-way communication complexity (which was sufficient for the application to streaming below) but were off by $\log t$ factor in the general case. The techniques for all of these lower bounds use information theory arguments that would take several lectures to develop.

Using this theorem we can obtain an essentially tight lower bound for approximating large frequency moments.

Theorem 0.8. *For* $p > 2$*, any* $1/3$*-error streaming algorithm on inputs from* $[M]$ *that approximated* F_p *within a factor better than* 2 *requires space* $\Omega(M^{1-2/p})$.

Proof. Let $M = n$. Choose $t = (4n)^{1/p}$. On input $A_1, \ldots, A_t$ for $UDISJ_n^t$ the players run the streaming algorithm for approximating $F_p(A_1 \ldots A_t)$. The i-th for $i < t$ player will simulate the computation while the elements of A_i are read and will write the contents of the storage at the end of its segment on the blackboard.

If $A_1, \ldots, A_t$ are disjoint then each of the elements in $[n]$ occurs are most once in the input so $F_p(A_1 \ldots A_t) \leq n$.

If $A_1, \ldots, A_t$ intersect in k then $F_p(A_1 \ldots A_t) \geq f_k^p = t^p = 4n$. It follows that any approximation of F_p within a factor of 2 can distinguish the two cases and hence allow the players to solve $UDISJ_n^t$ using at most $(t-1)S$ bits of communication where S is the space bound. It follows that $(t-1)S$ is $\Omega(n/t)$ and hence S is $\Omega(n/t^2) = \Omega(n/n^{2/p}) = \Omega(n^{1-2/p})$. □

We now see how to use the $GapHamming$ lower bound to derive lower bounds for the computation of F_0. This lower bound approach is due to Woodruff.

Theorem 0.9. *For $\varepsilon = 1/\sqrt{n}$, any streaming algorithm computing F_0 with error at most ε requires $\Omega(1/\varepsilon^2)$ space.*

Proof. For $x, y \in \{0,1\}^n$ that are inputs to $GapHamming$ if we view x and y as subsets of $[n]$, then $F_0(xy) = |x \cup y|$. It is not hard to see that as sets, x and y cover the elements of $x \cap y$ twice and cover the elements of the symmetric difference $x\Delta y = (x \cup y) - (x \cap y)$ once each. It is therefore easy to see that $|x| + |y| + |x\Delta y| = 2|x \cup y|$. The size of $x\Delta y$ as a set is precisely the Hamming distance $\Delta(x, y)$. Therefore we have $|x| + |y| + \Delta(x, y) = 2F_0$ or, alternatively, $\Delta(x, y) = 2F_0 - |x| - |y|$.

we can use any streaming approximation for F_0 to compute a very good approximation to $\Delta(x, y)$ with communication not much larger than the space of the streaming algorithm. Alice will simulate the streaming algorithm and send both the current contents of the storage as well as $|x|$ to Bob who will compute the approximation to F_0 and hence of $\Delta(x, y)$ from which he can compute $GapHamming$. Sending $|x|$ requires only $\log_2 n$ bits. If we can approximate F_0 within a factor better than $1 \pm \varepsilon$ Then we can approximate $\Delta(x, y)$ within a factor $1 \pm O(\varepsilon)$ since wlog $|x| + |y|$, F_0, and $2F_0 - |x| - |y|$ are all linear in n. (The players can easily determine $|x| + |y|$ and reject any inputs for $GapHamming$ for which $|x| + |y|$ is sublinear in n. Any approximation of $\Delta(x, y)$ within a $1\pm\varepsilon$ factor for $\varepsilon = 1/\sqrt{n}$ can compute $GapHamming$ and hence requires communication $\Omega(n) = \Omega(1/\varepsilon^2)$. □

We finish with the lower bound proofs for $Index_n$ and $GapHamming$. For the former we need to use entropy. For a random X, write $p_x = \mathbb{P}[X = x]$ and define $H(X) = \sum_x p_x \log_2(1/p_x)$. $H(X, Y)$ is just H applied to the pair of random variables (X, Y) using $p_{xy} = \mathbb{P}[X = x, Y = y]$.

We have the following facts:

- If $X \in S$ then by the convexity of the logarithm, $H(X) \leq \log_2 |S|$.
- If we define $H(X|Y) = \mathbb{E}_y H(X|Y = y)$ where $X|Y = y$ is the random variable given by X conditioned on $Y = y$, then $H(X|Y) = H(X, Y) - H(Y)$. This latter is called the *chain rule*.

The randomized lower bound for the $Index$ function follows immediately from the following lemma.

