Communication Complexity 16:198:671 3/2/10

Lecture 7

Lecturer: Troy Lee Scribe: Luke Friedman

1 Streaming Algorithms

Let $S \in [n]^m$. (i.e. S is a string of length m with elements in the range 1 to n). Suppose we want to compute $f(S)$ for some function f, but m is huge, so that it is impractical to try to store all of m. In a streaming algorithm we assume that we see the input S one symbol at a time with no knowledge of what future symbols will be. The question then is how many bits must be kept in memory in order to successfully compute $f(S)$. If C is the number of bits of memory used then in general we are looking for algorithms for which $C \approx O(\log n + \log m)$. (Note that C includes all memory that is used during the computation of the algorithm, not just information that is stored about S)

Often we consider randomized streaming algorithms that compute approximations to $f(S)$. In this case, the algorithm correctly computes a c-approximation if whp it compute a value in the range $[\frac{1}{c}f(S), cf(s)]$.

Example 1.

Let $S \in [n]^{n-1}$, and assume that S contains every element in $[n]$ except for the number x. Let $f(S) = x$. A streaming algorithm to compute $f(S)$ can maintain a sum $T_k = \sum_{i=1}^{k} S_i$. At the end of the stream, it outputs $f(S) = \frac{n(n+1)}{2} - T_{n-1}$. This algorithm uses $O(\log n)$ memory.

1.1 Computing Frequency Moments

The results from this section and the next are from the Gödel prize winning paper "The Space Complexity of Approximating the Frequency Moments" by Alon, Matias, and Szegedy '96.

Let $S \in [n]^m$ and let

$$M_i = |\{j \in [m] : S_j = i\}|$$

Then the kth frequency moment, denoted F_k is

$$F_k = \sum_{i=1}^{n} M_i^k$$

So for example F_0 equals the number of distinct elements in the stream and F_1 equals the length of the stream. Also, by definition, F_∞ equals the number of occurrences of the most frequent element in the stream.

We can lower bound the memory needed to compute F_∞ by reducing the problem to the communication complexity of computing disjointness:

Claim 2. *If there exists a streaming algorithm to compute F_∞ using C bits of memory, then there exists a protocol to compute disjointness using C bits of communication.*

Proof. Suppose there exists a streaming algorithm A that computes F_∞ using C bits of memory.

Let (x, y) be an input to DISJ, where Alice gets x, Bob gets y, and $x, y \in \{0,1\}^n$.

Alice converts her input into a stream $a_x = \{i : x_i = 1\}$. Bob similarly converts his input into a stream $b_y = \{i : y_i = 1\}$. Alice then runs A on a_x and sends C bits of information to Bob representing the state of the algorithm after it has reached the end of a_x. Bob can then finish running A on b_y to compute $F_\infty(S)$, where $S = a_x \circ b_y$.

If $\text{DISJ}(x, y) = 1$ then $F_\infty(S) = 1$ and otherwise $F_\infty(S) = 2$. Therefore Alice and Bob are able to solve DISJ using C bits of communication. □

Because we have a lower bound of n for the communication complexity of the disjointness problem, as a corollary we get a lower bound of n on the amount of memory needed to compute F_∞. Also, since DISJ requires $\Omega(n)$ bits of communication even for randomized protocols, the result applies to randomized streaming algorithm as well. Furthermore, the gap between 1 and 2 of the possible values of $F_\infty(S)$ means that the result can applied in the approximation case as well.

1.2 Multiparty Number in the Hand Model and Frequency Moments

We can generalize the normal communication complexity model to allow more than two players. In the Multiparty Number in the Hand Model there are p players, and the ith player gets an input x_i. Every player is able to see only his own input. We imagine there is a blackboard, and when a player communicates he writes his message on the blackboard and all players can see it. The communication cost of computing a function $f(x_1, x_2, \ldots, x_p)$ is the total number of bits written on the blackboard during the worst case run of an optimal protocol.

