Differentially Private Release of Synthetic Graphs for Triangle-Motif Cut Structures

Pan Peng, Hangyu Xu

University of Science and Technology of China

ppeng@ustc.edu.cn
hangyuxu@mail.ustc.edu.cn

May 25, 2024

Graphs Are Everywhere

social network

- vertex: person
- edge: friendship relation

A social network illustration (by "paixin.com")

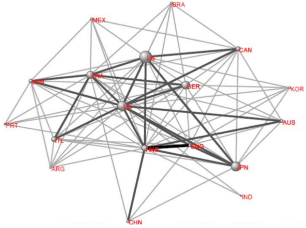
healthcare network

- vertex: person
- edge: doctor-patient relation

A healthcare network illustration (by "athenahealth.com")

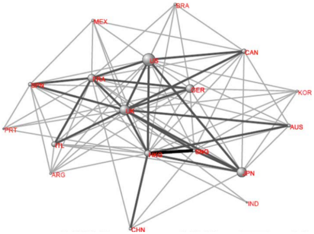
financial network

- vertex: economic entity
- edge: financial transaction-relation



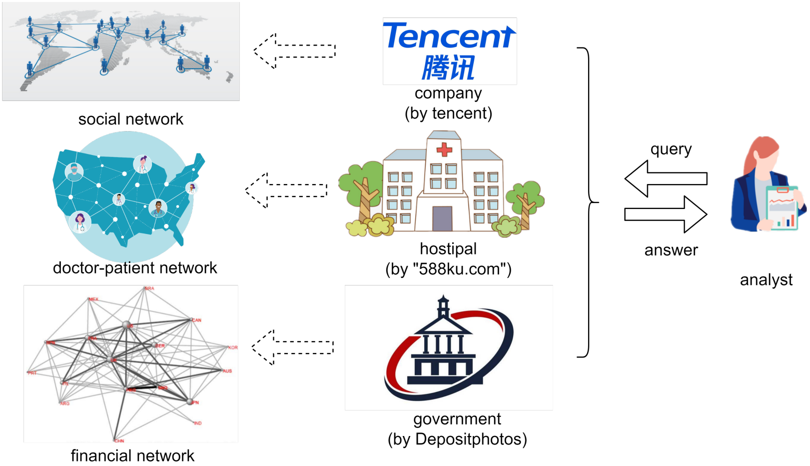
The Global Financial Network in 1985 (by Haldane)

- friendship relation is private
- doctor-patient relation is private
- financial transaction-relation is private
- ...



Private Release of Graph Statistics or Queries

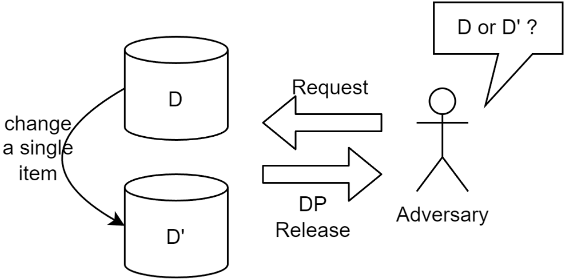
How to protect sensitive information from an analyst who can ask questions about a particular graph?



Differential Privacy

$\Downarrow$

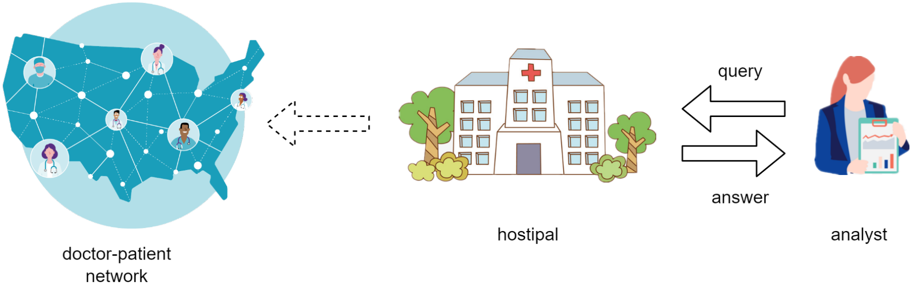
regardless of the adversary/analyst's knowledge about the query answers, the privacy of **individual users (or edges)** remains intact



Differentially Private (DP) Release of Graph Statistics or Queries

Two approaches:

- DP query release mechanisms
  - Input: graph+ queries
  - Output: privacy-preserving responses to the queries
  - dependent on the queries
  - # triangles, degree sequence, cut queries, ...



