Differential Privacy in Social Network Analysis

Tanmay Bakshi[1,✉]

[1]Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India

In this techno-savvy world, privacy is a myth. There are growing concerns about privacy of social network data, which hinder meaningful growth. Studies have shown how an analytical treatment of social graph networks can yield interesting trends, ultimately leading to revenue generation, societal upliftment, and technological advancement. However, a naive treatment by simply anonymizing the data and releasing it to public and private entities can pose an imminent threat to the anonymity of the entity participating in the social network. In the following report, I provide a concise yet seminal summary of this concern of privacy over social networks, a powerful guarantee known as "Differential Privacy", and discuss the vulnerabilities of the most prevalant form of social data models in today's times, namely Graph Neural Networks. Finally, some mechanisms to enforce plausible deniability (DP) over these GNNs via means of Aggregation Perturbation are discussed. This has opened up new avenues for scaling present-era social networks, and allow for next-gen innovation for the future.

ϵ-Differential Privacy | Graphs | Graph Neural Networks | Aggregation Perturbation

Correspondence: ***t.bakshi@cs.iitr.ac.in***

Introduction

DIFFERENTIAL PRIVACY describes a promise made by the data holder to a data subject, saying "You will not be affected by allowing your data to be used in any study or analysis, no matter the variety of datasets or information available with the data hosting entity." A medical school might create a database that consists of health records of the residents of a particular state or province. The students perform certain queries on the data and observe that a vast majority of cancer patients have historically been exposed to the practice of smoking, then the students might infer that smoking causes cancer. Has the data generating entity – namely, the smoker been harmed in this process? Perhaps yes, and no – if an insurance firm infers the subject's identity, he might see an increase in his insurance premiums. On the other hand, if a charity organization infers his identity, he might be brought into a quit-smoking program.

Along similar lines, operators of online social networks are increasingly disbursing potentially sensitive information about users and their relationships – the so-called "social network data". This data is being delivered to various suitors like advertisers, developers and researchers. Privacy is claimed to be protected by anonymizing potentially sensitive data – i.e., names and other demographic information that may identify a single data subject, often modeled as the node, is suppressed. While this may offer privacy, yet it does not guarantee anonymity. These two terms have been equivocally interchanged in several high-profile cases of data sharing (Netflix, Google, etc.)

Differential privacy aims to address this paradox of learning nothing about an individual while learning useful information about a population, to form the basis of making inferences. It is not an algorithm, but an agreement or a promise that has evolved into a privacy standard for use on tabular data that provides strong guarantees of privacy without making assumptions about an attacker's background knowledge. Going back to the smokers' example – after the inference of "smoking is bad", can it be said that the smoker's privacy has been compromised? Differential privacy will take the view that it was not, with the mathematically sound rationale that the impact on the smoker is the same , independent of whether or not he was in the study. Specifically, it ensures that any sequence of outputs, in response to queries put forth by the adversary, is "essentially" equally likely to occur, independent of the presence or absence of any individual.

The paper introduces the concept of differential privacy in a systemic and analytical manner, by building upon traditional differential privacy. Then, it delves into the study of known standards of differential privacy for network data, and also outlines a third standard, out-link privacy. However, there was a lack of implementational detail of any kind in the paper, hence I chose to focus on the concept of ensuring differential privacy in social networks, which, by design of convenience, are facilitated by Graph Neural Networks.

II. PROMISE OF TRADITIONAL DIFFERENTIAL PRIVACY

Differential Privacy was developed by Cynthia Dwork (1) at Microsoft Research Labs, which provided a mathematical guarantee of privacy that certain queries can satisfy.

(A). Model of Computation. (2)

We assume the existence of a trusted and trustworthy curator, who happens to hold the data of individuals in a database D, typically comprised of n rows. The intuition is that each row contains the data of a single individual, the data subject, and then, intuitively, the privacy goal is to simultaneously protect every individual row while permitting statistical analysis of the database as a whole.

Let I be the set of individuals who contribute to the data-set D_I. We use $\mathcal{F} : \mathcal{D} \rightarrow \mathbb{R}^k$ to refer to the desired non-privatized analysis performed on a data-set and $\mathcal{Q} : \mathcal{D} \rightarrow \mathbb{R}^k$ to refer to the privatized implementation of $\mathcal{F}$. We refer to the publicly released, privatized analysis results as $\mathcal{R}$. We define D_I to be the true world, from which analysis was taken. Secondly, "neighbouring worlds" to this world are

defined, each differing in the presence/absence of a particular data subject in the survey.

