UVA | BIOCOMPLEXITY INSTITUTE

Efficient and Scalable Network Science with Privacy

Dung Nguyen

11/3/23, Virginia Commonwealth University

Privacy

- The growth of huge datasets (both private and publicly)
- The need to keep sensitive information private
- Privacy attacks aim to predict sensitive (hidden) information
  - There are many types of attacks (with different targets, settings, assumptions, resources)
  - Also there are many privacy models

Different techniques for privacy, especially graph data (very partial list)

- Naive ID Removal
- Edge Editing (EE) based techniques[1]
- k-anonymity based techniques[2]
- Aggregation/Class/Cluster based techniques[3]
- Random Walk (RW) based schemes[4]
- Differential Privacy (DP) based techniques[5]

[1] Ying and Wu 2013

[2] Zhou and Pei 2008, Liu and Terzi 2008, Zou et al. 2009, Cheng et al. 2010

[3] Hay et al. 2008, Bhagat et al. 2009, Thompson and Yao 2009

[4] Mittal et al. 2013, Liu et al. 2016

[5] Sala et al. 2011, Proserio et al. 2012, Wang and Wu 2013, Xiao et al. 2014

Membership attack

- Attackers want to determine if an individual is in a dataset
- Risks of being in a dataset (or studies, surveys)
  - A case-control study shows that smoking causes cancer
  - What are the risks of a participant P? What happends to his insurance premium if P is a smoker?
  - Not in the study: very low chance P is a smoker
  - In the case-control study: chance P is a smoker is $1/2$

Attribute Inference attack

- Attribute Inference attack in Online Social Networks [6]
- Given an online social network that contains both private and public information
- Attackers can infer some private information (e.g., cities where people live) from the public information

[6] Gong and Lu 2018

Risks: De-anonymity

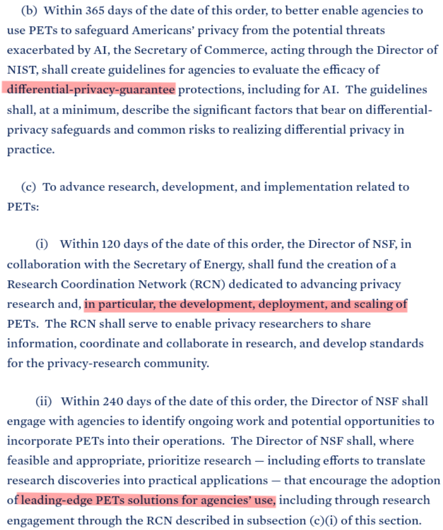
- Massachussetts personal medical reports [7]
  - anonymized medical reports
  - publicly available voter registration records
  - matching based on few data fields: DOB, race, gender, Zipcode

[7] Sweeney 1997

Risks: Netflix linkage attack

- Large-scale de-anonymization attack [8]
- Netflix subscribers's anonymized viewing histories
  - Only rating and dates
  - Data are sampled (about 1/10)
  - Unspecified perturbation
- The Internet Movie Database (IMDb), where user rates movies
- De-anonymization is successful even with
  - Some data perturbation
  - Back-ground knowledge not be precise
    - Attacker may have some guesses about the target, such as approximated rating, approximated dates

[8] Narayanan and Shmatikov 2008



(b) Within 365 days of the date of this order, to better enable agencies to use PETs to safeguard Americans' privacy from the potential threats exacerbated by AI, the Secretary of Commerce, acting through the Director of NIST, shall create guidelines for agencies to evaluate the efficacy of differential-privacy-guarantee protections, including for AI. The guidelines shall, at a minimum, describe the significant factors that bear on differential-privacy safeguards and common risks to realizing differential privacy in practice.

(c) To advance research, development, and implementation related to PETs:

(i) Within 120 days of the date of this order, the Director of NSF, in collaboration with the Secretary of Energy, shall fund the creation of a Research Coordination Network (RCN) dedicated to advancing privacy research and, in particular, the development, deployment, and scaling of PETs. The RCN shall serve to enable privacy researchers to share information, coordinate and collaborate in research, and develop standards for the privacy-research community.

(ii) Within 240 days of the date of this order, the Director of NSF shall engage with agencies to identify ongoing work and potential opportunities to incorporate PETs into their operations. The Director of NSF shall, where feasible and appropriate, prioritize research — including efforts to translate research discoveries into practical applications — that encourage the adoption of leading-edge PETs solutions for agencies' use, including through research engagement through the RCN described in subsection (c)(i) of this section.

