5. Applications to streaming

Lecturer: Prahladh Harsha *Scribe: Girish Varma*

In this lecture, we will see applications of communication complexity to proving lower bounds for streaming algorithms. Towards the end of the lecture, we will introduce combinatorial auctions, and we will see applications of communication complexity to auctions in the next lecture. The references for this lecture include Lecture 7 of Troy Lee's course on communication complexity [Lee10], Lecture 9 of Piotr Indyk's course on streaming [Ind07] and the chapter on combinatorial auctions by Blumrosen and Nisan [BN07] in the book Algorithmic Game Theory.

5.1 Streaming Algorithms

Streaming algorithms are algorithms that work on input data having very large size. The input data is so large that the algorithm can make only one or a few passes over the input, while using very small space. For this reason, the input data is often referred to as a data steam. If the input is $y_1 y_2 \dots y_n$ where each $y_i \in [m]$, an "ideal" streaming algorithm will use only $O(\text{poly}(\log m, \log n))$ space (i.e, just polynomially larger space than required to store the address or identity of any element in the stream). Such restricted algorithms are needed for example in a router to collect some statistics about the data packets that flow through it.

A theoretical study of streaming algorithms was initiated by the seminal paper of Alon, Matias and Szegedy [AMS99]. In this paper, they gave streaming algorithms for computing frequency moments of data streams and also proved nearly matching lower bounds for the problem.

Definition 5.1 (Frequency Moments)**.** *Given a data stream $y_1, y_2, \dots, y_n$ where each $y_i \in [m]$, the* frequency *of $i \in [m]$ in the stream is $x_i = |\{j \mid y_j = i\}|$. The vector $x = (x_1, x_2, \dots, x_m)$ is called the* frequency vector*. The p^{th}* frequency moment *of the input is defined as follows:*

$$F_p = \begin{cases} |\{i \mid x_i \neq 0\}| & \text{if } p = 0 \\ \max_i x_i & \text{if } p = \infty \\ \|x\|_p^p = \sum x_i^p & \text{otherwise.} \end{cases}$$

F_0 is the number of distinct elements in the stream while F_1 is the number of elements (with repetition). Clearly, $F_1 = n$ and can be computed using $O(\log n)$ space. It is easy to see that F_0 can be computed exactly using m bits of space. In fact, all the moments can be computed using $m \log n$ space by keeping track of the frequency vector x. Using tools such as pairwise independence, Alon, Matias and Szegedy showed that F_0 can be approximated by a randomized streaming algorithm that uses at most $O(\log m, \log n)$ space (c.f., [AMS99, Proposition 2.3]). Alon, Matias, Szegedy [AMS99] (with improvements due

to Saks and Sun [SS02]; Bar-Yossef, Jayram, Kumar and Sivakumar [BJKS04]; and Indyk and Woodruff [IW05]) proved the following theorem about how well small-space algorithms can approximate the frequency moments.

Theorem 5.2.

1. *For $p \in [0,2]$, there is a randomized streaming algorithm that $(1+\varepsilon)$-approximates F_p in space $O(poly(\log m, \log n))$.*
2. *For $p > 2$, any randomized streaming algorithm that $(1+\varepsilon)$-approximates F_p requires $\Omega(m^{1-2/p})$ space.*[1]
3. *For $p > 2$, there is a randomized streaming algorithm that $(1+\varepsilon)$-approximates F_p in space $O(m^{1-2/p})$.*[2]

All of the lower bounds mentioned in the above theorem are proved via communication complexity, which we illustrate in this lecture. To begin with, we will show that any algorithm that computes F_∞ exactly requires linear space by reducing the problem to disjointness.

Theorem 5.3. *Any randomized streaming algorithm that computes F_∞ requires $\Omega(m)$ space.*

Proof. Using a streaming algorithm for computing F_∞ using space C, we will come up with a protocol for disjointness with communication complexity C. Since we know that any randomized protocol for computing disjointness requires $\Omega(m)$ space, the theorem follows.

Suppose Alice, Bob are given inputs $X = \{a_1, a_2, \ldots a_k\}, Y = \{b_1, b_2, \ldots b_{k'}\} \subseteq [m]$ respectively. Alice and Bob will then simulate the streaming algorithm on the data stream $a_1 a_2 \cdots a_k b_1 b_2 \cdots b_{k'}$. Note Alice has only the first half of the stream, while Bob has only the second half. Alice runs the streaming algorithm on her part $(a_1 a_2 \cdots a_k)$ and passes the snapshot of the memory used at this point to Bob, which is of size at most C. Bob will continue running the algorithm till the end of the input. Observe that F_∞ for this stream is either 1 or 2 depending on whether the sets are disjoint or not. Thus, Alice and Bob can determine if the sets are disjoint using the output of the algorithm. □

A couple of remarks on the proof.