(B). Definitions and Terminology.

(B).1. Graph Neural Networks. The fundamental bases on which we have built the concept of Differential Privacy remains to be Social Networks. In this current techno-savvy world, social networks have evolved from small, distributed, deterministic records to massive, heterogeneous, integrated graphs. Graph Neural Networks are the new norm that allow for inference studies and surveys to take place on such graphs. Graph Neural Networks (GNNs) have emerged as a powerful tool for learning from graph-structured data, and their popularity has surged due to their ability to achieve impressive performance in a wide range of applications, including and not limited to social network analysis, drug discovery, particle physics, and traffic prediction (3), (4).
GNNs have established the state-of-the-art at learning from structural connectivity of graphs by iteratively updating node embeddings via the mechanism of message passing and transformations.
These message passing mechanisms between adjacent nodes, spanning across global graphs have a considerable privacy risk due to the privacy-sensitive nature of user data. Direct publication of the model parameters ω^* will violate ϵ-Differential Privacy, since it reveals information about the underlying "training" data. One might try to address this issue by adding Laplacian noise to ω^* but it is rather challenging given the complex correlation between D and ω^*. Recent studies have shown that various attacks, such as link stealing, membership inference, and node attribute inference can successfully intrude the privacy of these graph datasets (**?**).

(B).2. ϵ-Differential Privacy. A randomized query,

$$Q : \mathcal{D} \to \mathbb{R}^k$$

satisfies ϵ-Differential Privacy, if, for any two possible neighboring datasets D_1, D_2, and any possible query result R:

$$\frac{Pr[Q(D_1) = R]}{Pr[Q(D_2) = R]} \leq e^{\epsilon}$$

Here ϵ is a small, positive value that controls the trade-off between privacy and accuracy/explainability, and is often set by the curator. Setting a smaller value of ϵ provides high degree of privacy, but as a result of the obfuscation due to added noise, the inference capability decreases.

III. PROBLEM DEFINITION AND CHALLENGE

Say, we are given a graph **G = (V, E)** be a directed, unweighted graph with a set of nodes $V = \{v_1, \ldots, v_N\}$ and edges E represented by an adjacency matrix $A \in \{0,1\}^{N \times N}$. Node features are represented by a matrix $X \in \mathbb{R}^{N \times d}$, where X_i denotes the d-dimensional feature vector of node v_i. Nodes are also populated with different kinds of labels, using a one-hot encoding denoted by $Y \in \{0,1\}^{N \times C}$, where C is the number of classes, and Y_i is a one-hot vector indicating the label of node v_i.
We work in a transductive setting, where, in the graph, a certain set of nodes have node labels, and the task is to predict labels for the other nodes in the graph.
Now, a node-classification model (GNN-based) can be formalized by a parameter set Θ. This model takes in the learned node embeddings **X** and the adjacency matrix **A**, and outputs the corresponding predicted labels as

$$\hat{\mathbf{Y}} = \mathbf{F}(X, A; \boldsymbol{\Theta})$$

This equation above represents the Inference Stage of the model, where the structural information of the graph is passed, along with the learned node embedding. So, even if we obfuscate the node embedding, there is still the risk of inference attacks possible here.
Secondly, during the training phase, our objective will be to minimize a standard loss function, (Cross-entropy loss) as

$$\Theta^* = \arg\min_{\Theta} \sum_{v \in V} \ell_1(\hat{Y_v} - Y_v)$$

During this training phase, elaborate layers of message passing and aggregation between nodes of a graph are at work. Hence, after **K** iterations, each node has some knowledge about its updated, aware, **K**-hop neighborhood (and potentially the entire graph). Hence, with certain queries to the model, it is possible to extract compromising information about participating entities. (5).
Hence, the goal is to preserve the privacy of graph datasets for both the training, as well as the inference phase using efficient techniques. Theoretically, it can be conclusively shown that, just like in the case of tabular data, the definition of DP can be adapted to graph datasets. Intuitively, along the lines of "neighboring worlds" interpretation, we have "adjacent-graphs".

Definition : (Edge-level adjacent graphs). Two graphs, G and G_0 are edge-level adjacent, if one can be obtained by removing a single edge from the other. Therefore, G and G_0 differ by at most one edge.