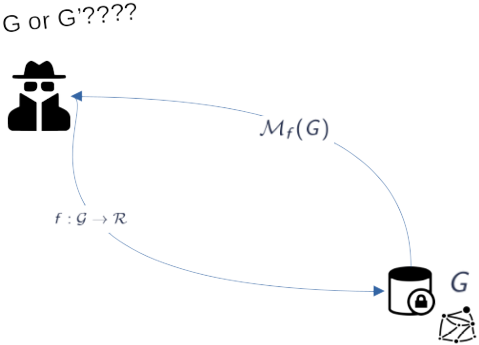
Introduction to Differential Privacy

- Intuitively, differential privacy guarantees that a randomized algorithm behaves similarly on similar input databases
- **Definition**: A randomized algorithm $M : \mathcal{D} \rightarrow R$ is (ϵ, δ)-differentially private:
  - For all subsets $O \subset \mathcal{R}(M)$
  - For all pairs of neighbors $x, x' \in \mathcal{D}$: $x \sim x'$,
  - $\Pr[M(x) \in O] \leq e^{\epsilon} \Pr[M(x') \in O] + \delta$
- Privacy promises
  - Protect against membership attack
    - Protect the absence and presence of an individual
    - With correct setup, DP is effective against membership attack [9]
- Strong privacy degrades utility
  - In some problems, DP algorithms have trivial utility bounds

[9] Gong and Lu 2018

Settings of differential privacy model

- A trusted curator, that protects a private dataset
- The curator takes some query f
- The curator calculates on the private dataset, and returns a ”privatized” output for the query
- Attacker may know “something” about the dataset



Graph

- A graph consists of 2 sets: nodes (a.k.a. vertices) and edges
- An edge connects a pair of nodes
- Edges can be directed or undirected.
  - Generally speaking, “graph” implies an undirected graph
- Nodes and edges may have labels (weights)
  - “weighted graph”

Differential Privacy models in graph

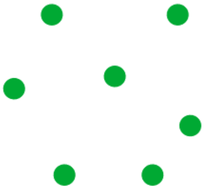
- Differential Privacy concept relies on neighborhood definition
  For any pair of neighbor graphs $G \sim G'$:

$$\Pr[M(G) \in \mathcal{O}] \leq e^{\epsilon} \Pr[M(G') \in \mathcal{O}] + \delta$$

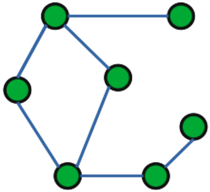
- Edge-privacy: 2 neighbor graphs differ by 1 edge
- Node-privacy: 2 neighbor graphs differ by 1 node and its adjacent edges
- In general, node-privacy is much harder
- Other models: node-label-privacy, edge-label-privacy, etc

Edge-privacy model and what it protects

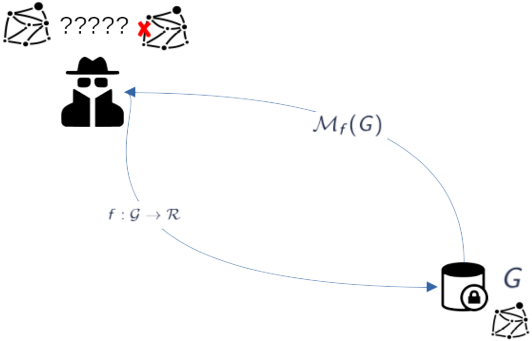
Public Information



Private Data

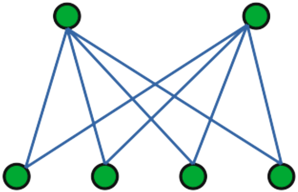


Edge-privacy model and what it protects



Toy problem: Triangle counting in the Edge-privacy model

Question: How an edge may change the count of triangles?



