Edge Local Differential Privacy for Dynamic Graphs[1]

Presented by
Sudipta Paul

Nausica Group,
Department of Computing Science,
Umeå University

Supervisor: **Vicenç Torra**

[1] It is a collaboration work with Julián Salas from Universitat Oberta de Catalunya, Barcelona, Spain. This research was partially funded by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

Outline

Introduction

- Huge amount of data is generated every day in networked systems.
- Nevertheless, in reality, nearly all networks undergo changes, with nodes or edges arriving or going away as the system develops.
- Therefore, static graph networks are not adequate to model these kinds of network structures.

Dynamic Graph

- Data on dynamic networks here is a collection of successively obtained, equally spaced snapshots of the network topology
- These snapshots are a set of different networks defined on the same set of nodes.
- A dynamic network graph model consists of an initial state G_0 and states G_i, for $i = 1, \ldots T$, defined by:

$$G_i = \mathcal{A}_{1-\alpha,1-\beta}(G_{i-1})$$

We will denote it as: $G(G_0, T, \alpha, \beta)$.

Challenges of Privacy in Dynamic Graph

- The adversaries can use their information about the structural graph to infer private information from the graph.
- Proper privacy models have been developed for static graphs following k-anonymity and differential privacy.

Motivation

- The extension of the definition of local differential privacy for edges to dynamic graphs;
- The privacy mechanisms for providing graphs compliant with edge-local differential privacy for dynamic graphs. This is achieved by applying the noise-graph mechanism;

Noise Graph Mechanism

For any graph G with n nodes, and two probabilities p_0 and p_1 We define the following noise-graph mechanism:

$$\mathcal{A}_{p_0,p_1}(G) = G \oplus G_0 \oplus G_1,$$

Such that:

$$G_0 = G' \setminus G \text{ for } G' \in \mathcal{G}(n, 1-p_0)$$
$$G_1 = G'' \cap G \text{ for } G'' \in \mathcal{G}(n, 1-p_1).$$

Stochastic matrix associated to the noise graph

The probabilities of randomization of an edge or a non-edge in a graph G after applying the noise-graph mechanism $\mathcal{A}_{p_0,p_1}$ are represented by the following stochastic matrix:

$$P = P(\mathcal{A}_{p_0,p_1}) = \begin{pmatrix} p_0 & 1-p_0 \\ 1-p_1 & p_1 \end{pmatrix}$$

Dynamic-network-graph-model

The dynamic network graph model consists of an initial state G_0 and states G_t, for $t = 1, \ldots T$, defined by:

$$G_t = \mathcal{A}_{1-\alpha,1-\beta}(G_{t-1})$$

We will denote it as: $G(G_0, T, \alpha, \beta)$.

Local differential privacy

A randomized algorithm π, satisfies ε-local differential privacy if for all inputs x, x' and all outputs $y \in Range(\pi)$:

$$P(\pi(x) = y) \leq e^{\varepsilon} P(\pi(x') = y) \tag{1}$$

Edge Differential Privacy

- A randomized algorithm $\mathcal{A}$ satisfies ε-edge local differential privacy if for all pairs of nodes u, v, all times stamps t and edge values i, j, k:

$$Pr[\mathcal{A}(u, v, t; i) = k] \leq e^{\varepsilon} Pr[\mathcal{A}(u, v, t; j) = k]$$

  we say that $\mathcal{A}$ is ε-edge locally differentially private (ε-eLDP).

Algorithm 1: Dynamic Network mechanism

Let $G = G_0, \ldots G_T$ be a dynamic graph. We define the dynamic network mechanism as:

$$\mathcal{D}_{p_0,p_1}(G) = G(g_0, T, 1 - p_0, 1 - p_1),$$

where, $g_0 = \mathcal{A}_{p_0,p_1}(G_0)$. That is, the protected dynamic graph $g_0, g_1, \ldots, g_T$ corresponds to