Differentially Private (DP) Release of Graph Statistics or Queries

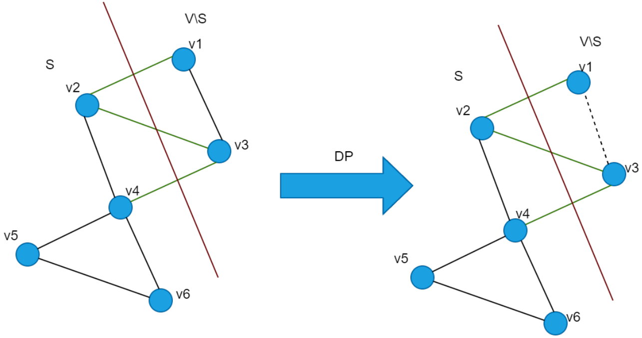
Two approaches:

- DP query release mechanisms
  - Input: graph+ queries
  - Output: privacy-preserving responses to the queries
  - dependent on the queries
  - # triangles, degree sequence, cut queries, ...

- DP release of a synthetic graph
  - Input: graph
  - Output: a graph that well approximates the property of the true one, while preserving privacy.
  - independent of queries
  - cut structure, spectral information, ...

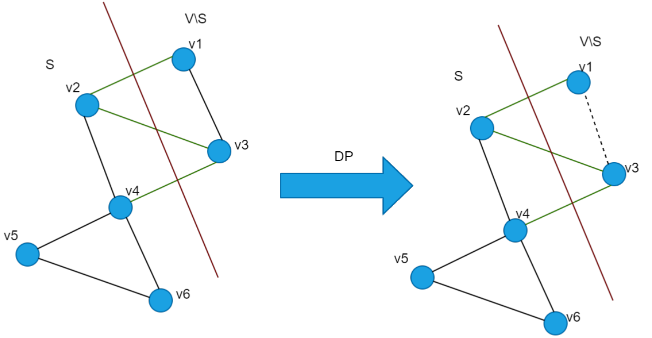
An Example: DP Synthetic Graphs for Cut Structure

- Input: Graph $G = (V, E)$.
- Output: A DP synthetic graph $G' = (V, E')$, such that for every $S \subset V$, $\mathrm{Cut}^{(G)}(S, V \setminus S) \approx \mathrm{Cut}^{(G')}(S, V \setminus S)$.



An Example: DP Synthetic Graphs for Cut Structure

- Input: Graph $G = (V, E)$.
- Output: A DP synthetic graph $G' = (V, E')$, such that for every $S \subset V$, $\mathrm{Cut}^{(G)}(S, V \setminus S) \approx \mathrm{Cut}^{(G')}(S, V \setminus S)$.



- received great attention in DP community: [GRU12], [EKKL20], [LUZ24]
- upper and lower bounds $\Theta(\sqrt{mn/\epsilon})$ [EKKL20], [LUZ24]

One Comment and One Question

Though powerful, the private synthetic graph for cut structure fails to capture higher-order structure of the graph

- higher-order structure: motifs or subgraphs (triangles, cliques, ...)
- edge: can be viewed as low-order structure

Question: **can we privately release a graph that preserves the higher-order cut structure?**

Higher-Order Structure

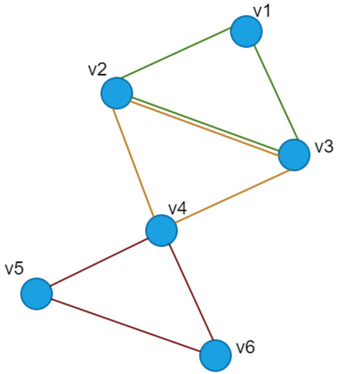
Motif

A **motif** refers to a frequently occurring subgraph within complex networks. ([MSOI+02])

- motif: triangle, wedge, cliques, ...

Applications:

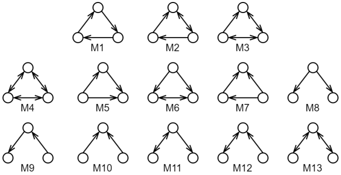
- graph clustering
- graph data visualization
- social/biological network analysis



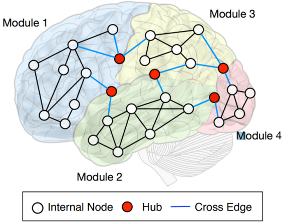
Motif

networks exhibit rich **higher-order** organizational structures ([BGL16])

