Differentially Private Synthetic Graphs Preserving Triangle-Motif Cuts

Author: Pan Peng, Hangyu Xu

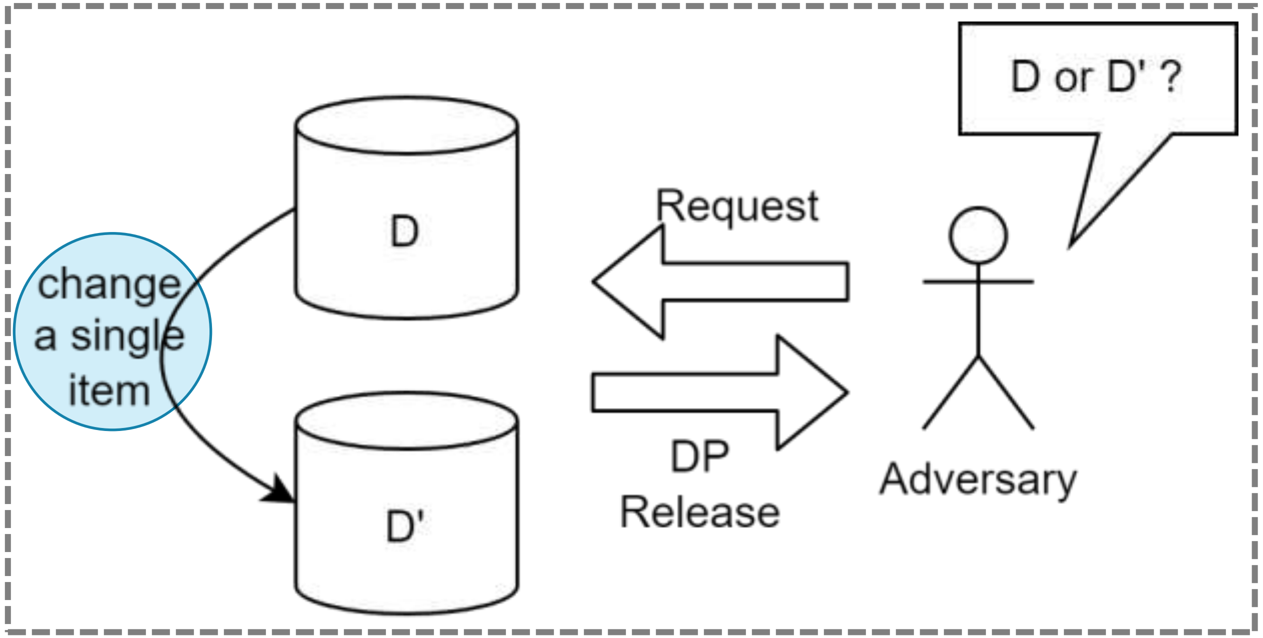
Presenter: Hangyu Xu

University of Science and Technology of China

1 Background

Differential Privacy (DP)

- Requirement: Individual information changes do not have a distinguishable effect on results
- Goal: To ensure utility, aim to minimize error



$$\Pr[\mathcal{M}(x) \in S] \leq \Pr[\mathcal{M}(y) \in S] \cdot e^{\varepsilon} + \delta$$

(ε, δ)-DP

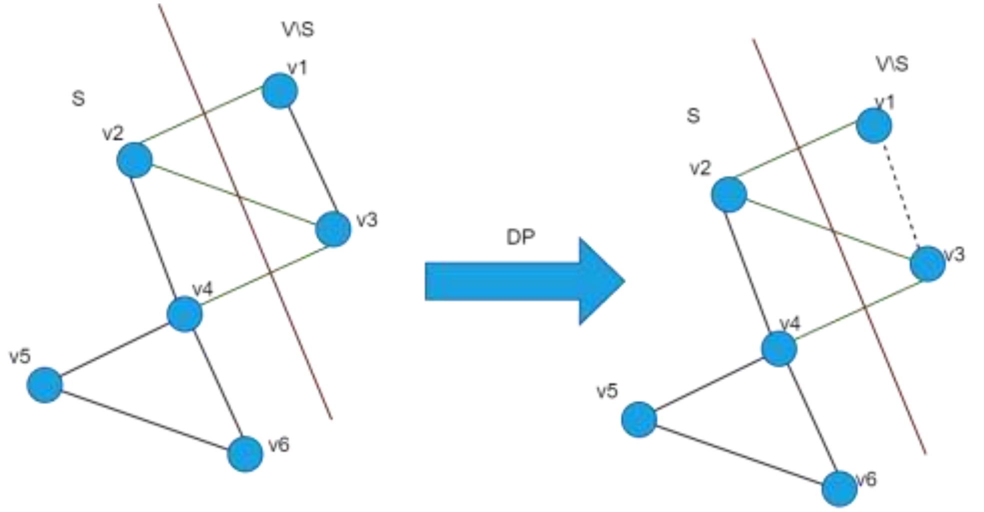
Post-processing

Applying any function to the output of a DP algorithm (without accessing the original data) does not weaken the privacy guarantee.