$$g_i = \mathcal{A}^{i+1}_{p_0,p_1}(G_0).$$

Algorithm 2: Parallel Protection Mechanism

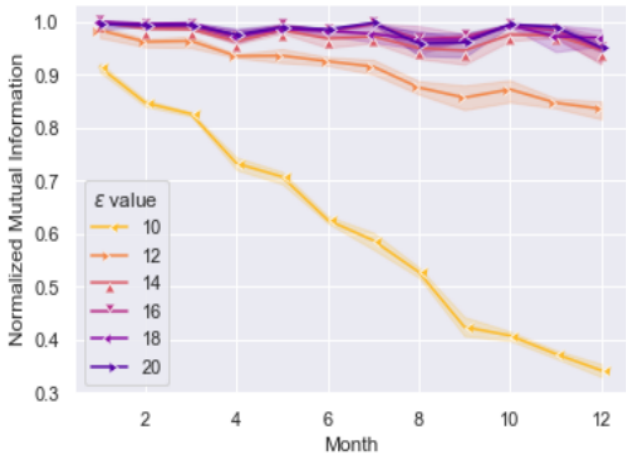
Let $G = G_0, G_1, \ldots, G_T$ be a dynamic graph. Let $\mathcal{A}_{p_0,p_1}$ denote the noise-graph mechanism. Then, we define the parallel protection of the dynamic graph with parameters p_0 and p_1 as the protection process that provides $\tilde{G} = \tilde{G}_0, \tilde{G}_1, \ldots, \tilde{G}_T$ with $\tilde{G}_i = \mathcal{A}_{p_0,p_1}(G_i)$ for $i = 0, \ldots, T$. We denote the parallel protection of a dynamic graph G with parameters p_0 and p_1 as $\mathcal{A}^{||}_{p_0,p_1}(G)$.

Experiment and Results I

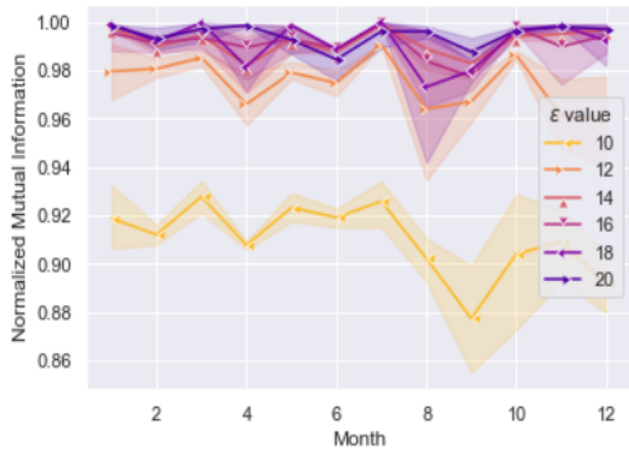
Table 1: Preprocessed datasets statistics

Dataset	No. of nodes	No. of Edges	Avg. Snapshot Density
CAIDA-AS	5,715	403,761	0.0010
DBLP	25,439	450,878	0.00007

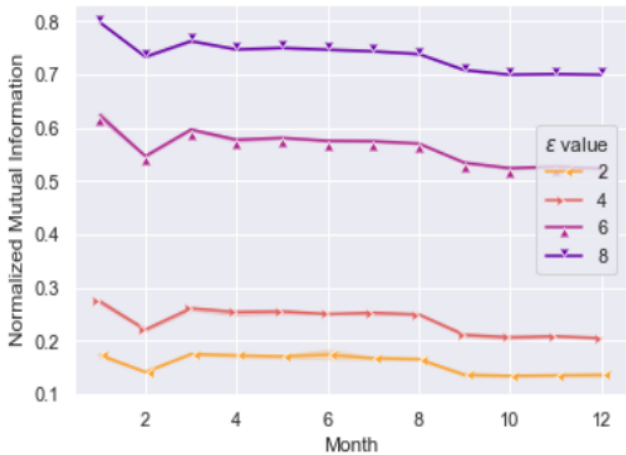
Experiment and Results (NMI vs Month for CAIDA-AS data)