PROPOSITION : ProGAP

While surveying probable candidates to ensure Edge-Level and Node-Level Differential Privacy in Graph datasets, the techniques proposed in ProGAP(6) and GAP(7) were found to be highly effective in many respects. Aggregation Perturbation technique (7) is used to enforce and ensure differential privacy during the training phase, as it adds stochastic noise to the output of the aggregation step, in proportion to its sensitivity.

Privacy Mechanism. The aggregation step is effectively summing up/pooling messages received from the 1-hop

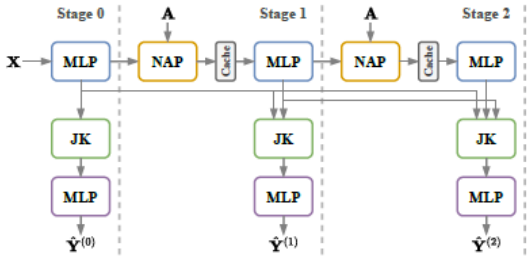

Figure 1: An example ProGAP architecture with three stages (depth = 2). MLP and JK represent multi-layer perceptron and Jumping Knowledge [49] modules, respectively. NAP denotes the normalize-aggregate-perturb module used to ensure the privacy of the adjacency matrix, with its output cached immediately after computation to save privacy budget. Training is done progressively, starting with the first stage and then expanding to the second and third stages, each using its own head MLP. The final prediction is obtained by the head MLP of the last stage.

Fig. 1. NAP mechanism

neighborhood of the node in consideration. Hence, perturbing this function with noise can be seen as an attempt to hide the presence of a single edge (or more, for node-level privacy.) This can potentially ensure DP at both training and inference stages. By obfuscating each individual Aggregation step, this method can be generalized to arbitrary number of hops. Since there is no constraint, as such, over each node's embedding, the conventional, non-aggregation step can cause the resulting embedding to become highly sensitive towards a dominant node that influences the embedding. Thus, if we try to add Gaussian Noise to such a case, a significant amount of the privacy budget shall be extinguished, leading to low learnability.

Hence, a separate Aggregation module has been implemented, called **Normalize-Aggregate-Perturb (NAP)** mechanism.

Model Architecture and NAP Mechanism. The NAP module is defined as:

$$\mathrm{NAP}\left(\mathrm{A}, \mathrm{X}; \sigma\right) = \left[\sum_{j=1}^{N} \frac{\mathrm{X}_j}{\|\mathrm{X}_j\|_2} \mathrm{A}_{j,i} + \mathcal{N}(0, \sigma^2 \mathrm{I}_d) \mid \forall i \in \{1, \ldots, N\}\right], \tag{7}$$

Fig. 2. NAP mechanism

This module first normalizes each of the relevant embedding that is to be aggregated, performs the aggregation, as well as perturbs the embedding by adding the pre-computed Gaussian Noise. It is possible to show that the resulting model provides edge-level DP at the training level, as every query to the adjacency matrix is immediately perturbed with noise. The issue here is that, as the depth of the GNN increases, the privacy budget will increase exponentially, as more and more aggregations will be performed spanning across the entire graph.

Subverting the Issue of Privacy Cost. To subvert the issue of exponentially increasing privacy cost over K-hops, the paper uses a progressive training technique, done over K+1 stages. First, a shallow, 1-hop submodel is trained, embeddings computed and perturbed. Then, these embeddings are cached, and progressively used to train the next submodel of 2-hops and so on, till K-hops are covered. So basically, the outputs of the NAP at the end of each query are cached and reused in further queries. This ensures more expressivity, reduced privacy cost and leads to better performance.

EXPERIMENTS

Datasets Used. The papers that describe ProGAP and GAP architectures use a variety of social network datasets to establish the concreteness of their concept.

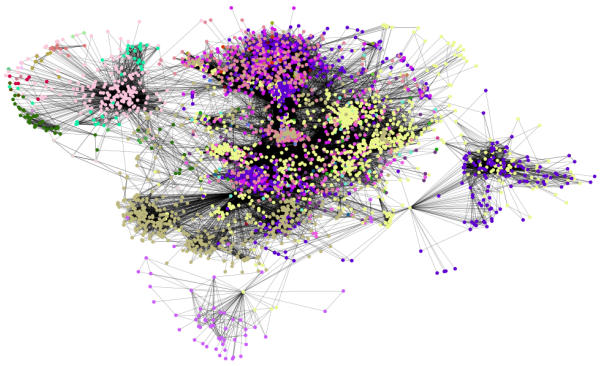

Fig. 3. NetworkX Plot of the Community Distribution of the Facebook Dataset using Spring Layout

Facebook : A collection of anonymized social network profiles of students from a U.S. University. The edges depict friendships and the task is to predict the student's class year.