1 Background

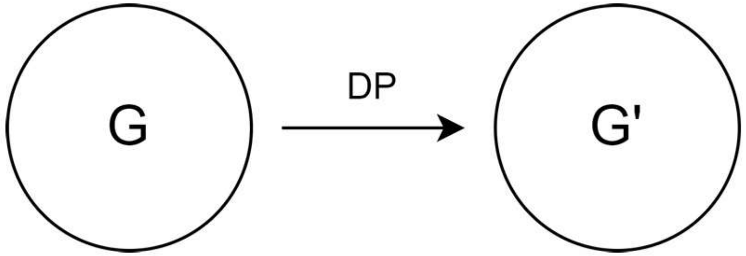
Question: How can we release a synthetic graph under differential privacy while preserving cut structues?

- A lot of work have explored this area[GRU12,EKKL20,LUZ24]
- **Upper Bound**: $\tilde{O}(\sqrt{mn})$ **Lower Bound**: $\Omega(\sqrt{mn})$



differentially privately preserving cut structures

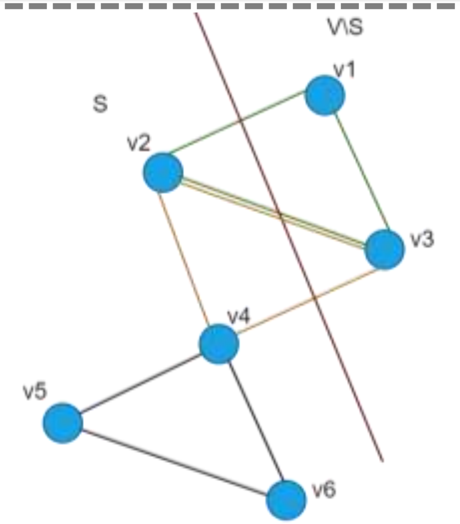
$$\mathrm{Cut}^{(G)}(S, V \setminus S) \approx \mathrm{Cut}^{(G')}(S, V \setminus S)$$



1 Background

- Though powerful, the private synthetic graph for cut structures fails to capture higher-order structure of the graph.
- **Motif**: [MSOI+02]
  - a frequently occurring subgraph within complex networks (e.g. triangles, wedges, cliques,...)
  - Applications: graph clustering, graph data visualization, network analysis
- **Motif Cut**:
  - the sum of weights of the motifs crossing $(S, V \setminus S)$

$$S = \{v_2, v_4, v_5, v_6\}$$
$$\mathrm{Cut}_{\triangle}^{(G)}(S, V \setminus S) = 2$$

