COMS E6998-9: Algorithms for Massive Data (Spring'19)	Mar 5, 2019

Lecture 13: Streaming for dynamic graph problems: connectivity

Instructor: *Alex Andoni* Scribes: *Anh Phung*

1 Review

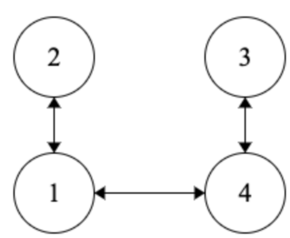
- Dynamic Sampling (tool): we want a linear sketch for $x \in \{-1, 0, 1\}^n$ (under general updates to x) such that the estimator produces (i, x_i), where $i \in \{j : x_j \neq 0\}$ is chosen randomly (over the choice of the sketch and estimator).
- Dynamic Sampling - Full: this algorithm works with probability at least $1 - \frac{1}{n^2}$ and uses $O(\log^4 n)$ space.

2 Streaming for dynamic graphs

- Model: Streaming for dynamic graphs is similar to streaming for graphs, except we can also **delete** existing edges.

  Example 1. *Stream is* $(1, 2); (2, 3); (3, 4); (4, 1); -(2, 3)$.

  The resulting graph is



- Problem: We want to determine connectivity for dynamic graphs.
- Goal: Use only $O(n(\log n)^{O(1)})$ space.

The first naive approach is to use a spanning tree H like before. On deletion, we simply delete the edge if it exists in H. However, this simple approach won't work, as seen in the example above. The graph should be connected, but if we delete edge $(2, 3)$, H will now only have 2 edges left, namely $(1, 2)$ and $(3, 4)$.

We will present an algorithm inspired by Boruvka's algorithm for parallel spanning tree construction.

Definition 2. *For node v, define $x^v \in R^p$, where $p = \binom{n}{2}$ = number of possible edges, as:*

$$x^v_{(u,w)} = \begin{cases} 1 & \text{if } u = v, (u,w) \in E, u < w \\ -1 & \text{if } u = v, (u,w) \in E, u > w \\ 0 & \text{otherwise} \end{cases}$$

For instance, $x^2 = (-1, 0, 0, 1, 0, 0)$ in example 1.

Observation 3. *For all v, keep a DS - full sketch on x^v.*

- Operationally, on seeing new edge (a, b), we update

$$x^a_{(a,b)} += \begin{cases} 1 & \text{if } a < b \\ -1 & \text{if } a > b \end{cases}$$

  $x^b_{(b,a)}$ is updated similarly. On deleting edge (a, b), we update with the reverse sign.

- At the end of the stream, DS - full will sample a **random** edge incident to v. Every node has a random edge sampled, thus there are n sampled edges in total. It's possible that there are repeated edges among them.

Observation 4. *We can collapse these edges (i.e. identifying connected components).*

Fix Q = a connected subset of the graph and define $x^Q = \sum_{v \in Q} x^v$.

Fact 5.

$$x^Q_{(u,v)} = \begin{cases} 1 & \text{if } (u,v) \in E, u \in Q, v \notin Q, u < v \\ -1 & \text{if } (u,v) \in E, u \in Q, v \notin Q, u > v \\ 0 & \text{otherwise} \end{cases}$$

Note that if $u, v \in Q$ and $(u, v) \in E$, at the index of that edge, x^u and x^v will have different sign, thus they cancel when adding up together.

Therefore, if S is the DS - full sketch, then

$$S(x^Q) = S(\sum_{v \in Q} x^v) = \sum_{v \in Q} S(x^v)$$

using linearity of S. This means that if we keep DS - full sketches for each vertex, we can get a DS - full sketch for any connected component Q, i.e. we can sample a random edge incident to Q.

Algorithm. Streaming and keep DS - full sketch $S(x^v)$ for each v. On estimation:

- Keep connected components $Q_1, \ldots, Q_n$.
- For each Q_i, compute $S(Q_i)$ using the equation above.
- Sample edge using $S(Q_i)$.
- Update connected components by combining Q_i and Q_j if there is a sampled edge between them.
- Repeat until there is only 1 connected component left or no more edges.

Claim 6. *The number of iterations is $O(\log n)$.*

Proof. In each iteration, the number of connected component with incident edges goes down by a factor of at least 2. Therefore the number of iterations is $O(\log n)$. □

There is a final caveat with this algorithm so far. The DS - full sketches are not independent between iterations. An easy fix is to keep $k = O(\log n)$ iid DS - full sketches on x^v for each v. Denote them by $S_1(x^v), \ldots, S_k(x^v)$. In iteration l, use sketches $S_l(x^1), \ldots, S_l(x^n)$.

The total space used by the algorithm is $O(n) \times O(\log n) \times O(\log^4 n) = O(n \log^5 n)$.

Recent development.

In 2012, an algorithm is shown for dynamic graphs connectivity streaming.

In 2013, there is a new algorithm for data structure of dynamic graphs connectivity. It maintains a graph G under addition / deletion of edges and can answer queries "are i, j connected in current G?". The time per operation is $(\log n)^{O(1)}$.

3 Projects

- Proposal: 1-2 pages in team of 2 - 4 people. It must include references, problem, main statements, goal (either survey, implementation or research-based).
- Presentation
- Final write-up: 10 pages, excluding references, of original research.