- $M_1 \sim M_7$ $\Rightarrow$social networks
- M_{13} $\Rightarrow$structural hubs in the brain
- $M_8 \sim M_{13}$ $\Rightarrow$air traffic networks



social network

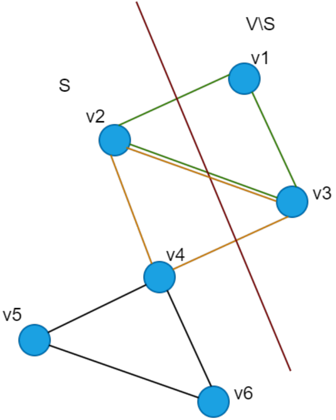


structural hubs in the brain
(by Lifang He)

air traffic networks
(by Marcelo Ferreira da Costa Gomes)

Motif Cut

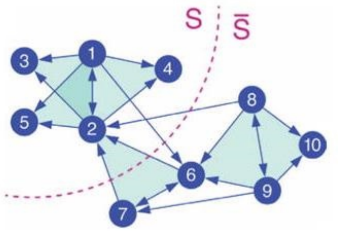
The **motif size** of a cut refers to the amount of motifs crossing the cut.



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

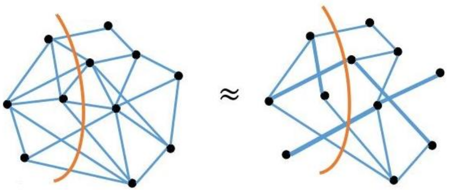
Applications of Motif:

1. graph clustering ([BGL16])
   - motif cut can be used to find better clusters



(by [BGL16])

2. motif cut sparsifier ([KMSST22])
   - speeding up motif-cut based graph algorithms



(by Alexandr Andoni)

The Formal Definitions and Our Problem

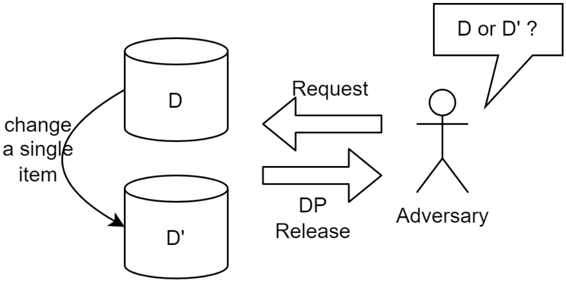
Differential Privacy

Definition ((ε, δ)-differential privacy)

$\mathcal{M}$ is a (ε, δ)-differentially private (DP) algorithm, if

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

where x and y are neighboring inputs (different in weight by 1).



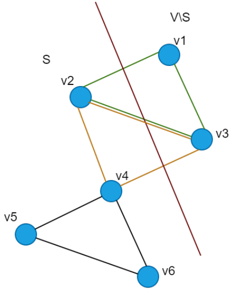
Motif Size of Cut

Given graph G, the M-**motif size** of a cut $(S, V \setminus S)$ is:

- Informally: the sum of weights of the motif M crossing $(S, V \setminus S)$
- Formally:

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

$w(I)$ is the product of weights of the edges of the motif instance I



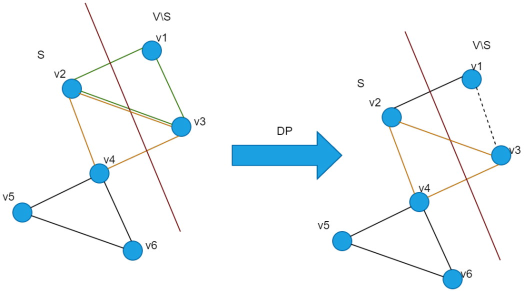
- $S = \{v_2, v_4, v_5, v_6\}$
- $\mathrm{Cut}_{\triangle}^{(G)}(S, V \setminus S) = 2$

The Problem: Private Release of Synthetic Graphs for Motif Cut

- Problem: How one can efficiently release a synthetic graph that well preserves the **motif size** of all cuts, in a differentially private manner?

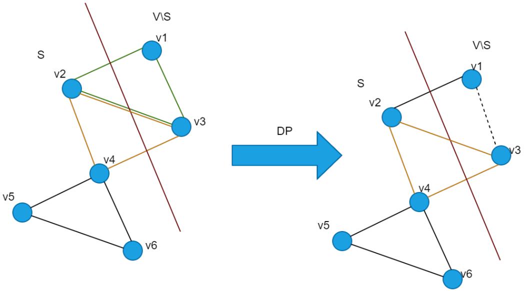
The Problem: Private Release of Synthetic Graphs for Motif Cut

- Problem: How one can efficiently release a synthetic graph that well preserves the **motif size** of all cuts, in a differentially private manner?
- Input: A weighted graph $G = (V, E)$ and a motif M.
- Output: A weighted DP synthetic graph $G' = (V, E')$, such that for every $S \subset V$, $\mathrm{Cut}_M^{(G)}(S, V \setminus S) \approx \mathrm{Cut}_M^{(G')}(S, V \setminus S)$.