Example 1: Attacker's perspective

- In this example, the attacker know something about the hidden data
- The attacker want to know if there exists a connection between node u and v
- They know that u and v have n common friends (and the common friends don't know each other)
    - This setting is extreme
- They suspect that the hidden data may be either G or G' (on the previous slide)
- They initiate the attack by sending "triangle count" query to the data curator
- They observe the response to determine if the hidden dataset is G or G'

Example 1a: Naive data curator's response (No privacy)

- Exact count of triangles
- Attacker expect to receive 0 or n
- Easy to expose connection between u and v

Example 1b: Curator DP-response

- By DP properties, then for any response $o : \Pr[M(G) = o] \leq e^{\epsilon} \Pr[M(G') = o]$
- Like before, the attacker expect if they receive 0 or n
- They will analyze the probabilities of G and G' conditioned on the response they get
- The attacker receive response 0:

$$\frac{\Pr[0 \text{ is observed} | \text{Input is } G]}{\Pr[0 \text{ is observed} | \text{Input is } G']} \leq e^{\epsilon}$$

- By Bayes rule

$$\frac{\Pr[\text{Input is } G | 0 \text{ is observed}]}{\Pr[\text{Input is } G' | 0 \text{ is observed}]} \leq e^{\epsilon}$$

- The same is true for n
- Conclusion: With small ϵ, they cannot differentiate G and G'

Example 1c: Curator DP-response

- The attacker are smart, they analyze some output ranges
  - Because getting a low triangle count may favor the possibility that the input is G
- For example, they define $S = [0..\log n)$ i.e. low count range
- The attacker receive response $o \in S$ (we call S is observed):

$$\frac{\Pr[\text{Input is } G | S \text{ is observed}]}{\Pr[\text{Input is } G' | S \text{ is observed}]} \leq e^{\epsilon}$$

- Conclusion: They cannot differentiate G and G', no matter what they observes

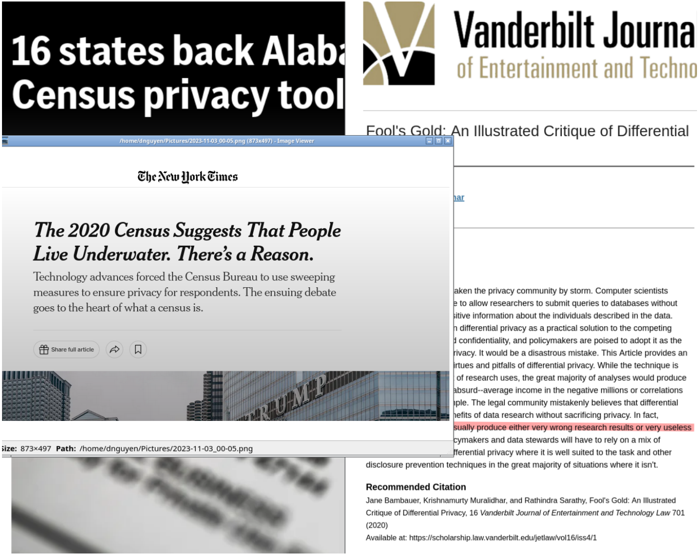
Examples of sensitive networks

- For edge-privacy model, we assume that nodes are public
- Nodes are humans, edges indicate close physical interactions
- Bipartite graph: nodes are (1) mobile transmitters and (2) mobile devices. An edge indicates a device comes to a transmitter
- Nodes are airports, edges indicate private flights (during some timeframes)

Objective

Can we solve graph mining problems under differential privacy guarantee with high accuracy?

- In some problems, DP algorithms must incur trivial utility bounds
  - E.g., Vertex cover
- No universal way to effectively transform a non-private algorithm to differentially private variant
  - Except for simple statistics with low, well-defined sensitivities



16 states back Alab
Census privacy tool

/home/dnguyen/Pictures/2023-11-03_00-05.png (873x497) - Image Viewer

The New York Times

The 2020 Census Suggests That People Live Underwater. There's a Reason.

Technology advances forced the Census Bureau to use sweeping measures to ensure privacy for respondents. The ensuing debate goes to the heart of what a census is.

ize: 873×497 Path: /home/dnguyen/Pictures/2023-11-03_00-05.png

Vanderbilt Journa
of Entertainment and Techno

Fool's Gold: An Illustrated Critique of Differential

har

aken the privacy community by storm. Computer scientists
e to allow researchers to submit queries to databases without
sitive information about the individuals described in the data.
h differential privacy as a practical solution to the competing
d confidentiality, and policymakers are poised to adopt it as the
rivacy. It would be a disastrous mistake. This Article provides an
irtues and pitfalls of differential privacy. While the technique is
of research uses, the great majority of analyses would produce
absurd--average income in the negative millions or correlations
ple. The legal community mistakenly believes that differential
hefits of data research without sacrificing privacy. In fact,
sually produce either very wrong research results or very useless
cymakers and data stewards will have to rely on a mix of
fferential privacy where it is well suited to the task and other
disclosure prevention techniques in the great majority of situations where it isn't.

