Lecture 11

Last time

- Limitations of streaming algorithms
- Communication complexity

Today

- Graph streaming
- Linear sketching for graph connectivity
- L_0 sampling

Graph Streams

- Consider a stream of edges $\langle e_1, \ldots, e_m \rangle$
  defining a graph G with $V = [n]$ and $E = \{e_1, \ldots, e_m\}$

- Semi-streaming: space restriction of $O(n \operatorname{polylog} n)$ bits

- What can we compute about G in this model?

Connected Components

Goal: Compute the number of connected components

Spanning Forest Algorithm

1. Initialize a union-find data structure with singletons for all vertices to represent a forest F on $[n]$ with no edges.
2. For each edge (u, v), if u and v are in different sets in F, merge their sets.
3. Return the number of sets in F.

Analysis:

- In the final forest, each set (tree) corresponds to a connected component
- Space: $O(n \log n)$ bits

Based on Andrew McGregor's slides: https://people.cs.umass.edu/~mcgregor/711S18/graph-1.pdf

Dynamic Graph Streams

- Edges can be added and deleted
- Each stream update specifies an edge e and whether it is added or deleted
- Can we still compute connected components?

Graph Sketching: Motivating Example

- There are n people in a social network
- Each has the corresponding row of the adjacency matrix of the network
- Each can write a postcard to Mark Zukerberg
- How many bits should each postcard contain, so that he can determine whether the network is connected w.h.p.?

Today: $O(\text{polylog}\, n)$ bits suffice

Corollary: $O(n\, \text{polylog}\, n)$ bits suffice to compute whether a dynamic stream of edges corresponds to a connected graph

[Ahn Guha McGregor 12]

First Ingredient: Borůvka's Algorithm

Consider a different (non-streaming) algorithm for computing a spanning forest

Spanning Forest Algorithm 2 (Borůvka's Algorithm)

1. Initially put each node in its own component.
2. Repeat until no more changes are made:
3. For each connected component, pick an incident edge (if one exists).
4. Merge all components connected by the selected edges.

Analysis:

- There are at most $\log n$ rounds since, in round $i = 1,2, \ldots$, every connected component either grows to size at least 2^i or stops growing.
- The set of selected edges includes a spanning forest of the graph.

Based on Andrew McGregor's slides: https://people.cs.umass.edu/~mcgregor/711S18/graph-2.pdf

Second Ingredient: Sketch for L_0 Sampling

Problem: Given a stream of elements from $[N]$ with insertions and deletions, output an element with non-zero (positive or negative) frequency (w.h.p.).

More general L_p Sampling:

If the final frequency vector is f, return an index $I \in [N]$ and $R \in \mathbb{R}$ with

$$\Pr[I = i] = \pm\varepsilon \frac{|f_i|^p}{||f_i||_p^p} + N^{-c} \text{ and } R = (1 \pm \varepsilon) f_i \quad \text{(for each } i \in [N]\text{)}$$

L_0 Sketching Lemma

There exists a random matrix $\mathcal{A} \in \mathbb{R}^{O(\log^2 N) \times N}$ such that, for each $x \in \mathbb{R}^N$, with probability at least $1 - \delta$ (for $\delta = 1/\mathrm{poly}(N)$), we can learn (i, x_i) for some $x_i \neq 0$ from $\mathcal{A}x$.

- Union Bound: If we have multiple vectors $x^{(1)}, \ldots, x^{(t)}$, then we can find a non-zero entry from each of them from $\mathcal{A}x^{(1)}, \ldots, \mathcal{A}x^{(t)}$ w. p. $\geq 1 - \delta t$.
- Linearity: Given $\mathcal{A}x$ and $\mathcal{A}y$, we can find a non-zero entry from $z = x + y$, since $\mathcal{A}z = \mathcal{A}(x + y) = \mathcal{A}x + \mathcal{A}y$.

Based on Andrew McGregor's slides: https://people.cs.umass.edu/~mcgregor/711S18/graph-2.pdf

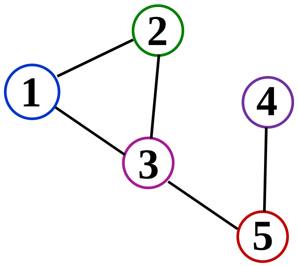
Third Ingredient: Signed Vertex-Edge Vectors

Associate each node $i \in [n]$ with a vector of length $\binom{n}{2}$ indexed by node pairs.

- An entry indexed by a pair $\{i, j\}$ is $\begin{cases} \mathbf{1} & \text{if } \{i,j\} \in E \text{ and } i > i \\ -\mathbf{1} & \text{if } \{i,j\} \in E \text{ and } i < i \\ \mathbf{0} & \text{otherwise} \end{cases}$

Example:

	{1,2}	{1,3}	{1,4}	{1,5}	{2,3}	{2,4}	{2,5}	{3,4}	{3,5}	{4,5}
$x^{(1)} =$	1	1	0	0	0	0	0	0	0	0
$x^{(2)} =$	−1	0	0	0	1	0	0	0	0	0
$x^{(3)} =$	0	−1	0	0	−1	0	0	0	1	0
$x^{(4)} =$	0	0	0	0	0	0	0	0	0	1
$x^{(5)} =$	0	0	0	0	0	0	0	−1	0	−1



Lemma

Non-zero entries of $\sum_{i \in S} x^{(i)}$ correspond to edges between S and V/S.

