COMS E6998-9: Algorithms for Massive Data (Fall'23) Sep 18, 2023

Lecture 4: Streaming for Dynamic Graphs

Instructor: *Alex Andoni* Scribe: *Sian Lee Kitt*

1 Dynamic sampling tool

Maintain sketch of $x \in \mathbb{Z}^n$ under updates $(i_j, \delta_j) \in [n] \times \mathbb{Z}$ s.t. at the end we can produce a sample (S, X_S) s.t. $Pr[S = i] = \frac{\mathbb{1}[x_i \neq 0]}{\|x\|_0} \pm \frac{1}{n^3}$
Recall Case 2.2: when $\|x\|_0 = 1$ or $\|x\|_0 > 1$
Our solution maintains three quantities
- $\alpha = \sum_i x_i i$
- $\beta = \sum_i x_i$
- $\gamma = \sum_i x_i z^i \mod p$ where p is a prime $> n^4$ and z is random from $\mathbb{Z}_p$

Test: $\gamma = \beta z^{\frac{\alpha}{\beta}} \mod p$, if not this implies $\|x\|_0 > 1$. Otherwise output $(\frac{\alpha}{\beta}, \beta)$

Proof. of correctness.

$$\begin{aligned} \text{Test passes if } \gamma &= \beta z^{\frac{\alpha}{\beta}} \pmod p \\ \Leftrightarrow \sum x_i z^i &= \beta z^{\frac{\alpha}{\beta}} \pmod p \\ x_S z^S &= S z^S \pmod p && \text{if } \|x\|_0 = 1 \text{ and } x_S \neq 0 \end{aligned}$$

$\square$

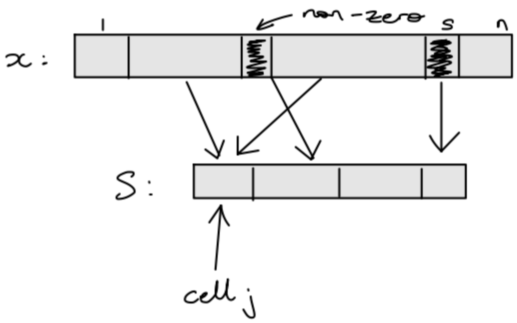
Suppose $\|x\|_0 > 1$ then $p(z) = \sum_{i=1}^n x_i z^i - \beta z^{\frac{\alpha}{\beta}} \pmod p \neq 0$ since there is at least one non-zero term with a power of z.
Failure occurs when $p(z) = 0$ for chosen z.

$$\begin{aligned} Pr_z[p(z) = 0] &\leq \frac{n}{p} && \text{since } p(z) \text{ is of degree } n \text{ and } z \in \mathbb{Z}_p \\ &\leq \frac{1}{n^3} && \text{since } p > n^4 \end{aligned}$$

Case 3

Case 3: Fix k, $\|x\|_0 \leq k$, find all non-zeros of x. Target space will be $O(k \cdot logk)$
Sketch:

- Pick a random hash function $h : [n] \to [2k]$ (maps each coordinate of x into something of size $2k$).



– The figure above displays a linear sketch which throws all coordinates of x into different buckets in S. The number of buckets is $2k$ (or any constant factor of the total number of non-zeros in x, k). In cell j of S, store Case 2.2 sketch on vector $x\Big|_{h=j}$ (interpreted as h(x) $\to j$).

– Fix s s.t $X_s \neq 0$, then the $P[X_s$ is isolated], that is, no collisions with other non-zero coordinates is:

$$
\begin{aligned}
Pr[s \text{ is isolated}] &\geq \frac{\#\text{ of free spaces}}{\#\text{ of total spaces}} \\
Pr[s \text{ is isolated}] &\geq \frac{k+1}{2k} && \text{in worst case } k-1 \text{ cells are occupied in } S \\
&\geq \frac{1}{2}
\end{aligned}
$$

– Hence we succeed in extracting one non-zero coordinate of x with probability $\frac{1}{2}$.

