Pan-Private Graph/Geometric Streaming Algorithms

Cody Freitag[1]

DIMACS REU

Summer 2016

[1]Supported by the NSF

Overview

Review [1]

Streaming Algorithms:

- Data updates come once, are processed, and disappear
- Want to approximately answer queries minimizing storage, processing time, and response time

Pan-privacy: [DNPRY '05]

- Two data streams are neighbors iff they only differ in a single entry.
- A streaming algorithm is *pan-private* iff for any two neighboring data streams it takes as input, a computationally unbounded intruder can't distinguish the internal state or outputs for each stream.

[1] See Intro Presentation for more details!

Pan-Privacy

- Let $\mathcal{A}\colon \mathcal{S} \to \mathcal{I} \times \mathcal{O}$ be a randomized streaming algorithm
  - $\mathcal{S}$ is the set of all possible data streams
  - $\mathcal{I}$ is the set of all possible internal states
  - $\mathcal{O}$ is the set of all possible outputs
- $\mathcal{A}$ is ϵ-*pan-private* iff for any two neighboring data streams $S \sim S' \in \mathcal{S}$ and for all $I \subseteq \mathcal{I}$, $O \subseteq \mathcal{O}$,

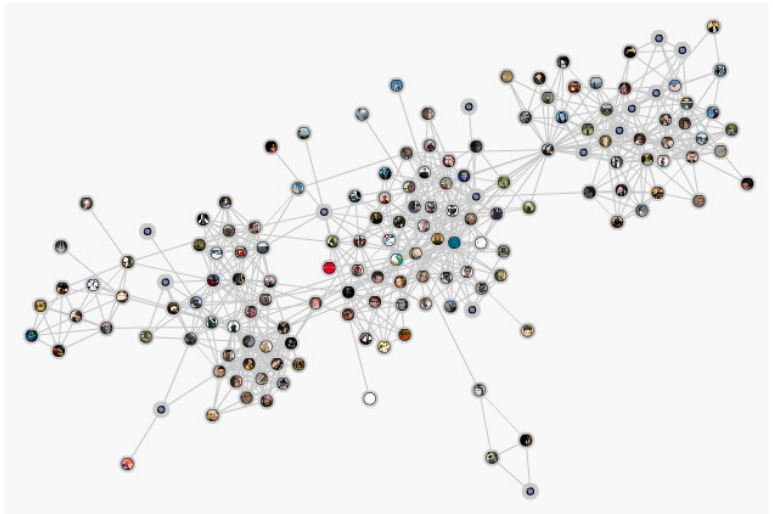
$$\Pr[\mathcal{A}(S) \in (I, O)] \leq e^{\epsilon} \Pr[\mathcal{A}(S') \in (I, O)]$$

Utility

- Let OPT_S be the true value of a query, $\mathcal{Q}$, on a data stream S.
- $\mathcal{A}$ is an (α, β)-approximation algorithm iff for all $S \in \mathcal{S}$ with probability $1 - \beta$,

$$(1 - \alpha)\mathcal{Q}(\mathcal{A}(S)) \leq \mathsf{OPT}_S \leq (1 + \alpha)\mathcal{Q}(\mathcal{A}(S))$$

Graph Density



- Data: a vector representation of an adjacency matrix
- Updates: (e, add) or (e, delete)
  - Assume simple graph
- Density Query: What fraction of edges are present?

Graph Density – Algorithm

Preliminaries:

$$D_0(\epsilon) = \begin{cases} 1 & \text{w.p. } \frac{1}{2} \\ 0 & \text{w.p. } \frac{1}{2} \end{cases}$$

$$D_1(\epsilon) = \begin{cases} 1 & \text{w.p. } \frac{1}{2} + \frac{\epsilon}{4} \\ 0 & \text{w.p. } \frac{1}{2} - \frac{\epsilon}{4} \end{cases}$$

Lemma

For $0 \leq \epsilon \leq 1/2$, $x \in \{0, 1\}$,

$$\Pr_{b \sim D_x}[b] \leq e^{\epsilon} \cdot \Pr_{b \sim D_{1-x}}[b]$$

Graph Density – Algorithm

- Initialization
  - Pick $M \subseteq E$ of $m = \text{poly}(1/\epsilon, 1/\alpha, \log(1/\beta))$ random edges.
  - Create a table where each $e \in M$ has a single bit, b_e.
  - Initially, draw $b_e \sim D_0(\epsilon)$.
- Processing
  - On update $(e, ?)$, if $e \notin M$, do nothing.
  - Otherwise if $? = \text{add}$, redraw $b_e \sim D_1(\epsilon)$.
  - And if $? = \text{delete}$, redraw $b_e \sim D_0(\epsilon)$.
- Output
  - Compute $\theta =$ fraction of 1's in table.
  - Output estimate $\hat{f} = 4(\theta - 1/2)/\epsilon + Lap(1/(\epsilon \cdot m))$

Graph Density – Privacy Analysis

Internal State:

- Let e be edge streams S and S' differ on.
- If $e \notin M$, perfect privacy.
- Otherwise, previous lemma tells us the internal state is pan-private.

