Communication-Efficient Triangle Counting under Local Differential Privacy

Jacob Imola* (UCSD)  Takao Murakami* (AIST)  Kamalika Chaudhuri (UCSD)

Subgraph Counts
- **Triangle** is a set of 3 nodes with 3 edges.
- **$k$-star** consists of a central node connected to $k$ other nodes.

<table>
<thead>
<tr>
<th>Shape</th>
<th>Name</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Triangle</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2-star</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>3-star</td>
<td>6</td>
</tr>
</tbody>
</table>

Clustering Coefficient
- Probability that two friends of a user is also a friend. → Useful for friend suggestion.
- $= 3 \times \#\text{triangles} / \#\text{2-stars}$ (40% in the above graph).
Outline

- Privacy Issues
  - #Triangles/#k-stars can reveal sensitive edges. [Imola+, UseSec21]

- Local Differential Privacy (LDP)
  - User obfuscates her personal data by herself (i.e., no trusted third party).
  - Privacy is protected against attackers with any background knowledge.

Outline diagram:

- **Local DP**
  - Original data: \(\text{randomizer} \rightarrow \mathcal{R} \rightarrow \text{noisy data} \)

- **Centralized DP**
  - Database: \(D\)
  - Function: \(f(D) + \text{Lap.}\)
  - Illegal access
Outline

- Subgraph Counting under LDP [Imola+, UseSec21]
  - \(k\)-stars can be accurately estimated within 1 round.
  - #triangles can be accurately estimated within 2 rounds.
  - But the DL cost is extremely large, e.g., 400 Gbits (6 hours when 20 Mbps).

Our Contributions
- We dramatically reduce the DL cost with several new algorithmic ideas.
  - 400 Gbits (6 hours) \(\rightarrow\) 160 Mbits (8 seconds). 😊
Contents

**Preliminaries**
(LDP on Graphs, [Imola+, UseSec21])

**Our Proposal**
(Overview, Selective DL, Double Clipping)

**Experiments**
(Datasets, Experimental Results)
LDP on Graphs

- **Graph**
  - Can be represented as an adjacency matrix $A$ (1: edge, 0: no edge).
  - User $v_i$ knows her neighbor list $a_i$ ($i$-th row of $A$).

- **Local Graph Model**
  - User $v_i$ obfuscates her neighbor list $a_i$ and sends noisy data $\mathcal{R}_i(a_i)$ to a server.
LDP on Graphs

\- Edge LDP [Qin+, CCS17]
  \- Protects a single bit in a neighbor list $a \in \{0,1\}^n$ with privacy budget $\varepsilon$.

Randomizer $\mathcal{R}$ provides $\varepsilon$-edge LDP if for all $a, a' \in \{0,1\}^n$ that differ in one bit and all $y \in Y$,

$$\Pr[\mathcal{R}(a) = y] \leq e^\varepsilon \Pr[\mathcal{R}(a') = y]$$

\- 1 edge affects 2 elements of $A$ $\Rightarrow$ each edge is protected with at most $2\varepsilon$.
\- Our triangle algorithm uses only $\Rightarrow$ each edge is protected with $\varepsilon$.

adjacency matrix $A$

graph $G$

graph $G'$

Indistinguishable (at most $2\varepsilon$)
Triangle Counting under LDP [Imola+, UseSec21]

- Triangles
  - Challenging because a user cannot see an edge between others.

- 1st Round
  - Each user applies RR to each bit of her neighbor list. → edge LDP.
  - Each user sends noisy edges. Server publishes noisy graph $G'$.
Triangle Counting under LDP [Imola+, UseSec21]

- 2nd Round
  - Each user can count **triangles including one noisy edge** using noisy graph $G'$.
  - Each user sends $\#$noisy triangles (+ corrective term) + Lap. $\rightarrow$ edge LDP.
  - Server calculates an unbiased estimate of $\#$triangles.

DL cost is extremely large because $G'$ is dense. 😞
Contents

Preliminaries
(LDP on Graphs, [Imola+, UseSec21])

Our Proposal
(Overview, Selective DL, Double Clipping)

Experiments
(Datasets, Experimental Results)
Overview

- Our Approach
  - We use asymmetric RR to make a sparse noisy graph $G'$. 
    - DL cost is significantly reduced at the cost of the estimation error.
  - We propose two techniques (selective DL and double clipping) to reduce the error.
Selective Download

- Full DL Strategy (ARRFull)
  - User $v_i$ downloads all noisy edges, i.e., noisy graph $G'$.
  - 1 noisy edge $(v_j, v_k)$ causes 2 incorrect noisy triangles $\triangle$. $\rightarrow$ Large estimation error.

- Selective DL Strategies (ARROneNS and ARRTwoNS)
  - Make the two triangles less correlated with each other by adding independent noise.
  - In ARROneNS, $v_i$ downloads noisy edge $(v_j, v_k)$ s.t. $(v_i, v_k)$ is a noisy edge.
  - In ARRTwoNS, $v_i$ downloads noisy edge $(v_j, v_k)$ s.t. $(v_i, v_j)$ and $(v_i, v_k)$ are noisy edges.