We will consider a certain promise problem unique disjointness UDISJ_p in this model. The problem is only defined for inputs $x_1, \ldots, x_p$ which either have a unique intersection, in which case the value is 0, or if the inputs are *pairwise* disjoint, in which case the value is 1.

It is possible using information theoretic techniques as discussed in the previous lecture to give strong randomized lower bounds for this problem. In particular, Gronemeier [Gro09] has recently shown (improving previous bounds [AMS99, BYJKS04, CKS03]) that

$$R(\text{UDISJ}_p) = \Omega\left(\frac{n}{p}\right)$$

We can derive lower bounds for the problem of computing the kth frequency moment of a stream by reducing that problem to UDISJ_p where $p = n^{1/k}$.

Claim 3. *If there exists a streaming algorithm to compute F_k using C bits of memory, then there exists a $C \cdot n^{1/k}$ bit $n^{1/k}$ party protocol for $UDISJ_{n^{1/k}}$.*

Proof. The proof is similar to the reduction from F_∞. Fix k and let $p = n^{1/k}$. Suppose there exists a streaming algorithm A to compute F_k using C bits of memory. On input $(x_1, \ldots x_p)$ to UDISJ$_p$, player i converts his input x_i into a stream $a_{x_i} = \{i : \text{ the } i\text{th bit of } x_i \text{ equals } 1\}$. We assume that the total number of ones in all the strings $x_1, \ldots, x_p$ is n. The players then together run A on the stream $S = a_{x_1} \circ \ldots \circ a_{x_p}$. Whenever player i finishes running A on the portion of S corresponding to a_{x_i}, he writes the state of the algorithm on the blackboard using C bits.

At the end of the algorithm the players have computed $F_k(S)$ using $Cn^{1/k}$ bits of communication. This is sufficient to determine UDISJ$_p(x_1, \ldots, x_p)$ since if UDISJ$_p(x_1, \ldots, x_p) = 1$ then $F_k(S) = n$, and if UDISJ$_p(x_1, \ldots, x_p) = 0$ then $F_k(S) \geq n - 1 + n = 2n - 1$. □

As a corollary we get that computing F_k requires space $\Omega n^{1-2/k}$.

1.3 An algorithm for computing F_0

We now give an efficient randomized streaming algorithm for approximating F_0 from [AMS99]. (This is not related to communication complexity but is a nice example of a nontrivial streaming algorithm).

Let $S = [n]^m$ be the stream and let d be some integer such that $2^d \geq n$. We will view each element of the string S_i as an element of $GF(2^d)$ The streaming algorithm works as follows:

```
1  MAX-R ← 0
2  Randomly choose a, b ∈ GF(2^d)
3  foreach element S_i ∈ S
4        do
5           Z_i ← aS_i + b
6           r_i ← max r such that the r rightmost bits of Z_i are all 0
7           if r_i > MAX-R
8              then MAX-R ← r_i
9           return 2^MAX-R
```

During its execution the algorithm only needs to keep track of a, b and $MAX\text{-}R$, each of which takes $O(\log n)$ bits, so the algorithm uses $O(\log n)$ bits.

Now we show that the algorithm is correct with high probability.

Claim 4.

$$\Pr[2^{\text{MAX-R}} \in [F_0/c, c \cdot F_0]] \geq 1 - 2/c$$

Proof. Fix an r, and for each x that appears in the stream S let $W_{x,r}$ be the indicator random variable that is 1 if the r_i in line 6 of the algorithm when $S_i = x$ is at least r and 0 otherwise. Because the variable Z_i is uniformly distributed over elements of $GF(2^d)$, $\mathbf{E}[W_{x,r}] = 1/2^r$ for any x. Let X be the set of all elements $x \in [n]$ that appear in the stream. Define Y_r as

$$Y_r = \sum_{x \in X} W_{x,r}$$

By linearity of expectation we have that $E[Y_r] = F_0/2^r$.