Output:

- Adding $Lap(1/(\epsilon \cdot m))$ to output guarantees privacy of output.
- See differential privacy literature for why this works!

Graph Density – Utility Analysis

- Let f be the true fraction of 1's in the table. Then,

$$\begin{aligned} \mathrm{E}[\theta] &= f \cdot (\frac{1}{2} + \frac{\epsilon}{4}) + (1 - f) \cdot (\frac{1}{2}) \\ &= \frac{1}{2} + \frac{f\epsilon}{4} \\ \Rightarrow f \approx \hat{f} &= \frac{4}{\epsilon} \cdot (\theta - \frac{1}{2}) \end{aligned}$$

- Since $m \geq \mathrm{poly}(1/\alpha, \log(1/\beta))$, we get with probability $1 - \beta$,

$$\hat{f} - \alpha \leq f \leq \hat{f} + \alpha$$

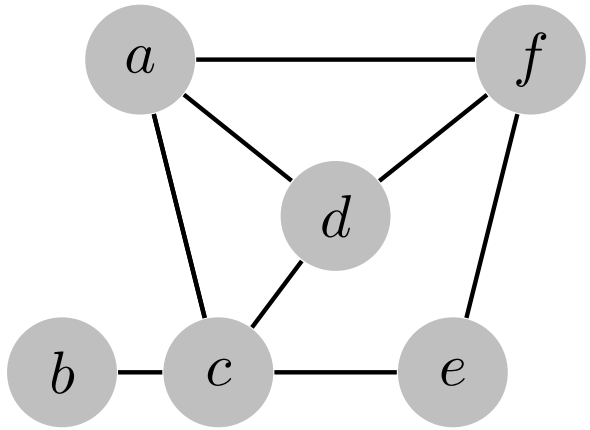
  by a Chernoff bound.
- Can get multiplicative error using a hashing technique on $O(\log(n))$ instances!

Graph Density – Theorem

Theorem

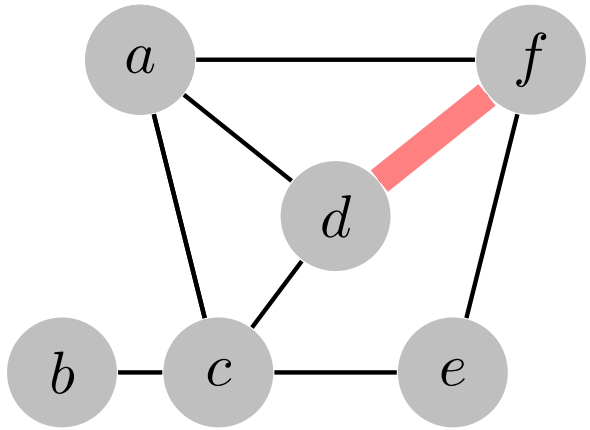
For $0 \leq \epsilon \leq 1/2$, there exists an ϵ-pan-private algorithm that with probability $1 - \beta$ gives an α-multiplicative approximation for graph density. Furthermore, it uses $\mathrm{poly}(1/\epsilon, 1/\alpha, \log(1/\beta), \log(n))$ space, processing time, and output time.

Unfortunate (lack of) Other Results



- Triangle Counting: How many triangles are in the graph?
- Can achieve additive error using similar method to previous algorithm... but getting a multiplicative error would require too much space—a factor of $O(n^3/T)$ more.

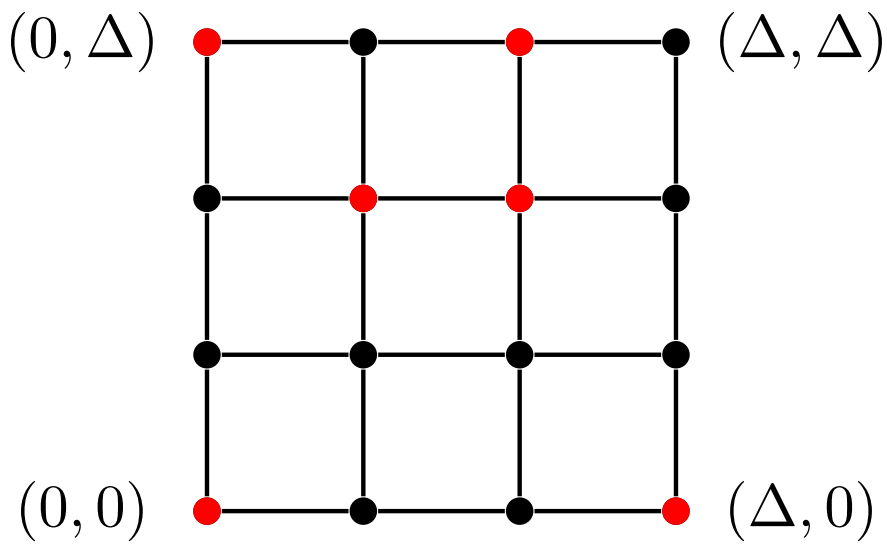
Unfortunate (lack of) Other Results



- Triangle counting and most other non-trivial graph streaming algorithms uniformly sample random vertices or edges to build a better estimator.
- Unfortunately, this isn't possible while maintaining a private internal state.