Proof: An entry of $\sum_{i \in S} x^{(i)}$ indexed by $\{j, k\}$ can be non-zero only if $\{j, k\} \in E$ and it is adjacent to a node in S. But if $j, k \in S$, then this entry is $1 - 1 = 0$. So exactly one of j, k is in S.

Based on Andrew McGregor's slides: https://people.cs.umass.edu/~mcgregor/711S18/graph-2.pdf

What to Write on the Postcard

- Person at node i sends: $\mathcal{A}_1 x^{(i)}, \ldots, \mathcal{A}_{\log n} x^{(i)}$, where $\mathcal{A}_1, \ldots, \mathcal{A}_{\log n}$ are independent random matrices for L_0 sampling
- Mark Zukerberg simulates Borůvka's Algorithm:
  - Identify an incident edge from each node i by finding a non-zero entry of $x^{(i)}$ from $\mathcal{A}_1 x^{(i)}$

    Non-zero entries correspond to incident edges

  - In round t, identify an incident edge from each component S, by finding a non-zero entry of $\sum_{i \in S} x_i$ from

$$\sum_{i \in S} \mathcal{A}_t x^{(i)} = \mathcal{A}_t \sum_{i \in S} x_i$$

Based on Andrew McGregor's slides: https://people.cs.umass.edu/~mcgregor/711S18/graph-2.pdf

L_0 Sketching: Main Idea

- For each $j \in \{0, \dots, \log n\}$, independently sample a 2-wise independent hash function $h_i : [N] \to [2^i]$

Each element j of $[N]$ is added to S_i w.p. 2^{-i}

- Each h_i implicitly defines the set $S_i = \{j \in [N] : h_i(j) = 0\}$

To sketch each vector x, for all $S \in \{S_0, \dots, S_{\log n}\}$, compute

$$a = \sum_{j \in S} j x_j \,;\;\; b = \sum_{j \in S} x_j \,; \text{ estimate } d = (1 \pm 0.1)\left|\left|x_S\right|\right|_0 \text{ with } \delta = n^{-O(1)}$$

Our distinct elements estimator work in streams with deletions, too!

To output the index of a non-zero entry of x

Only one x_j is non-zero

- Select the smallest i^* with S_{i^*} such that $d = 1 \pm 0.1$ (if one exists)

Then $a = j x_j$ and $b = x_j$

- Return a/b

Analysis

Lemma

Let $P = \{i \in [N]: x_i \neq 0\}$ be positions of non-zero entries. For some S,

$$\Pr[|P \cap S| = 1] \geq 1/8$$

Proof: Pick i such that $2^{i-2} \leq |P| \leq 2^{i-1}$. Then $1/4 \leq |P| \cdot 2^{-i} \leq 1/2$

$$\Pr[|P \cap S| = 1] = \sum_{j \in P} \Pr[j \in S_i, k \notin S_i\ \forall k \in P \backslash \{j\}]$$

$$= \sum_{j \in P} \Pr[j \in S_i] \cdot \Pr[k \notin S_i\ \forall k \in P \backslash \{j\} \mid j \in S_i]$$

By Product Rule

$$\geq \sum_{j \in P} \frac{1}{2^i} \cdot \left(1 - \sum_{k \in P \backslash \{j\}} \Pr[\, k \in S_i \mid j \in S_i \,]\right)$$

By a union bound

$$\geq \sum_{j \in P} \frac{1}{2^i} \cdot \left(1 - \sum_{k \in P \backslash \{j\}} \Pr[k \in S_i]\right)$$

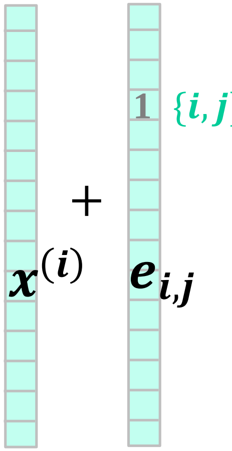
By pairwise independence

$$\geq \frac{|P|}{2^i} \cdot \left(1 - \frac{|P|}{2^i}\right) \geq \frac{1}{4} \cdot \left(1 - \frac{1}{2}\right) = \frac{1}{8}$$

By $1/4 \leq |P| \cdot 2^{-i} \leq 1/2$

From Postcards to Streaming Algorithm

- Space to store each hash function: $O(\log N) = O(\log n)$
- Number of hash functions is polylog(n)
- Each message uses polylog(n) bits
- Total space: n polylog(n)



- To insert an edge $\{i, j\}$, where $i < j$:

$$\mathcal{A}_t x^{(i)} \leftarrow \mathcal{A}_t x^{(i)} + \mathcal{A}_t e_{i,j}$$
$$\mathcal{A}_t x^{(j)} \leftarrow \mathcal{A}_t x^{(j)} - \mathcal{A}_t e_{i,j}$$

where $e_{i,j}$ is the vector of length $\binom{n}{2}$ with exactly one non-zero entry

- To delete an edge $\{i, j\}$, where $i < j$:

$$\mathcal{A}_t x^{(i)} \leftarrow \mathcal{A}_t x^{(i)} - \mathcal{A}_t e_{i,j}$$
$$\mathcal{A}_t x^{(j)} \leftarrow \mathcal{A}_t x^{(j)} + \mathcal{A}_t e_{i,j}$$