Suppose that $2^r > c \cdot F_0$. In this case we have that $E[Y_r] < 1/c$. Because Y_r is a nonnegative integral random variable, this implies that $Pr[Y_r > 0] < 1/c$.

On the other hand, suppose $2^r < F_0/c$. Because the $W_{x,r}$ are pairwise independent, we have that

$$\mathrm{Var}(Y_r) = F_0 \frac{1}{2^r}(1 - \frac{1}{2^r}) < \frac{F_0}{2^r} = E[Y_r]$$

Therefore, by Chebyshev's Inequality,

$$Pr[Y_r = 0] \leq \frac{\mathrm{Var}(Y_r)}{E[Y_r]^2} < \frac{1}{E[Y_r]} = \frac{2^r}{F_0} < \frac{1}{c}$$

By a union bound we therefore get that

$$Pr[2^{\text{MAX-R}} \notin [F_0/c, c \cdot F_0]] \leq 2/c$$

which proves the claim. □

2 Lopsided Set Disjointness

We will now prove communication complexity lower bounds for the Lopsided Set Disjointness problem, which turns out to have many applications to data structure problems. The bound originally was proved in [MNSW98]. Here we follow the presentation of Pătraşcu given in his blog post `http://infoweekly.blogspot.com/2008/05/being-rich-i_23.html`.

The problem is similar to Disjointness: Alice is given an input $x \in \{0,1\}^u$, Bob is given an input $y \in \{0,1\}^u$, and the goal is for the players to determine whether there exists an i such that $x_i = y_i = 1$. However, we also include the additional guarantee that $|x|_w = n$ and $|y|_w = u/2$, with $n \ll u$. (Here $|x|_w$ denotes the hamming weight of a binary string—the number of ones that appear in the string.)

Let a be the number of bits that Alice sends during a protocol for Lopsided Set Disjointness, and b be the number of bits that Bob sends. Trivially the players can solve the problem if Alice sends Bob $a = \log \binom{u}{n} = O(n \log(u/n))$ bits specifying her input or if Bob sends Alice $b = u$ bits specifying his input. Therefore in some sense a bit sent by Alice carries more information, and it makes sense to measure the cost of a protocol in terms of a and b separately and to study the tradeoffs between these two quantities.

For instance, we can show an upper bound on the size of b in terms of the size of a as follows: Let k be a parameter and imagine the players divide their inputs into k equal blocks of size u/k. Alice first sends Bob the name of every one of her blocks that contains at least one 1. This takes $\log(\binom{k}{n})$ bits. Next Bob sends the values of his input for every one of these blocks, which takes $\frac{u}{k} \cdot n$ bits. Optimizing over the parameter k, we get that

$$b \leq \frac{u}{2^{a/n}}$$

For our lower bound we will show that this tradeoff is essentially optimal: We are able to show that

$$b \geq \frac{u}{2^{O(a/n)}}.$$

2.1 Proof

Call a communication matrix $[U, V]$ rich if there are V columns each with at least U 1's in them. We consider rows labeled by inputs to Alice and columns labeled by inputs to Bob.

Claim 5. *Let M be a $[U, V]$ rich communication matrix and suppose there exists a correct (a, b) protocol for the communication problem. Then there exists a $[\frac{U}{2^a}, \frac{V}{2^{a+b}}]$ all one submatrix in M.*

Proof. The proof is by induction on a and b. Originally by assumption we have that M itself is a $[U, V]$ rich rectangle.

Now suppose the protocol has entered a submatrix R of M which is $[U^{'}, V^{'}]$ rich, and that Bob sends a bit to Alice. This splits R into two subrectangles $R^{'}$ and $R^{''}$, one of which contains at least half of the columns from R. Therefore either $R^{'}$ or $R^{''}$ will be $[U, V/2]$ rich.