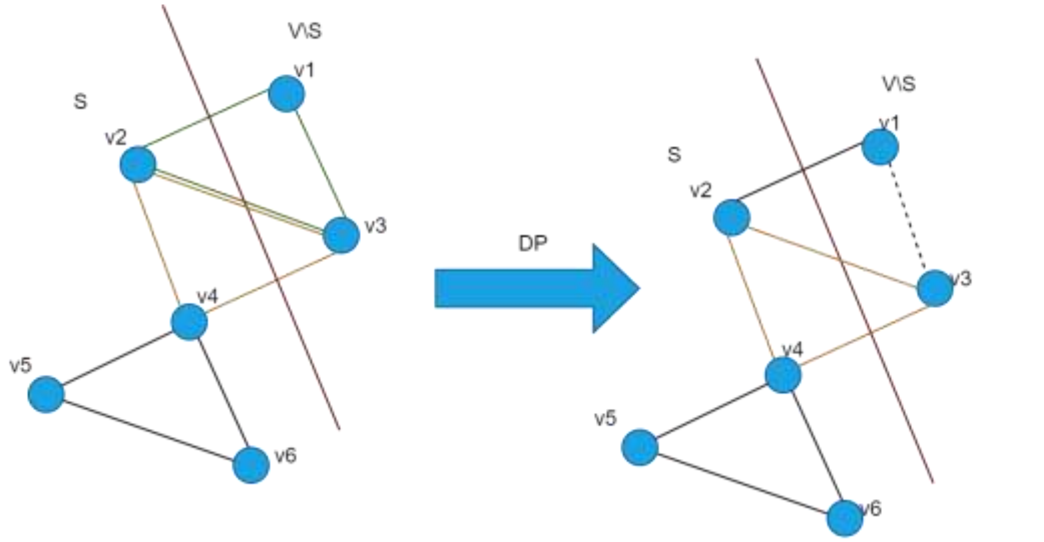


$$\mathrm{Cut}_M^{(G)}(S, V \setminus S) = \sum_{I \in \mathcal{M}(G,M): I \text{ crosses } (S, V \setminus S)} w(I)$$

1 Background

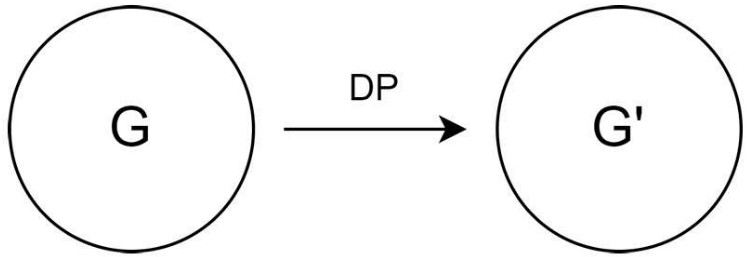
Question: How can we release a synthetic graph under differential privacy while preserving motif-cut structures?

- Generalizing the classical cut problem
- No prior results for motif cuts

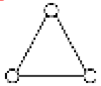
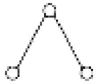
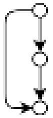
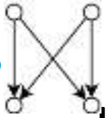


differentially privately preserving triangle-motif cut structures

$$\mathrm{Cut}_M^{(G)}(S, V \setminus S) \approx \mathrm{Cut}_M^{(G')}(S, V \setminus S)$$



1 Background

- Network Analysis: [BGL16]
  - Triangles  → Social network analysis
  - Wedges  → Transportation, healthcare networks
  - Feedforward loops  and bi-fans 

    → Interconnection patterns, neural networks

- Graph Clustering: [BGL16]
  - Some clustering algorithms rely on motif-cut structures
  - Motif-based embeddings capture clustering structures more precisely

- Graph Sparsification: [KMSST22]
  - Motif-cut sparsification can accelerate related methods
  - Applying sparsification to synthetic graphs can preserve privacy

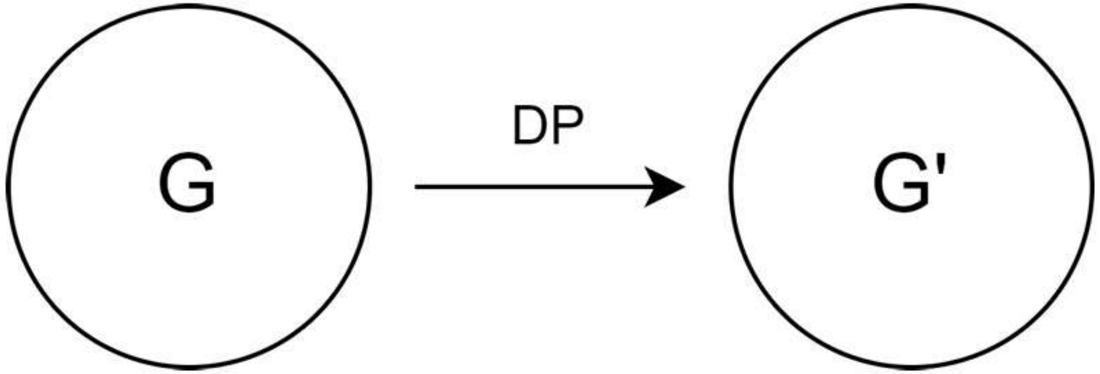
exists private individual information

2 Our Results

We focus on triangle-motif cuts:

- **Upper Bound:** Given an unweighted graph G, we propose an (ε,δ)-DP algorithm that releases a synthetic graph G' in polynomial time, such that with high probability, G' approximates the sizes of all triangle cuts in G with an additive error of at most $\tilde{O}(\sqrt{m\ell_3(G)}n/\varepsilon^{\frac{3}{2}})$
- **Lower Bound**: For a graph G, any (ε,δ)-DP algorithm that approximates the sizes of all triangle cuts in G with high probability must incur an additive error of at least $\Omega(\sqrt{mn\ell_3(G)}/\varepsilon)$

local sensitivity,
$\approx$ d (the maximum degree)



the error of any triangle cut is **at most**:

$$\tilde{O}(\sqrt{m\ell_3(G)}n/\varepsilon^{\frac{3}{2}})$$

$$\left|\mathrm{Cut}_{\triangle}^{(G)}(S,V\setminus S)-\mathrm{Cut}_{\triangle}^{(G')}(S,V\setminus S)\right|=\tilde{O}(\sqrt{m\ell_3(G)}n/\varepsilon^{\frac{3}{2}})$$

3 Overview of Algorithm

Convex optimization problem:

privacy regularizer

convexity regularizer

$$\min_{\mathbf{w}\in\mathcal{X}}\max_{\mathbf{X}\in\mathcal{D}}\{\begin{pmatrix} \mathbf{0} & \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle \\ \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle & \mathbf{0}\end{pmatrix} \bullet \mathbf{X} + \lambda \log\det(\mathbf{X}) + C_2 n \sum_{e\in\binom{V}{2}} (\mathbf{w}_e - \overline{\mathbf{w}}_e)^2\}$$

$$\mathcal{X} = \{\mathbf{w} \in \mathbb{R}_+^{\binom{V}{2}} : \sum_{e\in\binom{V}{2}} \mathbf{w}_e = W, \mathbf{w}_e \leq C_1 \cdot w_{\max}\}$$

Additional constraints added for upper bound analysis

$$\mathcal{D} = \{\mathbf{X} \in \mathbb{R}^{2n} : \mathbf{X} \text{ is symmetric}, \mathbf{X} \succeq \frac{1}{n}\mathbf{I}_{2n}, \text{and } \mathbf{X}_{ii} = 1 \text{ for } \forall i\}$$

Lemma ([BGL16])

$$\mathrm{Cut}_\triangle^{(G)}(S, V\setminus S) = \frac{1}{2}\mathrm{Cut}^{(G_\triangle)}(S, V\setminus S)$$

$$(\mathbf{A}_M)_{ij} = \sum_{I\in\mathcal{M}(G,M):i,j\in V_I} w(I)$$

not for general motifs

3 Overview of Algorithm

Algorithm: (Privately applying stochastic mirror descent method)

for $t = 1, \ldots, T$ **do**

Find the maximizer $\mathbf{X}^{(t)} = \arg\min_{\mathbf{X} \in \mathcal{D}} F_{\triangle}(\mathbf{w}^{(t)}, \mathbf{X})$, where $F_{\triangle}$ is defined in Equation (6);

Choose a random vector $\zeta \sim N(\mathbf{0}, \mathbf{I}_{2n})$ and release $(\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta$;

Compute the approximate gradient for all $e \in \binom{V}{2}$:

$$\mathbf{g}_e^{(t)} = \left((\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta\zeta^T(\mathbf{X}^{(t)})^{\frac{1}{2}}\right) \bullet \begin{pmatrix} \mathbf{0} & \mathbf{D}_{\triangle}^{(e)(t)} \\ \mathbf{D}_{\triangle}^{(e)(t)} & \mathbf{0} \end{pmatrix} + 3 \sum_{s \in V \setminus \{i,j\}} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})(\mathbf{w}_e^{(t)} - \overline{\mathbf{w}}_e + \mathrm{Lap}(\frac{1}{\varepsilon_0}))$$

Mirror Descent Step: $\mathbf{w}^{(t+1)} = \mathrm{MD\_Update}(\mathbf{w}^{(t)}, \mathbf{g}^{(t)}, W, \mathbf{u}, \eta)$;

Guassian noise

newly added noise

novel update step (corresponds to projection)

3 Overview of Algorithm

Novel update step:

Solution:(Greedy: find the nearest solution to the primal)

- Sort the entries e by the decreasing order of $\mathbf{w}_e^{(t)} \exp(-\eta \mathbf{g}_e^{(t)})$.
- Try assign them proportional to $\mathbf{w}_e^{(t)} \exp(-\eta \mathbf{g}_e^{(t)})$.
- If there are ones larger than C_1, truncated.