Full sketch

Repeat sketch above $t = 2logk$ times:
Using hash functions: $h_1, \ldots, h_t : [n] \to [2k]$
Store S_i corresponding to h_i, $i = 1 \ldots 2k$
Extraction: go over all cells of $S_1 \ldots S_t$:
    extract the isolated coordinates (if exist) using Case 2.2.

$$
\begin{aligned}
Pr[s \text{ is isolated in at least one } S_i] &= 1 - Pr[s \text{ is not isolated for } i = 1 \dots t] \\
&\geq 1 - (\frac{1}{2})^t \\
&\geq 1 - \frac{1}{4k} \\
Pr[\text{all } k \text{ non-zeros of } x \text{ are extracted}] &\geq 1 - Pr[\exists \text{ one non-zero of } x \text{ which is not isolated}] \\
&\geq 1 - k \cdot \frac{1}{4k} \\
&= \frac{3}{4}
\end{aligned}
$$

Case 4

In Case 3, we extracted all k non-zeros. In case 4 we are back to the situation where we only need to extract at random one non-zero coordinate but now the support of x is much larger. We cannot use the previous solution because of this larger space.
Case 4: $\|x\|_0 = S \in [2^j, 2^{j+1}]$. For example, think of 2^j as $\sqrt{n}$.
Look at a subset of coordinates I_j s.t. $\|x\Big|_{I_j}\|_0 \approx 1$.
I_j = random subset of $[n]$ where $Pr[i \in I_j] = 2^{-j}$.
Choose a random hash function $h : [n] \to \{0, 1, \dots, 2^j - 1\}$ and define $I_j = \{i : h(i) = 0\}$, where the $\mathbb{E}[|I_j|] = n \cdot 2^{-j}$.
We hope to sample exactly one.

$$
\begin{aligned}
Pr[\|x\Big|_{I_j}\|_0 = 1] &= \sum_{i=1}^{S} P(\text{particular coord is included}) \cdot P(\text{none of the other coords are included}) \\
&= \sum_{i=1}^{S} 2^{-j} \cdot (1 - 2^{-j})^{j-1} \\
&\geq S \cdot 2^{-j}(1 - 2^{-j})^S \\
&\approx S \cdot 2^{-j} e^{\frac{-S}{2^j}} \qquad \text{for small enough x: } 1 - x \approx e^{-x} \\
&\geq 2^j \cdot 2^{-j} e^{\frac{-2^{j+1}}{2^j}} \\
&= e^{-2}
\end{aligned}
$$

Just store Case 2.2 for $x\Big|_{I_j}$. There is at least e^{-2} probability in succeeding in producing a random s in supp(x).
Boost success probability to $1 - \frac{1}{n^3}$ by repeating $t = O(logn)$ times.

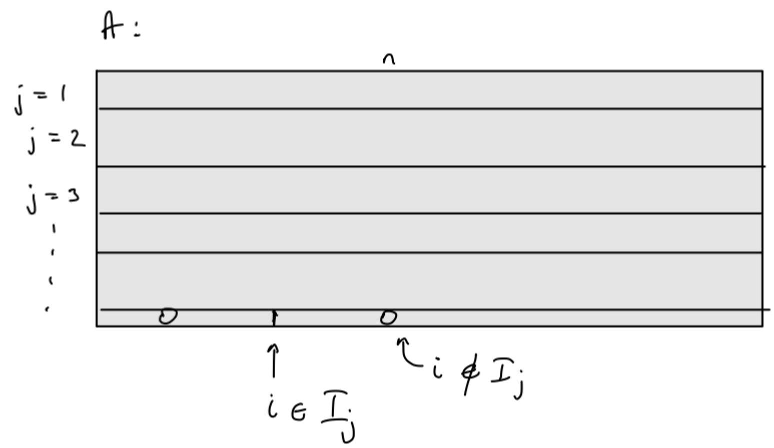
$$
\begin{aligned}
Pr[\text{succeed in producing a random } s \in \text{supp}(x) \text{ in at least one of the } t \text{ trials }] &= 1 - Pr[\text{we fail in all } t \text{ trials }] \\
&\geq 1 - (1 - e^{-2})^t \\
&\geq 1 - \frac{1}{n^3}.
\end{aligned}
$$

Case 5: Arbitrary x

- For each $j = 0 \ldots logn$, prepare Case 4
- To generate a random $s \in \text{supp}(x)$:

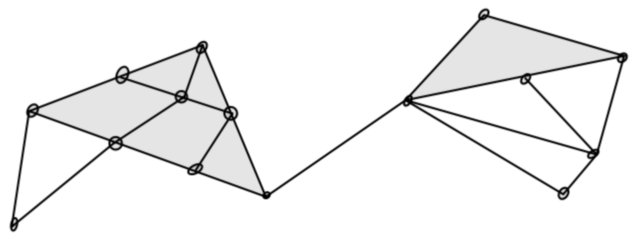
  just iterate $j = 0 \ldots logn$, and report first (s, x_s), $x_j \neq 0$, found.