The Problem: Private Release of Synthetic Graphs for Motif Cut

- Problem: How one can efficiently release a synthetic graph that well preserves the **motif size** of all cuts, in a differentially private manner?
- Input: A weighted graph $G = (V, E)$ and a motif M.
- Output: A weighted DP synthetic graph $G' = (V, E')$, such that for every $S \subset V$, $\mathrm{Cut}_M^{(G)}(S, V \setminus S) \approx \mathrm{Cut}_M^{(G')}(S, V \setminus S)$.



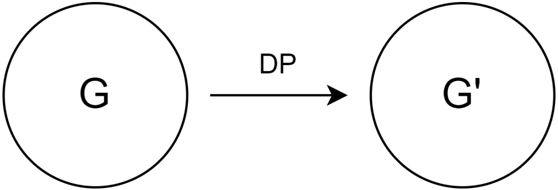
- when motif is edge, it is equivalent to cut structure
- we focus on **triangle motif**

Our Results

Theorem

There exists a polynomial time (ε, δ)-DP algorithm that given an n-vertex unweighted graph G with m edges, outputs a weighted graph G' such that with probability at least 3/4, for any cut (S, T),

$$|\mathrm{Cut}_{\triangle}^{(G)}(S, T) - \mathrm{Cut}_{\triangle}^{(G')}(S, T)| = \tilde{O}(\sqrt{m}n^{\frac{3}{2}}/\varepsilon^3).$$



for every cut
the difference is **at most**
$\tilde{O}(\sqrt{m}n^{\frac{3}{2}}/\varepsilon^3)$

(Informal) Theorem 1 (Upper Bound)

Theorem

There exists a polynomial time (ε, δ)-DP algorithm that given an n-vertex unweighted graph G with m edges, outputs a weighted graph G' such that with probability at least $3/4$, for any cut (S, T),

$$|\mathrm{Cut}_{\triangle}^{(G)}(S, T) - \mathrm{Cut}_{\triangle}^{(G')}(S, T)| = \tilde{O}(\sqrt{m}n^{\frac{3}{2}}/\varepsilon^3).$$

Remark:

- for a dense graph, the trivial approach has additive error as large as $\Theta(n^3)$.
- the edge cut case: $\Theta(\sqrt{mn})$ ([EKKL20], [UZ24])
- for weighted graph, additive error: $\tilde{O}(\sqrt{W}n^{3/2}w_{\max}^2/\varepsilon^3)$ (W: total edge weight; $w_{\max}$: maximum edge weight)
- can be extended to other 3-vertex motifs (e.g., wedges)

(Informal) Theorem 2 (Lower Bound)

Theorem

If an (ϵ, δ)-DP algorithm $\mathcal{M}$ answers the triangle-motif size queries of all (S, T)-cut about G up to an additive error α with probability at least $3/4$, then $\alpha \geq \Omega(\frac{m^2}{\varepsilon \cdot n^{3/2}})$

Remark:

- our algorithm is **nearly optimal** for unweighted dense graphs when $m = \Theta(n^2)$
  - upper bound: $\tilde{O}(\sqrt{m} n^{3/2}/\varepsilon^3) = \tilde{O}(n^{5/2}/\varepsilon^3)$
  - lower bound: $\Omega(\frac{m^2}{\varepsilon \cdot n^{3/2}}) = \Omega(n^{5/2}/\epsilon)$
- for K_h motifs, lower bound: $\Omega\left(\frac{m^{(h-1)^2/2}}{\varepsilon \cdot n^{h^2-3h+3/2}}\right)$.

Tempting Approaches

Tempting Approaches

Approach 1:

- utilize the motif cut sparsifier algorithm ([KMSST22]) and add some noise
- Failed Reason: sampling on existing edges and never outputting non-edges poses obstacle for DP

Tempting Approaches

Approach 1:

- utilize the motif cut sparsifier algorithm ([KMSST22]) and add some noise
- Failed Reason: sampling on existing edges and never outputting non-edges poses obstacle for DP

Approach 2:

- convert to a motif hypergraph and apply a private hyperedge cut release algorithm
- However: we do not know how to give a hypergraph cut release algorithm

Tempting Approaches

Approach 1:

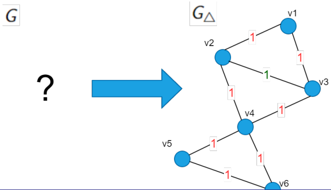
- utilize the motif cut sparsifier algorithm ([KMSST22]) and add some noise
- Failed Reason: sampling on existing edges and never outputting non-edges poses obstacle for DP