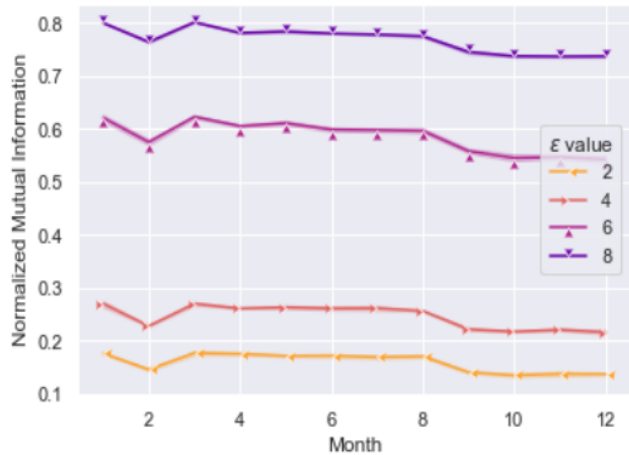
(a) Dynamic mechanism for large ε values



(b) Parallel mechanism for large ε values

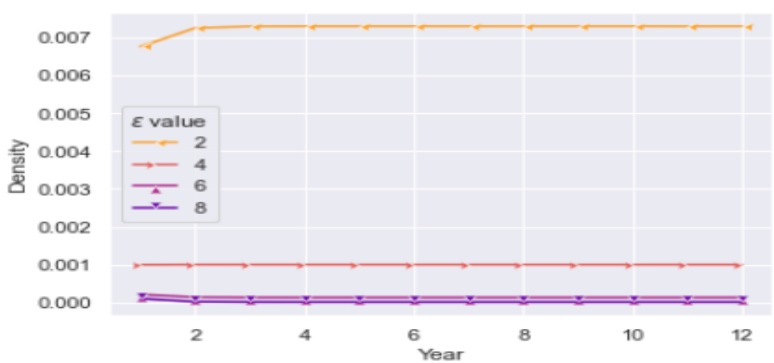


(c) Dynamic mechanism for small ε values

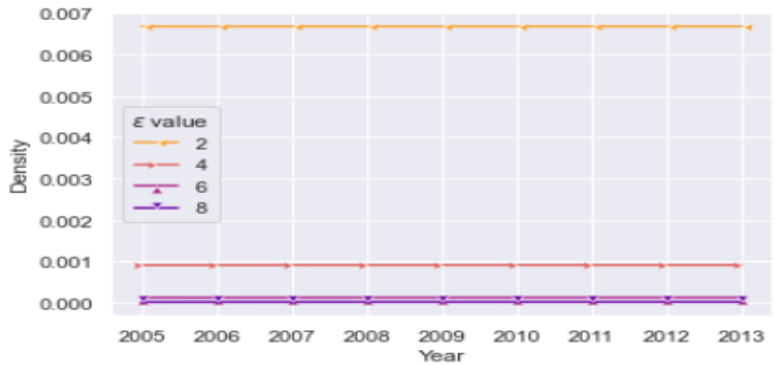


(d) Parallel mechanism for small ε values

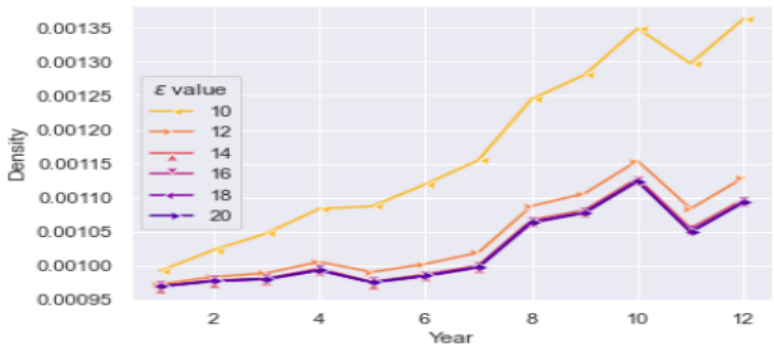
Experiment and Results III



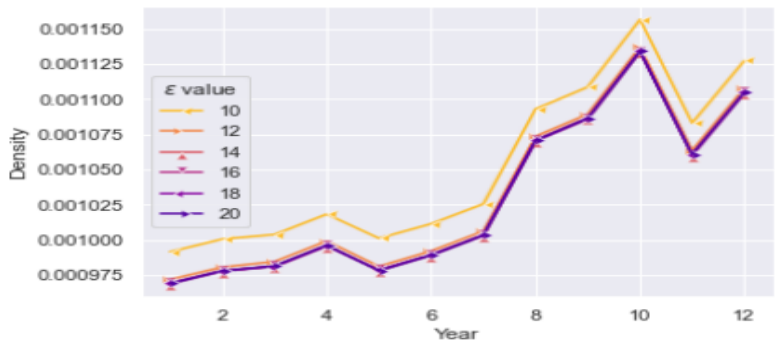
(a) Dynamic mechanism for small ε values



