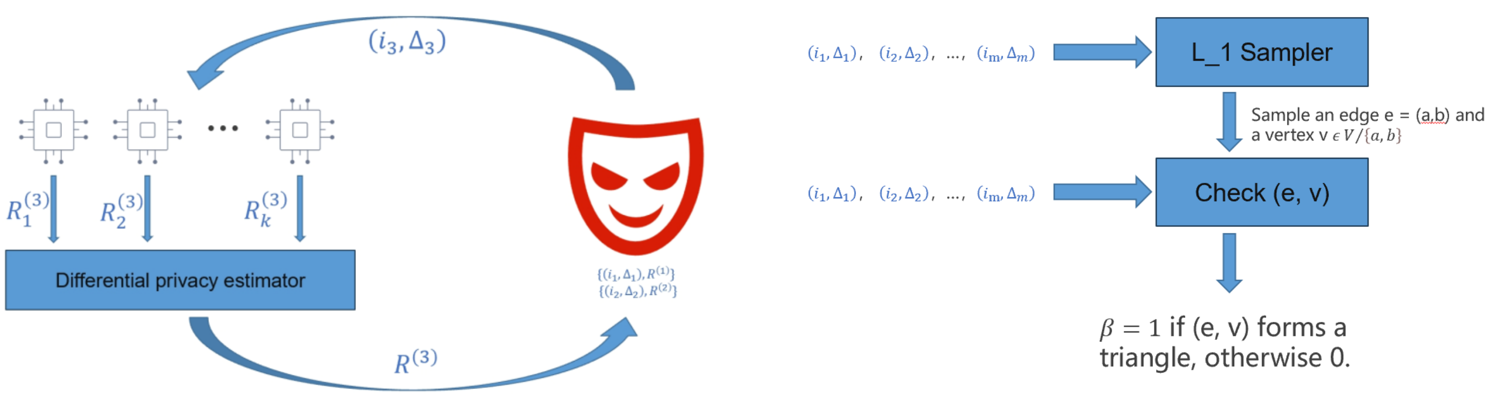
Streaming Algorithms for Triangle Counting: Adversarial Robustness and the Weighted Case

Jing CAO, Yicheng PAN, Pan PENG

Frontiers of Computer Science, DOI: 10.1007/s11704-025-41203-9

Problems & Ideas

- Problems of streaming triangle counting approaches:
  - Most streaming algorithms only give results at the end of stream
  - Existing streaming algorithms assume data stream generators lack continuous feedback and dynamic adjustment.
  - Existing streaming algorithms are only applicable to unweighted graphs.
- Ideas: We use Sketch switch technique, differential privacy, and L_1 Sampling to design algorithms to solve the above three problems



Left: Our adversarially robust algorithm's core idea - a data stream generator retains past information to create adaptive streams, employing differential privacy for robust results. Right: we provide the main idea of our proposed algorithm, which leverages L_1 sampling techniques in its design.

Main Contributions

- Contributions:
  - **Theorem 1.** There exists an adversarially robust streaming algorithm that $(1 \pm \varepsilon)$ approximates the number of triangles at each step of the stream using $\tilde{O}\left(\frac{1}{\varepsilon^{2.5}} \cdot \max_{i \in [m]} \left\{\frac{i \cdot n}{T^{(i)}}\right\}\right)$ bits of space.

    (m: #edges, n: #vertices, $T^{(i)}$: #triangles after the i-th update.)
  - **Theorem 2.** There exists a 2-pass fully dynamic streaming algorithm for $(1 \pm \varepsilon)$-approximating the weighted triangle counting $W_P(G)$, using $\tilde{O}\left(\frac{1}{\varepsilon^3} \cdot \frac{W_E \cdot n}{W_P(G)}\right)$ bits of space.

    (W_E: the total edge weight, $W_P(G)$: total triangle weight, the weight of a triangle is either arithmetic mean, maximum or minimum of its edges.)

  - We performed experiments[1] to validate our algorithms, It performs well in terms of accuracy and space efficiency.

1 see code in: https://github.com/WFEcj/My-FCS-Exp