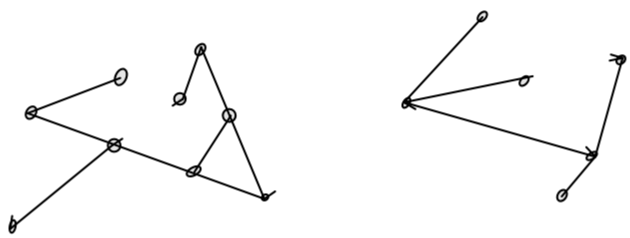
Correctness: Let j be the unique j s.t. $2^j \leq \|x\|_0 \leq 2^{j+1}$.
Space: $O(log^2 n)$ words since we needed $O(logn)$ for $logn$ iterations of $j \cdot$ $O(logn)$ for building Case 4 for each iteration $\cdot O(1)$ for space for Case 2.2.
It is important to note that all of these sketches are linear taking the form $= A \cdot x$.



2 Dynamic Connectivity

Setting: dynamic graph in stream (stream contains insertions and deletions of edges which are correct. For example, we never delete an edge which doesn't exist).
Problem: connectivity or spanning forest in $O(n(logn)^{O(1)})$ space, where n is number of vertices.
Suppose there is a graph where someone deletes multiple edges:





How do we determine the spanning forest and whether the graph is connected or not?

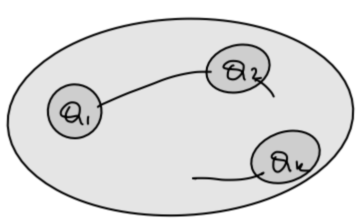
Algorithm 1 Boruka's Algorithm (offline (no streaming) algorithm for spanning forest)

Keep connected components, starting with each vertex as its own connected component
For t = $O(logn)$ times:
    for each connected component $Q \subseteq [n]$, pick an edge crossing Q
    compute new connected components with chosen edges
    repeat on new connected components

Correctness: Connected component (CC)
$CC_j := CCs$ after j steps
$CC =$ total $\#CC$s

Claim 1. $(CC_{j+1} - CC) \leq \frac{1}{2}(CC_j - CC)$

Proof. Consider a final connected component. Each CC at time j, Q_j, is paired with another CC. Since they are all paired, the number of CCs will drop by a factor of 2.

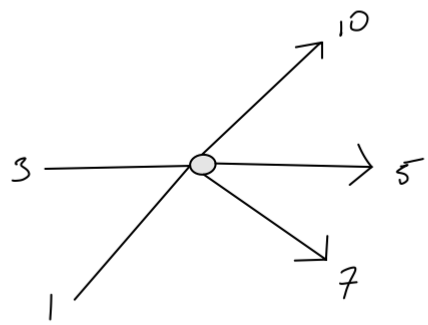


□

We need to simulate this algorithm in the streaming model.
Dynamic Connectivity: $\forall$ vertex v, we define vector $X_v \in \mathbb{R}^{\binom{n}{2}}$ (a node-edge incidence vector).

- $X_v(v, j) = +1$ if $(v, j) \in E, v < j$ where E is the set of edges
- $X_v(j, v) = -1$ if $(j, v) \in E, j < v$
- $X_v = 0$ otherwise

Though the graph is undirected, we can orient the edges for the sake of the vectors. For example, (arrows indicate going towards vertices with larger numbers)



$$\begin{array}{cccc} & (1,v) & (6,v) & (v,5) \\ X_v = & [-1 & -1 & +1] \end{array}$$

Dynamic connectivity basic sketch:

- $\forall$ vertices $v \in [n]$, just keep a Dynamic Sampling sketch for x_v

Space: $O(nlog^2 n)$.

Claim 2. *$\forall$ $Q \in [n], \sum_{v \in Q} x_v$ is a vertex-edge incidence vector on Q.*