Now consider the case that Alice sends a bit to Bob at this step. This will again split R into two subrectangles $R^{'}$ and $R^{''}$. In this case there are $V^{'}$ columns that contain at least $U^{'}/2$ 1's in them, half of which have to be in either $R^{'}$ or $R^{''}$, so either $R^{'}$ or $R^{''}$ will be $[U^{'}/2, V^{'}/2]$ rich.

Therefore at the end of the (a, b) protocol there will exist a $[\frac{U}{2^a}, \frac{V}{2^{a+b}}]$ rich rectangle. Because this rectangle has nonzero richness and therefore has at least one 1, and because all rectangles at the end of the protocol must be monochromatic, this implies the conclusion of the claim. □

Now a particular column of interest in the communication matrix M for Lopsided Set Disjointness is a string y with $u/2$ 1's in it that represents a possible input of Bob. For each such string, there are $\binom{u/2}{n}$ possible inputs of Alice that have n 1's and are disjoint from y. Therefore M is $[\binom{u/2}{n}, \binom{u}{u/2}]$ rich. By the claim, if there is an (a, b) protocol to solve Lopsided Set Disjointness it must produce a monochromatic 1 rectangle of size

$$\frac{\binom{u/2}{n}}{2^a} \times \frac{\binom{u}{u/2}}{2^{a+b}}$$

Suppose this monochromatic 1 rectangle is defined by $\{x_1, x_2, \ldots\} \times \{y_1, y_2, \ldots\}$. Because the rectangle is monochromatic 1, every x_i is disjoint from every y_j. Therefore if we define $X = \cup_{x_i}$ and $Y = \cup_{y_i}$, (where $x_i \cup x_j$ is the string that has a 1 in every bit position for which either x_i or x_j has a 1), then

$$|X|_w + |Y|_w \leq u \tag{1}$$

Also, because every row has $n1$'s from X and every column has $u/2$ 1's from Y, we know that the size of the rectangle is at most

$$\binom{|X|_w}{n} \times \binom{|Y|_w}{u/2}$$

From this we get that

$$\binom{|X|_w}{n} \geq \frac{\binom{u/2}{n}}{2^a} \tag{2}$$

$$\binom{|Y|_w}{u/2} \geq \frac{\binom{u}{u/2}}{2^{a+b}} \tag{3}$$

Next we use the following binomial inequality: when $B \leq A$ then

$$\frac{\binom{A}{C}}{\binom{B}{C}} \geq (A/B)^C$$

Using the inequality, we get from (2) that

$$\left(\frac{u}{2|X|_w}\right)^n < 2^a$$

And thus

$$|X|_w > \frac{u}{2^{O(a/n)}}$$

From (1) we have that

$$|Y|_w \leq u - |X|_w \leq u(1 - 2^{-O(a/n)})$$

Finally, using the binomial inequality on (3) we get that

$$\begin{aligned}
\left(\frac{u}{|Y|_w}\right)^{u/2} &\leq 2^{a+b} \\
\left(1 + 2^{-O(a/n)}\right)^{\Theta(u)} &\leq 2^{a+b} \\
e^{u/2^{O(a/n)}} &\leq 2^{a+b} \\
b &\geq \frac{u}{2^{O(a/n)}}
\end{aligned}$$

Here the second line uses the inequality $1/(1-\epsilon) > 1+\epsilon$ and the third line uses the estimate $(1+1/A)^B = e^{\Theta(B/A)}$.

References

[AMS99] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[BYJKS04] Z. Bar-Yossef, T. Jayram, R. Kumar, and D. Sivakumar. Information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.

[CKS03] A. Chakrabarti, S. Khot, and X. Sun. Near-optimal lower bounds on the multi-party communication complexity of set-disjointness. In *Proceedings of the 18th IEEE Conference on Computational Complexity*. IEEE, 2003.

[Gro09] A. Gronemeier. Asymptotically optimal lower bounds on the NIH multiparty information compleixty of the AND function and disjointness. In *Proceedings of the 26th Symposium on Theoretical Aspects of Computer Science*, pages 505–516, 2009.

[MNSW98] P. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.