Recommended Citation

Jane Bambauer, Krishnamurty Muralidhar, and Rathindra Sarathy, Fool's Gold: An Illustrated Critique of Differential Privacy, 16 *Vanderbilt Journal of Entertainment and Technology Law* 701 (2020)

Available at: https://scholarship.law.vanderbilt.edu/jetlaw/vol16/iss4/1

Laplace mechanism

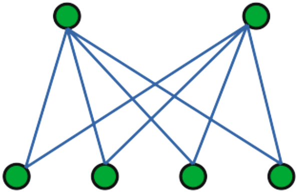
- The Laplace mechanism: Output a noisy response
  - For a function $f : \mathcal{D} \rightarrow \mathbb{R}^k$
  - Sensitivity $\Delta_f = max_{x \sim x'} ||f(x) - f(x')||_1$
  - Adding a Laplacian noise to the output of $f(x)$ guarantees ϵ-DP:
    - $M_f(x) = f(x) + Lap(\Delta_f / \epsilon)$
  - Applications:
    - When output space are real scalars, vectors
    - Statistical queries, with finite Δ

Triangle counting with Laplace mechanism

- Recall about Laplace machanism $M_f(G) = f(G) + Lap(\frac{\Delta}{\epsilon})$
- Recall about global sensitivity $\Delta = max_{G \sim G'} \|f(G) - f(G')\|_1$

Global sensitivity of triangle count

Question: How an edge may change the count of triangles?



- Applying them: $M_f(G) = f(G) + Lap(\frac{n}{\epsilon})$
- Utility guarantee: $\Pr[|M_f(G) - f(G)| > \frac{n \log n}{\epsilon}] < \frac{1}{n}$
- Can we do better?
  - Most realistic graphs are sparse
  - It's rare that a node is connected with $\Theta(n)$ other nodes

Alternative method: Smooth sensitivity

- **Definition:** [10]

$$S^*_{f,\beta}(G) = \max_{k=0,1,\dots,\binom{n}{2}} e^{-k\beta}(\max_{G':d(G,G')=k} LS_f(G'))$$

- $\max_{G':d(G,G')=k} LS_f(G')$ is "local sensitivity at distance k"
- **Noise calibration**
  - Laplacian noise (simplified): $M_f(G) = f(G) + Lap(2S^*_{f,\beta}(G)/\epsilon)$
    - (ϵ, δ)-differential privacy
  - Cauchy noise (simplified): $M_f(G) = f(G) + Cauchy(6S_{f,\beta}(G)/\epsilon)$
    - ϵ-differential privacy

[10] Nissim et al. 2011

Smooth sensitivity for triangle counting

- **Local sensitivity at distance k of triangle count:**

$$\max_{i \neq j, i, j \in [n]} \min \left(a_{ij} + \frac{k + \min(k, b_{ij})}{2}, n - 2 \right)$$

  - a_{ij}: number of common neighbors of nodes i and j
  - b_{ij}: number of half-built triangles involved i and j
- Computing a_{ij}: Using matrix multiplication AA^T

Accuracy comparison

- Global sensitivity: n
- Smooth sensitivity: $np^2 \log n$ for a random graph $\mathcal{G}(n, p)$
- Why do we favor "Smooth sensitivity"?
  - Real life networks usually have small p : $p = O(1/\sqrt{n})$

Approximated Smooth Sensitivity (our ongoing work)

- $\tilde{S}$: (γ, δ')-upper approximation of the Smooth Sensitivity S^* if:
- For any graph G, with probability at least $1 - \delta'$:
  - $S^*(G) \leq \tilde{S}(G) \leq e^{\gamma} S^*(G)$
- **Noise calibration**
  - Laplacian noise (simplified): $M_f(G) = f(G) + Lap(2\tilde{S}(G)/\epsilon)$
- Guarantee $(\epsilon, \delta + 2\delta')$-differential privacy