(b) Parallel mechanism for small ε values



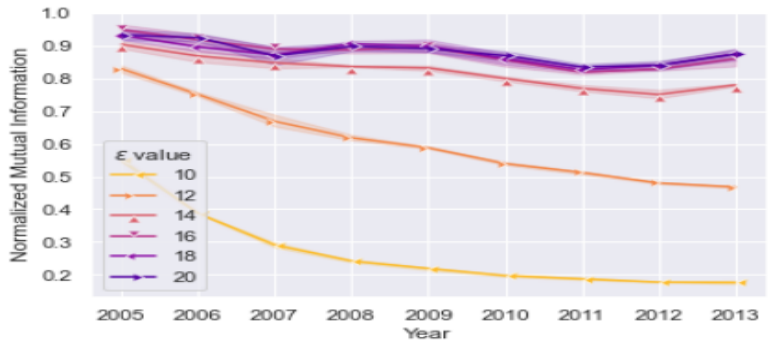
(c) Dynamic mechanism for large ε values



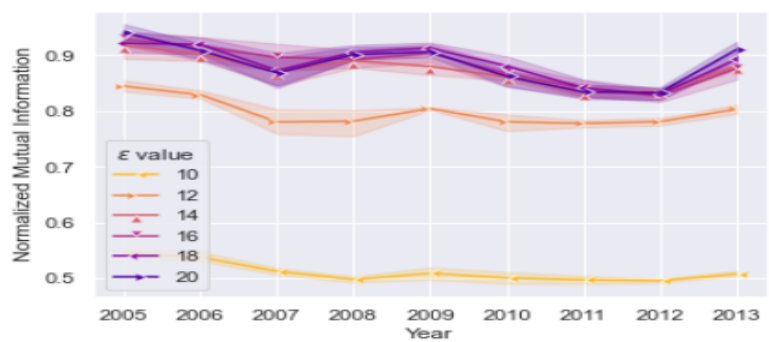
(d) Parallel mechanism for large ε values

Fig. 2: Densities for the snapshot-graphs obtained by applying the dynamic and parallel mechanisms to CAIDA-AS.

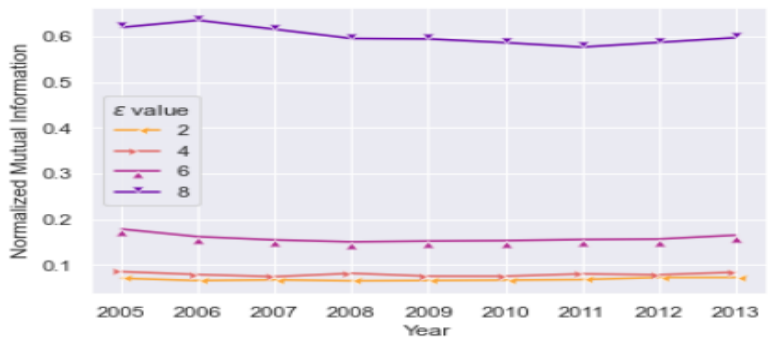
Experiment and Results IV



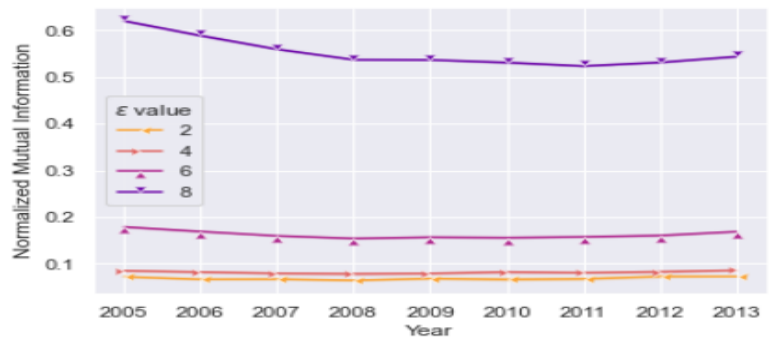
(a) Dynamic mechanism for large ε values