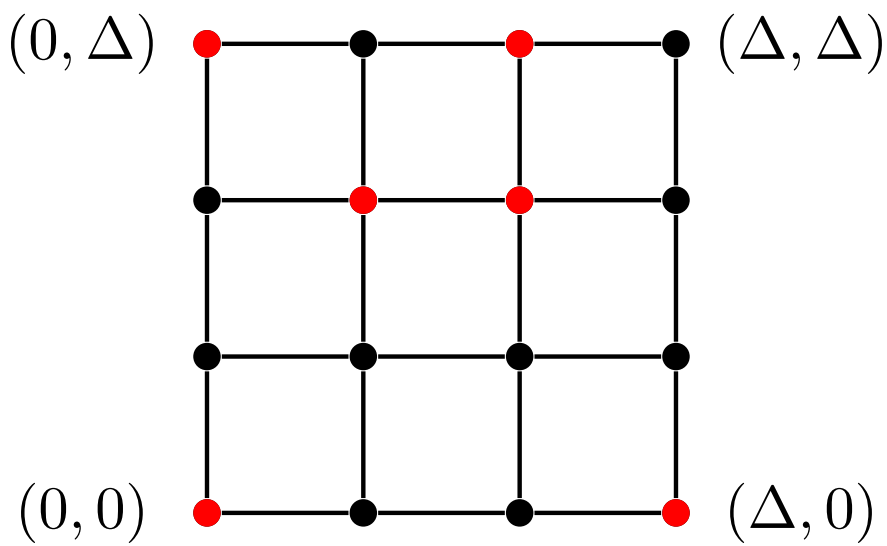
Geometric Model

- Data: a point set $P \subseteq \{0, 1, \ldots, \Delta\}^d$ in a discretized, bounded grid
- Updates: $((x, y), \text{add})$ or $((x, y), \text{remove})$



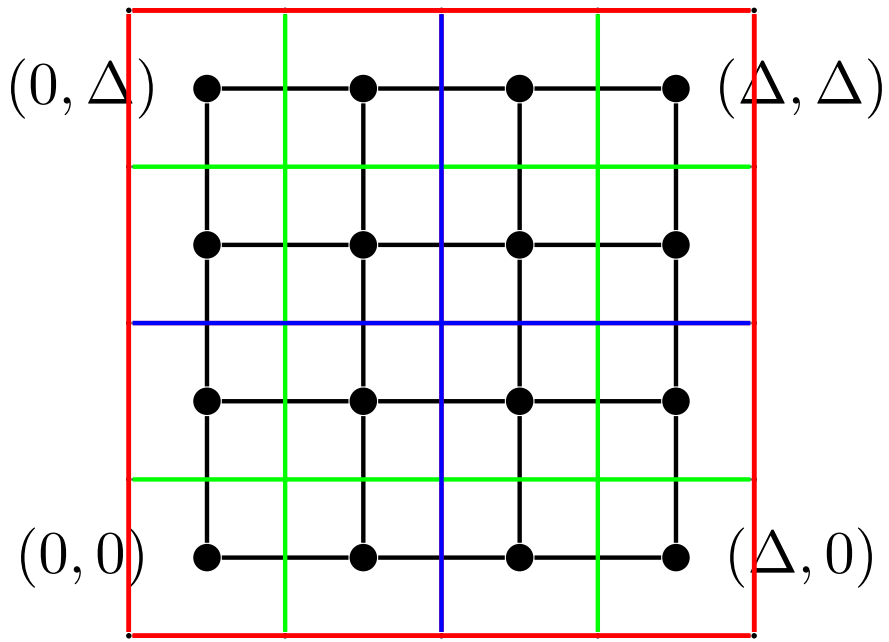
Facility Location

- Let $F \subseteq \{0, 1, \ldots, \Delta\}^d$ be a set of facilities that each cost f to open.
- Pan-privately approximate $\min_F(f \cdot |F| + \sum_{p \in P} d(p, F))$



One Geometric Approach

- Layer randomly shifted grids over the data
  - $G_0, G_1, \ldots, G_\Delta$, each G_i has cells of side length 2^i
- Gather statistics on the cells in different layers



Questions?