Remark 5.4.

- Multi-pass streaming algorithms: *Note that we can do the above reduction even in case of a streaming algorithm that makes multiple (say k) passes on the input. In this case, we get a protocol where at most $2kC$ bits are exchanged. Hence the above proof rules out even multi-pass streaming algorithms that use small space for this problem.*
- Approximation: *The proof shows that for every $\varepsilon > 0$, even $(2-\varepsilon)$-approximating F_∞ is not possible in $o(m)$ space.*

[1] The original paper of Alon, Matias and Szegedy proved a lower bound of $\Omega(m^{1-5/p})$ which was improved to $\Omega(m^{1-2/p})$ by subsequent works by Saks and Sun [SS02] and Bar-Yossef, Jayram, Kumar and Sivakumar [BJKS04].

[2] The original paper of Alon, Matias and Szegedy gave an algorithm that uses $O(m^{1-1/p})$ space which was later improved by Indyk and Woodruff [IW05] to match the lower bound.

In a later lecture, we will strengthen the above inapproximability result to show the following.

Theorem 5.5. *Any streaming algorithm (even randomized) that c-approximates F_∞ requires $\Omega(m/c^2)$ space.*

It is easy to obtain the $\Omega(m^{1-2/p})$-space lower bound for computing F_p using the above theorem.

Corollary 5.6. *Any streaming algorithm that computes F_p requires $\Omega(m^{1-2/p})$ space.*

Proof. The p-norm and ∞-norm satisfy the following inequality

$$\|x\|_\infty \leq \|x\|_p \leq m^{1/p}\|x\|_\infty.$$

So $\|x\|_p$ is an $m^{1/p}$-approximation for $\|x\|_\infty$. The lower bound now follows from Theorem 5.5 (since $F_p = \|x\|_p^p$ and $F_\infty = \|x\|_\infty$). □

5.2 Multi-party Communication and Streaming

In this section, we will use the pretext of attempting to prove Theorem 5.5 to introduce multi-party communication. We won't prove Theorem 5.5 now, but will instead see how multi-party communication complexity lower bounds are useful (for instance towards proving Corollary 5.6).

We extend the communication problem to a multi-party setting. Suppose there are k parties $A_1, A_2, \cdots A_k$, and k inputs $x_1, x_2, \ldots x_k$ and we want to compute some function $f(x_1, x_2, \ldots x_k)$. Two commonly considered models in this setting are as follows.

- With respect to the inputs:
  - **Number in hand (NIH)** : Each A_i knows only x_i.
  - **Number on forehead (NOF)** : Each A_i knows all x_j's except x_i.
- With respect to communication:
  - **Blackboard or Broadcast** : The message of any A_i is seen by all.
  - **Message passing** : Every message from some A_i is addressed to some other party A_j.

We will use the following variant of the disjointness problem.

Definition 5.7 ($\mathsf{UDISJ}_{m,t}$)**.** $\mathsf{UDISJ}_{m,t}$ *is the promise problem of distinguishing between the following inputs*

$$YES = \{(X_1, \ldots X_t) \mid X_i \subseteq [m] \text{ and } \forall u \neq v,\ X_u \cap X_v = \emptyset\}$$
$$NO = \{(X_1, \ldots X_t) \mid X_i \subseteq [m] \text{ and } \exists j \in [m],\ \forall u \neq v,\ X_u \cap X_v = \{j\}\}$$

Later in the course, we will see a proof of the following theorem.

Theorem 5.8 (Gronemeier [Gro09]). *The communication complexity of* $\mathsf{UDISJ}_{m,t}$ *under number in hand (NIH) model and broadcast communication is* $\Omega(m/t)$.

Theorem 5.8 will be proved later in this course. For now, let us see how it implies Corollary 5.6.

Proof of Corollary 5.6. The reduction is similar to that in the proof of Theorem 5.3, applying Theorem 5.8 with $t = m^{1/p}$.