<table>
<thead>
<tr>
<th>Original Graph (4-Cycle)</th>
<th>ARRFull</th>
<th>ARROneNS</th>
<th>ARRTwoNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_j \quad v_i \quad v_k \quad v'_i \quad v'_j \quad v'_k$</td>
<td>$v_j \quad v_i \quad v_k \quad v'_i \quad v'_j \quad v'_k$</td>
<td>$v_j \quad v_i \quad v_k \quad v'_i \quad v'_j \quad v'_k$</td>
<td>$v_j \quad v_i \quad v_k \quad v'_i \quad v'_j \quad v'_k$</td>
</tr>
</tbody>
</table>

Variance of the estimate ($d_{\text{max}}$: maximum degree)

- $O(nd_{\text{max}}^3)$
- $O(nd_{\text{max}}^2)$
- $O(nd_{\text{max}}^2)$
Double Clipping

Laplacian Noise

- [Imola+, UseSec21] added $\text{Lap}\left(\frac{d_{\text{max}}}{\varepsilon}\right)$ ($d_{\text{max}}$: maximum degree) at the 2nd round.
- But the sensitivity of #noisy triangles is much smaller than $d_{\text{max}}$ because:
  1. User $v_i$’s degree $d_i$ is much smaller than $d_{\text{max}}$.
  2. Noisy edges are sparse. → #noisy triangles involving $(v_i, v_j)$ is much smaller than $d_i$.

Double Clipping

- Dramatically reduces sensitivity by (1) edge clipping and (2) noisy triangle clipping.
Double Clipping

- **Edge Clipping**
  - Add the Laplacian noise (+ non-negative value) to user $v_i$’s degree $d_i$.
  - If $d_i$ exceeds the noisy degree $\tilde{d}_i$, remove edges to ensure $d_i \leq \tilde{d}_i$.

- **Noisy Triangle Clipping**
  - If #noisy triangles exceeds a threshold $\kappa_i$, reduce it to ensure #noisy triangles $\leq \kappa_i$.
  - We set $\kappa_i$ s.t. the triangle excess probability is very small, e.g., $10^{-6}$.

We use $\kappa_i$ ($\ll d_{max}$) as the sensitivity.
Contents

Preliminaries
(LDP on Graphs, [Imola+, UseSec21])

Our Proposal
(Overview, Selective DL, Double Clipping)

Experiments
(Datasets, Experimental Results)
Datasets

- **Gplus (Google+ Dataset)**
  - Social graph with 107614 nodes (users).
  - Average degree = 113.7.

- **IMDB (Internet Movie Database)**
  - Graph with 896308 nodes (actors).
  - Average degree = 63.7. More sparse than Gplus.
Experimental Results

- **Result 1: Relative Error vs. DL Cost**
  - Our proposals download user IDs for 1 (edges).
  - \([\text{Imola +, UseSec21}]\) downloads 0/1 for each user-pair \(\rightarrow\) 6G (Gplus) and 400G (IMDB).
  - In IMDB, our proposals achieve 160M bits with high accuracy (relative error \(\ll 1\)).

---

**Gplus (\(\varepsilon = 1\))**

- **[Imola +, UseSec21]**

**IMDB (\(\varepsilon = 1\))**

- **[Imola +, UseSec21]**

---

**Data Points**

- **Gplus**
  - \(190K\) 1.9M 19M 190M 1.9G 19G 190G
  - Relative Error
  - DL Cost (bits) 1.6G 16G 160G 16T

- **IMDB**
  - \(16M\) 160M 1.6G 16G 160G
  - Relative Error
  - DL Cost (bits) 1.6T 16T
Result 2: Full DL vs. Selective DL

- Selective DL significantly outperforms Full DL.
- ARROneNS outperforms ARRTwoNS. In ARRTwoNS, all noisy triangles have noisy edge \((v_i, v_j)\) in common and the sensitivity is not effectively reduced by double clipping.

Experiments

<table>
<thead>
<tr>
<th></th>
<th>Relative Error</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gplus</strong> ((\varepsilon = 1))</td>
<td>0.08</td>
</tr>
<tr>
<td><strong>Gplus</strong> ((\varepsilon = 2))</td>
<td>0.06</td>
</tr>
<tr>
<td><strong>IMDB</strong> ((\varepsilon = 1))</td>
<td>0.04</td>
</tr>
<tr>
<td><strong>IMDB</strong> ((\varepsilon = 2))</td>
<td>0.02</td>
</tr>
</tbody>
</table>

DL Cost = 190 Mbits

DL Cost = 16 Gbits

- **ARRFull** (double clipping)
- **ARROneNS** (double clipping)
- **ARRTwoNS** (double clipping)
Conclusions

This Work
- We proposed communication-efficient triangle counting under LDP with new algorithmic ideas: asymmetric RR, selective DL, and double clipping.

Future Work: 1-Round Triangle Counting
- We showed that this is possible in the shuffle model: https://arxiv.org/abs/2205.01429
- We would like to investigate whether this is possible under the local model.

400 Gbits (6 hours) $\rightarrow$ 160 Mbits (8 seconds)
Thank you for your attention!

Q&A

jimola at eng.ucsd.edu, takao-murakami at aist.go.jp