Approach 2:

- convert to a motif hypergraph and apply a private hyperedge cut release algorithm
- However: we do not know how to give a hypergraph cut release algorithm

Approach 3:

- convert to a triangle-motif weighted graph and apply a private edge cut release algorithm
- However: the resulting graph may not correspond to a triangle-motif weighted graph



An Inherent Difficulty

- non-linearity: for two graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, the following is NOT true

$$\mathrm{Cut}_{\triangle}^{(G_1)}(S, V \setminus S) + \mathrm{Cut}_{\triangle}^{(G_2)}(S, V \setminus S) = \mathrm{Cut}_{\triangle}^{(G_1+G_2)}(S, V \setminus S)$$

An Inherent Difficulty

- non-linearity: for two graphs $G_1 = (V, E_1)$, $G_2 = (V, E_2)$, the following is NOT true

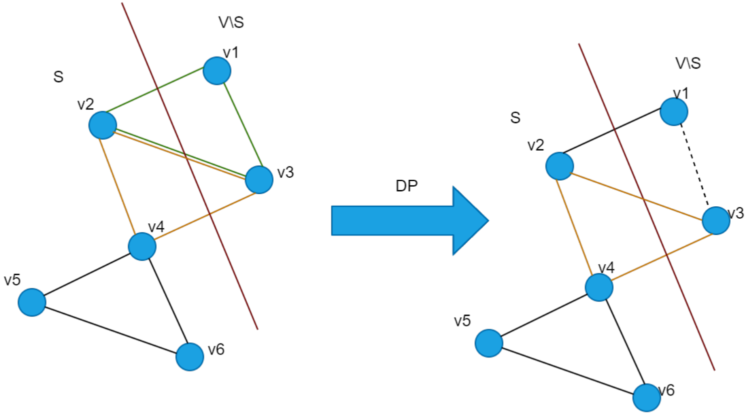
$$\mathrm{Cut}_{\triangle}^{(G_1)}(S, V \setminus S) + \mathrm{Cut}_{\triangle}^{(G_2)}(S, V \setminus S) = \mathrm{Cut}_{\triangle}^{(G_1+G_2)}(S, V \setminus S)$$

- This is in sharp contrast with the edge case

Our Approach

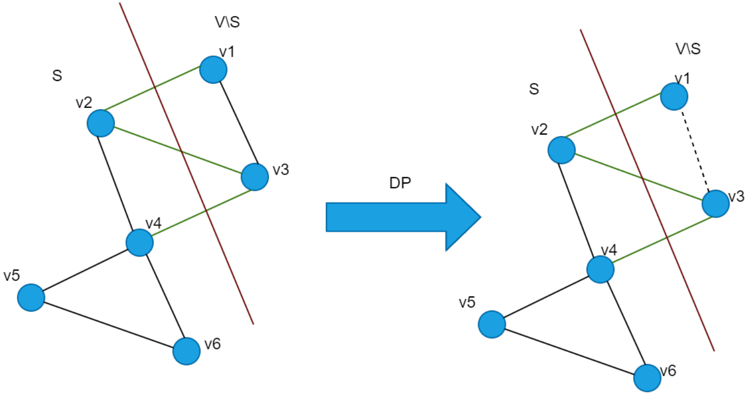
Recall: Our Problem

Given $\overline{G}$ with m edges, the algorithm tries to privately find a synthetic graph G, which approximates the triangle-motif cut of $\overline{G}$.



Recall: The [EKKL20] Approach for Edge Cuts

Given $\overline{G}$ with m edges, the algorithm tries to privately find a synthetic graph G, which approximates the edge cut of $\overline{G}$.



Recall: The [EKKL20] Approach for Edge Cuts

The cut difference in $\overline{G}$ and G, can be bounded by the **SDP**:

$$\max_{\mathbf{X}\in\mathcal{D}}\{\begin{pmatrix} \mathbf{0} & \mathbf{A}-\overline{\mathbf{A}} \\ \mathbf{A}-\overline{\mathbf{A}} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X}\}$$

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

$$\max_{\mathbf{x},\mathbf{y}\in\{0,1\}^n} \mathbf{x}^T\mathbf{A}\mathbf{y}$$

$$\Downarrow$$

$$\max_{\|\mathbf{u}_i\|=\|\mathbf{v}_i\|=1} \sum_{i,j} A_{i,j}\mathbf{u}_i\mathbf{v}_j$$

$$\Downarrow$$

$$\max_{\mathbf{X}\in\mathcal{D}}\{\begin{pmatrix} \mathbf{0} & \mathbf{A}-\overline{\mathbf{A}} \\ \mathbf{A}-\overline{\mathbf{A}} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X}\}$$

Recall: The [EKKL20] Approach for Edge Cuts

To preserve differential privacy, add **regularizer** $\lambda \log \det(\mathbf{X})$.