(b) Parallel mechanism for large ε values



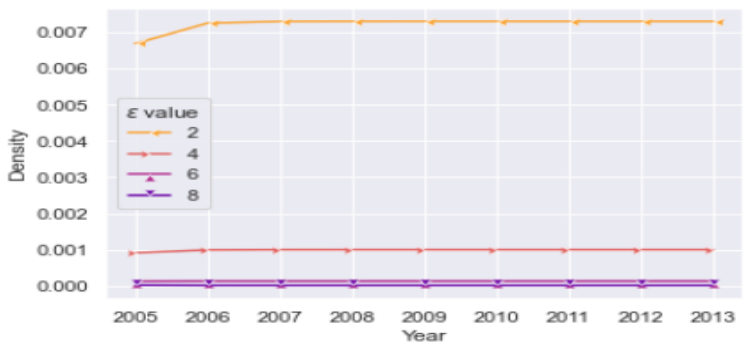
(c) Dynamic mechanism for small ε values



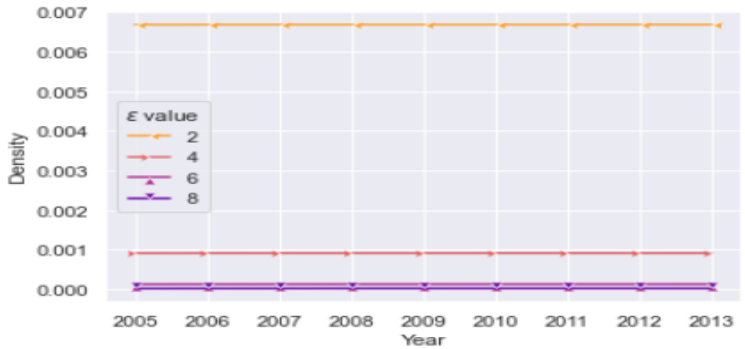
(d) Parallel mechanism for small ε values

Fig. 3: Normalized mutual information between the communities detected on the DBLP data and the data protected with the dynamic and parallel mechanisms for several ε values.

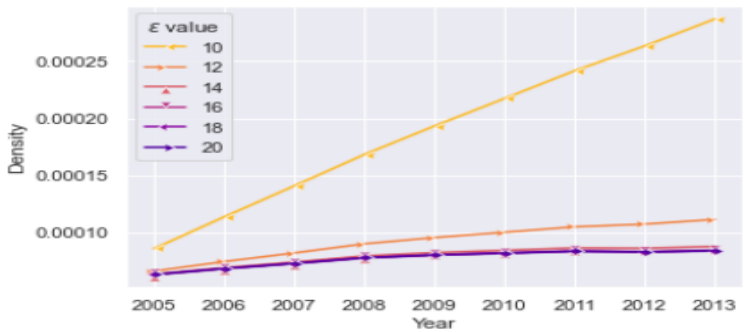
Experiment and Results V



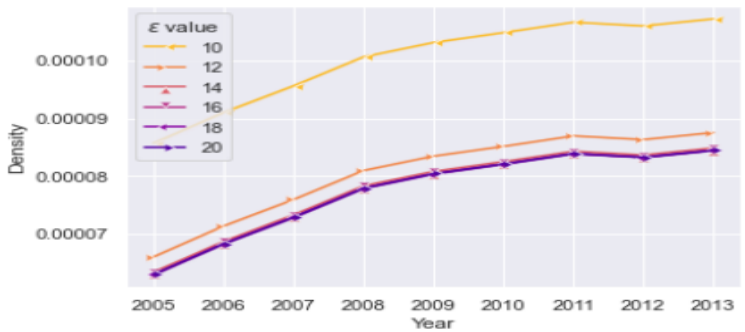
(a) Dynamic mechanism for small ε values



(b) Parallel mechanism for small ε values



(c) Dynamic mechanism for large ε values



(d) Parallel mechanism for large ε values

Fig. 4: Densities for the snapshot-graphs obtained by applying the dynamic and parallel mechanisms to DBLP.

Thank You!!