Lemma 0.10. *Let μ be the distribution of (X, J) on $\{0,1\}^n \times [n]$ that chooses X uniformly on $\{0,1\}^n$ and independently chooses J uniformly from $[n]$. Then $D_\delta^{A\to B,\mu}(Index_n) \geq (1 - H_2(\delta))n$.*

Proof. Let Π (for "protocol") be the distribution on the message sent by Alice. Given J and Π, Bob's answer is fixed, so in order for Bob to give the correct answer X_J with probability at least

$1-\delta$, X_J itself must have a bias of at least $1-\delta$ towards some fixed value conditioned on the values of Π and J. Therefore it must satisfy $H(X_J|\Pi, J) \leq H_2(\delta)$. Expanding on this we have

$$\begin{aligned}
H_2(\delta) &\geq H(X_J|\Pi, J) \\
&= \sum j = 1^n H(x_j|\Pi, J = j)\, \mathbb{P}(J = j) \\
&= \frac{1}{n}\sum_{j=1}^{n} H(X_j|\Pi, J = j) \\
&= \frac{1}{n}\sum_{j=1}^{n} H(X_j|\Pi) \qquad \text{since } \Pi \text{ and } X_j \text{ do not depend on } J \\
&\geq \frac{1}{n}\sum_{j=1}^{n} H(X_j|\Pi, X_1, \ldots, X_{j-1}) \qquad \text{since adding conditions only reduces entropy} \\
&= \frac{1}{n}\sum_{j=1}^{n} [H(X_1, \ldots, X_j, \Pi) - H(X_1, \ldots, X_{j-1}, \Pi)] \qquad \text{by the chain rule} \\
&= \frac{1}{n}\left[H(X_1, \ldots, X_n, \Pi) - H(\Pi)\right] \qquad \text{since the sum telescopes} \\
&= \frac{1}{n}\left[H(X_1, \ldots, X_n) - H(\Pi)\right] \qquad \text{since } \Pi \text{ is determined by } X_1, \ldots, X_n \\
&= \frac{1}{n}\left[n - H(\Pi)\right] \qquad \text{since } X \text{ in uniform} \\
&= 1 - H(\Pi)/n.
\end{aligned}$$

Rearranging gives the claim of the lemma. □

The following proof is due to Jayram, Kumar, and Sivakumar. The proof by another argument was due to Woodruff.

Theorem 0.11. $R_{1/3}^{A\to B}(GapHamming)$ *is* $\Omega(n)$.

Proof Sketch. Assume without loss of generality that n is odd. The general idea is a reduction using the lower bound for $Index_n$. The reduction itself will use a large number of additional shared random bits. We show how to derive a randomized protocol for $Index$ given a protocol for $GapHamming$. The idea will be a method to produce correlated random bits for Alice and Bob without interaction so that if u_j is 1, the bits will differ with probability at least $1/2 + c/\sqrt{n}$ and if u_j is 0, they will differ with probability at most $1/2 - c/\sqrt{n}$. Alice and Bob will repeat this independently n times to produce correlated random bit strings in which each pair of bits has the same bias towards either equality or difference. Hence by Chernoff bounds these strings will almost surely have Hamming distance either at least $n/2 + c'\sqrt{n}$ or at most $n/2 - c'\sqrt{n}$

It remains to show how a single pair of correlated bits is produced: Alice and Bob will use the random bits to produce a random string $r \in \{-1, 1\}^n$. Alice will interpret her input u as a vector

of ± 1 entries, v, given by $v_i = (-1)^{u_i}$. Alice's bit will be 1 if and only if $\sum_i r_i v_i$ is positive. Bob's bit will be 1 if and only if $r_j = 1$

Now for each fixed value of $(r_i)_{i \neq j}$ write $w = \sum_{i \neq j} r_i v_i$. Since n is odd, w is the sum of an even number of ± 1 values and hence is even.

If $w \neq 0$ then $|w| \geq 2$ and the sign of $\sum_i r_i v_i = r_j v_j + w$ must be the same as the sign of w, and hence independent of r_j. Therefore in this case, Alice's and Bob's bits are independent and hence are equal with probability $1/2$.

If $w = 0$ then Alice's bit is the sign of $r_j v_j = r_j (-1)^{u_j}$. Therefore Alice's bit is the same as that of Bob's bit iff $u_j = 0$, which is precisely the condition we want to hold. Now $w = 0$ happens with probability $\binom{n-1}{(n-1)/2}$ which is $\sim 1/\sqrt{2\pi n}$. Therefore over the random choice of r this has probability of being different of $1/2(1 + 1/\sqrt{2\pi n})$ if $u_j = 1$ and of $1/2(1 - 1/\sqrt{2\pi n})$ if $u_j = 0$ $\square$