Recall: The [EKKL20] Approach for Edge Cuts

To preserve differential privacy, add **regularizer** $\lambda \log \det(\mathbf{X})$.

$\Rightarrow$ Try to solve the following **optimization problem**:

$$\min_{\mathbf{w}\in\mathcal{X}'} \max_{\mathbf{X}\in\mathcal{D}} \{ \begin{pmatrix} \mathbf{0} & \mathbf{A}-\overline{\mathbf{A}} \\ \mathbf{A}-\overline{\mathbf{A}} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} + \lambda \log \det(\mathbf{X}) \}$$

where the total weight of G is m (privately released).

Recall: The [EKKL20] Approach for Edge Cuts

The objective function $f(\mathbf{w})$ is **convex**, optimize it by **stochastic mirror descent**:

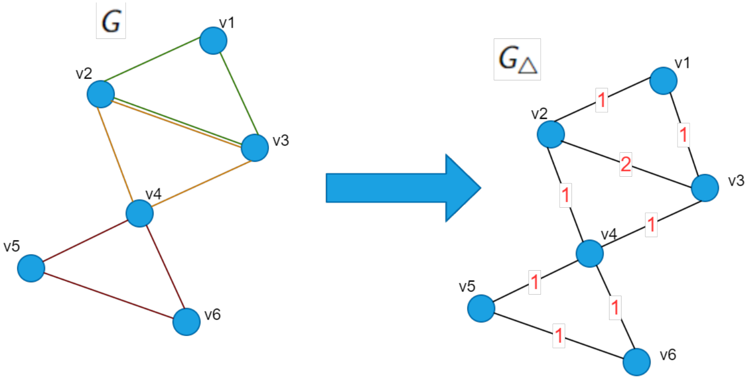
For T iterations:

- Privately release the gradient of $f(\mathbf{w})$: g_e
- Mirror descent step: $\mathbf{w}_e^{(t+1)} = m \frac{\mathbf{w}_e^{(t)} \exp(-\eta g_e^{(t)})}{\sum_e \mathbf{w}_e^{(t)} \exp(-\eta g_e^{(t)})}$

A Tool: Motif Weighted Graph

Given graph G and motif M, one can compute a **triangle-motif weighted** graph $G_\triangle$ with a weight vector $\mathbf{w}_\triangle$

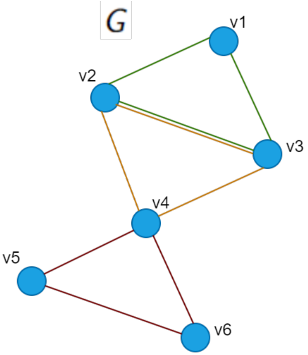
$\Rightarrow \mathbf{w}_\triangle(i,j)$ denotes the amount of triangles containing i,j simultaneously



A Tool: Motif Weighted Graph

Given graph G and motif M, one can compute a **triangle-motif weighted** graph $G_\triangle$ with a weight vector $\mathbf{w}_\triangle$

$\Rightarrow \mathbf{w}_\triangle(i,j)$ denotes the amount of triangles containing i,j simultaneously



$\mathbf{A}_\triangle$

	$v1$	$v2$	$v3$	$v4$	$v5$	$v6$
$v1$	0	1	1	0	0	0
$v2$	1	0	2	1	0	0
$v3$	1	2	0	1	0	0
$v4$	0	1	1	0	1	1
$v5$	0	0	0	1	0	1
$v6$	0	0	0	1	1	0

A Useful Equation

Lemma ([BGL16])

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

A Useful Equation

Lemma ([BGL16])

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

Cor:(considering edge cut in $G_{\triangle}$)
The maximum triangle-motif size cut difference in $\overline{G}$ and G, can be bounded by the following SDP up to a constant factor:

$$\max_{\mathbf{X} \in \mathcal{D}} \{ \begin{pmatrix} \mathbf{0} & \mathbf{A}_{\triangle} - \overline{\mathbf{A}}_{\triangle} \\ \mathbf{A}_{\triangle} - \overline{\mathbf{A}}_{\triangle} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} \}$$

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

A Useful Equation

Lemma ([BGL16])

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

Cor:(considering edge cut in $G_\triangle$)
The maximum triangle-motif size cut difference in $\overline{G}$ and G, can be bounded by the following SDP up to a constant factor:

$$\max_{\mathbf{X} \in \mathcal{D}} \{ \begin{pmatrix} \mathbf{0} & \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle \\ \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} \}$$

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

Note: We should optimize it w.r.t. $\mathbf{w}$ instead of $\mathbf{w}_\triangle$; o.w. we can't convert it back to a graph.