Theoretical Guarantees

	Noise Magnitude	Runtime
Global Sensitivity	$\Theta(n)$	$O(1)$
Ladder function	$O(np \log n)$	$O(\text{matrix mul. of size } n)$
Recursive mechanism	$O(np \log n)$	$O(mn)$
Restricted Sensitivity	$\Theta(\max_v deg^2(v))$	$O(mn)$
Blackbox Transformation	$\Theta(n \times \#_\Delta)$	$O(\text{non-private } \#_\Delta \text{ alg.})$
(Exact) Smooth Sensitivity	$O(np \log n)$	$O(\text{matrix mul. of size } n)$
Our method	$O(np \log n)$	$O(m \log n + n)$

Table: Summary of the characteristics of differentially private #triangle counting algorithms in the edge-privacy model. The magnitudes of noise reported on random graphs with uniform edge probability p.

Experimental Results

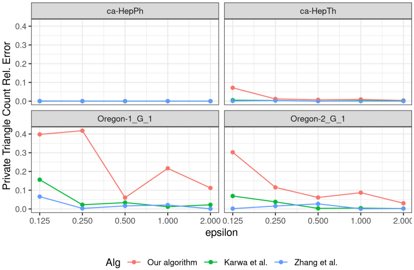


Figure: Triangle Count Relative Error, showing the accuracy of the private triangle count with noise calibrated by Our algorithm and the baseline methods (Exact Smooth & Ladder function)

$$\textsc{Relative Error} = \frac{|M_{f_\Delta}(G) - f_\Delta(G)|}{f_\Delta(G)}$$

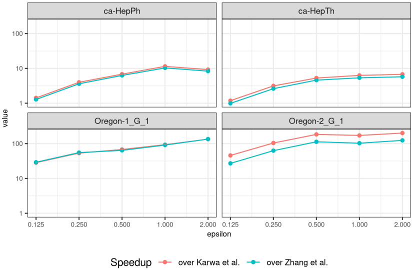


Figure: Speedup of Our algorithm (compared to the baselines) on selected networks.

How to approximate S^*

- **Approximate Local sensitivity at distance k of triangle count:**

$$\max_{i \neq j, i, j \in [n]} \min\left(\hat{a}_{ij} + \frac{k + \min(k, b_{ij})}{2}, n - 2\right)$$

  - $\hat{a}_{ij}$: **Approximated** number of common neighbors of nodes i and j
  - b_{ij}: number of half-built triangles involved i and j

- Computing $\hat{a}_{ij}$: Using “Diamond Sampling”

Diamond Sampling

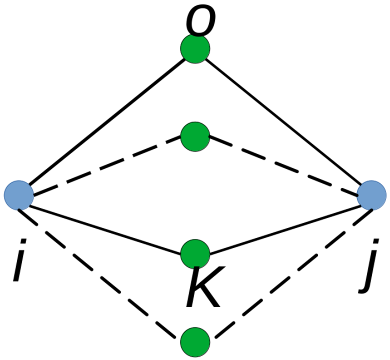


Figure: Examples of diamonds involve (i,j) in a graph. Green nodes are common neighbors of i, j.

- Observation: $\#diamonds$ involves $(i,j) \propto a_{ij}^2$
- Sample edge $(i,k) \propto deg(i) \times deg(k)$
- Sample o adjacent to i
- Sample j adjacent to k
- Check if (i,k,j,o) forms a diamond
- After s iters: count x_{ij}-the number of diamonds involves i,j
- Parallelization?

Convergence of diamond sampling

- When the number of iteration $s >$ Some threshold:
- With desired error probability β and error margin θ

$$\Pr\left[\left|\frac{x_{ij}\text{Something}}{s} - a_{ij}^2\right| > \theta a_{ij}^2\right] \leq \beta. \tag{1}$$

- Problem? Some threshold $= \frac{\text{Something}}{a_{ij}^2}$
- Solution:
  - Guess an upper bound τ that approximate $\max_{ij} a_{ij}^2$
  - Run the diamond sampling
  - Check if τ is close enough with our count $\max_{ij} x_{ij}$
  - Reduce τ if not and repeat the above 2 steps

Applications of Approximate Smooth Sensitivity

- Significantly improve running time
  - Several orders of magnitude speedup for triangle counting with DP
- Extensible for any query with known Smooth Sensitivity function
  - Unfortunately, there are not so many at the moment
- Possibility of faster DP algorithms on graphs
- Future work: Other types of subgraph counting, other sampling methods, etc

Thank you and Q/A.