Suppose the input to party i is the set $A_i = \{a_1^i, a_2^i, \cdots a_{k_i}^i\}$. Without loss of generality, assume that the union of the t sets is exactly m. Given a streaming algorithm for computing F_p, we will execute a multi-party NIH broadcast protocol by simulating the streaming algorithm on the the input

$$a_1^1, a_2^1, \cdots, a_{k_1}^1, \quad a_1^2, a_2^2, \cdots, a_{k_2}^2, \quad \cdots, \quad a_1^t, a_2^t, \cdots, a_{k_t}^t.$$

This simulation is performed by each party (in sequence) simulating the streaming algorithm on the part of the data stream it possesses. On completion of its input, the party broadcasts the state of the streaming algorithm so that the next party can proceed with the simulation. Observe that the total broadcast is $t \cdot S$ where S is the space of the streaming algorithm. If the input is a NO instance of $\mathsf{UDISJ}_{m,t}$, then F_p of the stream is $t^p + (m-1)$. On the other hand, if it is a YES instance, then F_p is exactly m. Thus, if $t^p > 1$, then knowing F_p distinguishes YES and NO instances, and so, by Theorem 5.8, the total broadcast should be $\Omega(m/t)$. Hence for $t = m^{1/p}$, it follows that $m^{1/p} \cdot S = \Omega(m/m^{1/p})$, yielding the lower bound of $S = \Omega(m^{1-2/p})$. □

5.3 Combinatorial Auctions

Consider an auction with m indivisible items and n bidders. Each bidder i, has a private valuation for any subset of the items ($v_i : 2^{[m]} \to \mathbb{R}^+$), satisfying monotonicity property (i.e. $S \subseteq T \Rightarrow v_i(S) \leq v_i(T)$ and $v_i(\emptyset) = 0$). The auction is supposed to give an allocation ($S_1, S_2, \ldots S_n$, S_i's disjoint) of sets of items to bidders and a corresponding set of proces $(P_1, \ldots, P_n)$.. There can be various goals for an allocation mechanism.

- Society's point of view: Maximize social welfare of the allocation (i.e. $\sum_{i \in n} v_i(S_i)$).
- Bidder's point of view: Maximize utility (i.e. $\sum_{i \in [n]} v_i(S_i) - P_i$).
- Auctioneer's point of view: Maximize revenue (i.e. $\sum_{i \in [n]} P_i$)

Observe that even expressing the input requires exponential space (the valuation function requires 2^m space). A simple case is the single-minded bidder: a bidder is interested in a particular bundle S and nothing else. Hence, the valuation for a single minded bidder can be easily expressed by a tuple (S^*, v^*) and his valuation function v is then defined as follows.

$$v(S) = \begin{cases} v^* & \text{if } S^* \subseteq S \\ 0 & \text{otherwise.} \end{cases}$$

Even for the simple case of single-minded bidders, it can be shown that maximizing social welfare is NP-hard. Thus, the task of allocation is an intractable problem. One may then ask if the problem is intractable due to computational reasons or is it hard for even more fundamental reasons. We will show (not surprisingly, using communication complexity) that even if the auctioneer and the bidders are allowed to do unbounded computation, they will need to communicate exponentially many messages if they have to maximize social welfare.

References

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. *The space complexity of approximating the frequency moments.* J. Computer and System Sciences, 58(1):137–147, 1999. (Preliminary Version in *28th STOC*, 1996). doi:10.1006/jcss.1997.1545.

[BJKS04] Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. *An information statistics approach to data stream and communication complexity.* J. Computer and System Sciences, 68(4):702–732, June 2004. (Preliminary Version in *43rd FOCS*, 2002). doi:10.1016/j.jcss.2003.11.006.

[BN07] Liad Blumrosen and Noam Nisan. *Combinatorial auctions*. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, eds., *Algorithmic Game Theory*, chapter 11, pages 267–300. Cambridge University Press, 2007.

[Gro09] Andre Gronemeier. *Asymptotically optimal lower bounds on the NIH-multi-party information complexity of the AND-function and disjointness.* In Susanne Albers and Jean-Yves Marion, eds., *Proc. 26th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009. doi:10.4230/LIPIcs.STACS.2009.1846.

[Ind07] Piotr Indyk. *6.895: Sketching, Streaming and Sub-linear space algorithms*, 2007. A course offered at MIT (Fall 2007).

[IW05] Piotr Indyk and David P. Woodruff. *Optimal approximations of the frequency moments of data streams.* In *Proc. 37th ACM Symp. on Theory of Computing (STOC)*, pages 202–208. 2005. doi:10.1145/1060590.1060621.

[Lee10] Troy Lee. *16:198:671 Communication Complexity*, 2010. A course offered at Rutgers University (Spring 2010).

[SS02] Michael E. Saks and Xiaodong Sun. *Space lower bounds for distance approximation in the data stream model.* In *Proc. 34th ACM Symp. on Theory of Computing (STOC)*, pages 360–369. 2002. doi:10.1145/509907.509963.