Overview of Algorithm

- The objective function with respect to $\mathbf{w}$ is **not convex**, we add a convexity regularizer $C_2 n \sum_{e \in \binom{V}{2}} (\mathbf{w}_e - \overline{\mathbf{w}}_e)^2$.
- To bound the error, we additionally require $\mathbf{w}$ to satisfy $\mathbf{w}_e \leq C_1$. (For weighted graph, here is $\mathbf{w}_e \leq C_1 w_{\max}$)
- i.e. $\mathcal{X} = \{\mathbf{w} \in \mathbb{R}_+^{\binom{V}{2}} : \sum_{e \in \binom{V}{2}} \mathbf{w}_e = W, \mathbf{w}_e \leq C_1 \cdot w_{\max}\}$

Overview of Algorithm

Solve the following **optimization problem**:

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

Overview of Algorithm

Solve the following **optimization problem**:

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

Overview of Algorithm

Solve the following **optimization problem**:

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

- The objective function $f_\triangle(\mathbf{w}) = \max_{\mathbf{X}\in\mathcal{D}} F_\triangle(\mathbf{w}, \mathbf{X})$ is **convex**

Overview of Algorithm

Solve the following **optimization problem**:

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

- The objective function $f_\triangle(\mathbf{w}) = \max_{\mathbf{X}\in\mathcal{D}} F_\triangle(\mathbf{w}, \mathbf{X})$ is **convex**
- optimize it by **stochastic mirror descent**

Overview of Algorithm

Solve the following **optimization problem**:

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

- The objective function $f_\triangle(\mathbf{w}) = \max_{\mathbf{X}\in\mathcal{D}} F_\triangle(\mathbf{w}, \mathbf{X})$ is **convex**
- optimize it by **stochastic mirror descent**

– the resulting graph is defined by the weight **w**

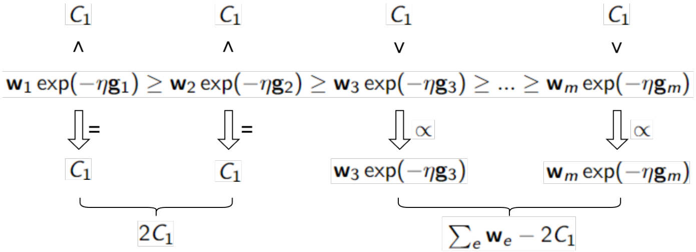
Overview of Analysis

Trouble: Since we require $\mathbf{w}$ to satisfy $\mathbf{w}_e \le C_1$, the previous mirror descent step becomes invalid.

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



Overview of Analysis

Privacy:

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

Overview of Analysis

Privacy:

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

Utilization:

- The cut distance to the original graph $\overline{G} \approx O(\frac{mn}{\sqrt{T}} + \lambda n)$
- Choose $T \approx \sqrt{\frac{m}{n}} \Rightarrow$ the additive erorr $\approx O(\sqrt{m}n^{3/2})$

The Lower Bound

Overview of Lower Bound

We employ the generalized discrepancy of 3-colorings of h-uniform hypergraphs.

Definition

Let $\mathbf{B}$ be a 0/1 matrix with $\binom{n}{3}$ columns and $\mathcal{C} \subseteq \{-1, 0, 1\}^{\binom{n}{3}}$ be the set of allowed K_3 colorings. Then

$$\mathrm{disc}_{\mathcal{C}}(\mathbf{B}) = \min\{\|\mathbf{B}\chi\|_\infty : \chi \in \mathcal{C}\}$$

Remark:

- If a mechanism is equivalent to compute some $\mathbf{Bx}$ with input $\mathbf{x}$, then $\mathrm{disc}_{\mathcal{C}}(\mathbf{B})$ somehow measures the difference between the output with **neighboring inputs**.

Definition

Construct a matrix $\mathbf{A}$ with $\binom{n}{3}$ columns:

$$\mathbf{A}_{(S,T),I} = \begin{cases} 1 & \text{if } I \in (S \times T) \\ 0 & \text{otherwise} \end{cases}$$

Remark:

- $\mathbf{A}$ is fixed and does not depend on G.
- Let $\mathbf{x}_{K_3} \in \{0,1\}^{\binom{n}{3}}$ be the indicator vector of K_3 in G. Then

$$(\mathbf{A}\mathbf{x}_{K_3})_{S,T} = \mathrm{cut}_{K_3}^{(G)}(S,T)$$

- $\mathbf{A}\mathbf{x}_{K_3}$ somehow measures the difference of motif size between neighboring graphs.