Correctness: Proved by KKT Conditions

Greedy: Sort by level of constraint violation

Truncates values that exceed constraints

$$\min_{\mathbf{w}} D_\Phi^{(t)}(\mathbf{w}, \mathbf{y}^{(t+1)})$$

$$\text{s.t.} \sum_{e \in \binom{V}{2}} \mathbf{w}_e = W \quad \text{and} \quad \mathbf{w}_e \le \mathbf{u}_e, \forall e$$

$$\underset{\wedge\; C_1}{\mathbf{w}_1 \exp(-\eta \mathbf{g}_1)} \ge \underset{\wedge\; C_1}{\mathbf{w}_2 \exp(-\eta \mathbf{g}_2)} \ge \underset{\vee\; C_1}{\mathbf{w}_3 \exp(-\eta \mathbf{g}_3)} \ge \ldots \ge \underset{\vee\; C_1}{\mathbf{w}_m \exp(-\eta \mathbf{g}_m)}$$

$$\Downarrow = \qquad \Downarrow = \qquad \Downarrow \propto \qquad \Downarrow \propto$$

$$\underbrace{C_1 \qquad C_1}_{2C_1} \qquad \underbrace{\mathbf{w}_3 \exp(-\eta \mathbf{g}_3) \qquad \mathbf{w}_m \exp(-\eta \mathbf{g}_m)}_{\sum_e \mathbf{w}_e - 2C_1}$$

3 Overview of Algorithm

Algorithm 3: MD_Update

Input: $\mathbf{w}^{(t)}, \mathbf{g}^{(t)}, W, \mathbf{u}, \eta$.

Output: $\mathbf{w}^{(t+1)}$.

1: Set $\mathbf{y}_e^{(t+1)} = \mathbf{w}_e^{(t)} \exp(-\eta \mathbf{g}_e^{(t)})$ for $\forall e \in \binom{V}{2}$ and let $N = \binom{n}{2}$

2: Sort edges in non-increasing order so that $\frac{\mathbf{y}_{e_1}^{(t+1)}}{\mathbf{u}_{e_1}} \geq \frac{\mathbf{y}_{e_2}^{(t+1)}}{\mathbf{u}_{e_2}} \geq \cdots \geq \frac{\mathbf{y}_{e_N}^{(t+1)}}{\mathbf{u}_{e_N}}$

3: Compute $S_i = \sum_{j=i}^{N} \mathbf{y}_{e_j}^{(t+1)}$ for $i = 1, \ldots, N$

4: Let $W_1 = W$

5: **for** $i = 1, \ldots, N$ **do**

6: $\quad \mathbf{w}_{e_i}^{(t+1)} = \min(\frac{W_i \cdot \mathbf{y}_{e_i}^{(t+1)}}{S_i}, \mathbf{u}_{e_i})$

7: $\quad$ Set $W_{i+1} = W_i - \mathbf{w}_{e_i}^{(t+1)}$

8: **end for**

9: **return** $\mathbf{w}^{(t+1)}$

optimal solution without additional constraints

Greedy: Sort by level of constraint violation

Truncates values that exceed constraints

4 Overview of Analysis

Privacy:

- The privacy loss per iteration $\approx O(\frac{\ell_3(G)}{\lambda})$
- Total privacy loss after T iterations $\approx O(\frac{\ell_3(G)\sqrt{T}}{\lambda})$
  $\Rightarrow$ run $T \approx \frac{\lambda^2}{\ell_3(G)^2}$ steps to achieve differential privacy

Advanced Composition

Utilization:

- The cut distance to the original graph $\overline{G} \approx O(\frac{mn}{\sqrt{T}} + \lambda n)$
- Choose $T \approx \sqrt{\frac{m}{\ell_3(G)}} \Rightarrow$ the additive error $\approx O(\sqrt{m\ell_3(G)}n)$

error from convex optimization

error from regularization terms

5 Summary

Results：

- The **first** (ε, δ)-DP mechanism：
  - Release a synthetic graph that approximates triangle-motif cut structures
  - Additive error: $\tilde{O}(\sqrt{m\ell_3(G)}n/\varepsilon^{3/2})$
  - Our algorithm generalizes to **weighted graphs**

- A lower bound of additive error: $\Omega(\sqrt{mn\ell_3(G)/\varepsilon})$
  - Our lower bound extends to any K_h-**motif cut**

Open questions:

- Improve the algorithm to achieve smaller additive error
- Generalize the algorithm to work for any motif cut

Thanks!