Overview of Lower Bound

Lemma ((Informal) discrepancy lower bound)

Given parameters σ, γ, *let* $\mathcal{C}_{\sigma,\gamma}$ *be the set of all vectors* $\chi = \mathbf{x}_{K_3} - \mathbf{x}'_{K_3}$ *where* $\mathbf{x}, \mathbf{x}'$ *are the indicator vector of edges of graphs satisfying some conditions.*

$$\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}) \geq \Omega(\sigma \cdot \gamma^2 n^{5/2}).$$

Lemma

If $\mathcal{M}$ is a mechanism for the motif size of all cuts that outputs $\mathbf{y}$ *with the input x, i.e.,* $\mathbf{y} = \mathcal{M}(\mathbf{x})$, *s.t.*

$$\|\mathbf{y} - \mathbf{A} \cdot \mathbf{x}_{K_3}\|_\infty \leq \frac{1}{2}\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}).$$

Then there exists a deterministic algorithm $\mathcal{A}$ s.t.

$$\|\mathcal{A}(\mathbf{y}) - \mathbf{x}\|_1 \leq \sigma\gamma n^2.$$

Overview of Lower Bound

Lemma

If $\mathcal{M}$ is a mechanism for the motif size of all cuts that outputs $\mathbf{y}$ with the input x, i.e., $\mathbf{y} = \mathcal{M}(\mathbf{x})$, s.t.

$$\|\mathbf{y} - \mathbf{A} \cdot \mathbf{x}_{K_3}\|_\infty \leq \frac{1}{2}\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}).$$

Then there exists a deterministic algorithm $\mathcal{A}$ s.t.

$$\|\mathcal{A}(\mathbf{y}) - \mathbf{x}\|_1 \leq \sigma\gamma n^2.$$

Remark:

- We can approximately recover the input graph if the output of a DP mechanism is too close to the accurate value.
  $\Rightarrow$ The mechanism can't be private.

Overview of Lower Bound

Lemma

If $\mathcal{M}$ is a mechanism for the motif size of all cuts that outputs $\mathbf{y}$ with the input x, i.e., $\mathbf{y} = \mathcal{M}(\mathbf{x})$, s.t.

$$\|\mathbf{y} - \mathbf{A} \cdot \mathbf{x}_{K_3}\|_\infty \leq \frac{1}{2}\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}).$$

Then there exists a deterministic algorithm $\mathcal{A}$ s.t.

$$\|\mathcal{A}(\mathbf{y}) - \mathbf{x}\|_1 \leq \sigma\gamma n^2.$$

Remark:

- We can approximately recover the input graph if the output of a DP mechanism is too close to the accurate value.
  $\Rightarrow$ The mechanism can't be private.
- The lower bound generalizes to any K_h-motif cut.

Conclusion

Summary: Our Results

- The **first** (ε, δ)-DP mechanism:
  - Release a synthetic graph that approximate the triangle-motif cut structures
  - Additive error: $\tilde{O}(\sqrt{m}n^{3/2}/\varepsilon^3)$
  - Our algorithm generalizes to **weighted graphs**
- A lower bound of error: $\Omega(m^2/(n^{3/2} \cdot \varepsilon))$
  - Our lower bound extends to any K_h-**motif cut**

$\Rightarrow$ **Nearly optimal** for unweighted dense graphs

Summary: Our Results

- The **first** (ε, δ)-DP mechanism:
  - Release a synthetic graph that approximate the triangle-motif cut structures
  - Additive error: $\tilde{O}(\sqrt{m} n^{3/2} / \varepsilon^3)$
  - Our algorithm generalizes to **weighted graphs**
- A lower bound of error: $\Omega(m^2 / (n^{3/2} \cdot \varepsilon))$
  - Our lower bound extends to any K_h-**motif cut**

$\Rightarrow$ **Nearly optimal** for unweighted dense graphs

Open questions:

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

Summary: Our Results

- The **first** (ε, δ)-DP mechanism:
  - Release a synthetic graph that approximate the triangle-motif cut structures
  - Additive error: $\tilde{O}(\sqrt{m}n^{3/2}/\varepsilon^3)$
  - Our algorithm generalizes to **weighted graphs**
- A lower bound of error: $\Omega(m^2/(n^{3/2} \cdot \varepsilon))$
  - Our lower bound extends to any K_h-**motif cut**

$\Rightarrow$ **Nearly optimal** for unweighted dense graphs

Open questions:

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

Thanks!