FB-100 : It is an extension of the Facebook dataset, but contains the social network of a 100 U.S. Universities taken together.

Reddit : Contains a set of Reddit posts as nodes, where edges represent if the same user commented on both posts, and the goal is to predict the post's subreddit on which it was posted.

Apart from this, other datasets were also used, consisting of a large number of nodes, and hence resulting in a large graph structure in the transductive setting.

Baselines and Implementation. The GraphSAGE (8) model, which is one of the most popular GNN architectures was used in the ProGAP and GAP papers for non-private performance comparison. Some pre-processing and post-processing layers were included as well to suit the task at hand.

Apart from this, other baselines were also used in the papers, along with the orchestrated GAP and ProGAP methods.

The data contained pre-loaded train, test and validation masks, as required for the transductive setting (75-15-10%

Table 2: Comparison of Experimental Results (Mean Accuracy ± 95% CI)

Privacy Level	Method	ϵ	Facebook	Reddit	Amazon	FB
Non-Private	GraphSAGE [16]	∞	**84.7 ± 0.09**	99.4 ± 0.01	93.2 ± 0.07	74.0
	GAP [41]	∞	80.5 ± 0.42	**99.5 ± 0.01**	92.0 ± 0.10	66.4
	ProGAP (Ours)	∞	84.5 ± 0.24	99.3 ± 0.03	**93.3 ± 0.04**	**74.4**
Edge-Level Private	MLP	0.0	50.8 ± 0.20	82.5 ± 0.08	71.1 ± 0.18	34.9
	EdgeRand [46]	1.0	50.2 ± 0.50	82.8 ± 0.05	72.7 ± 0.1	34.9
	GAP [41]	1.0	69.4 ± 0.39	97.5 ± 0.06	78.8 ± 0.26	46.5
	ProGAP (Ours)	1.0	**77.2 ± 0.33**	**97.8 ± 0.05**	**84.2 ± 0.07**	**56.9**

Fig. 4. Result Table drawn from (6)

respectively). We vary ϵ within $\{0.1, 0.5, 1, 2, 5, 10\}$ for the edge-level privacy ($\epsilon = \infty$ corresponds to the non-private setting).

The GNN depth, number of MLP_{base} layers for feature-set reduction, the MLP_{head} layers for classification, along with the batch-size, optimizers, type of aggregation, and other parameters are all tunable, and the results can be compared by setting different configurations for each training. It is particularly hard to present a comprehensive analysis of all this, in lieu of large size of the dataset, huge computational power required and limited resources available.

Results and Inference

A wide plethora of exploratory and detailed analysis have been presented in the papers (7) (6), pertaining to the effects of model depth, parameters, etc. However, for the scope of my original paper which talks of accuracy-privacy trade-off, the analysis of the aforesaid is provided below.

The level of non-privacy corresponds to $\epsilon = \infty$, and edge-privacy corresponds to $\epsilon = 1$. The node classification accuracy is a representative of the inference capability of the perturbed model. In case where no perturbation was done (non-private case), both GAP and ProGAP models perform comparable to GraphSAGE model, which might be achieving higher performance due to attention modules. Note, that the GraphSAGE model is essentially non-private in all of the assessments. For the edge-level private cases, in Facebook, Reddit, and FB-100 datasets (among others) GAP and ProGAP perform extremely better than the baseline established by the simple MLP model that doesn't even take the graph structural information into account.

Epilogue : Future Scope

There are several scopes to develop upon this research, some of which was explored by me

As mentioned, it is indeed very difficult to obfuscate highly connected nodes, as due to their heavy influence on the graph structure as a whole. Their embeddings influence other embeddings, and in order to allow plausible deniability for the participation of such nodes, a high privacy budget is spent. This decreases the accuracy as well. Hence, if we are able to pre-process the graph data structure, and construct a histogram plot for the node centralities (betweenness/eigenvector etc.), we can think of a suitable histogram

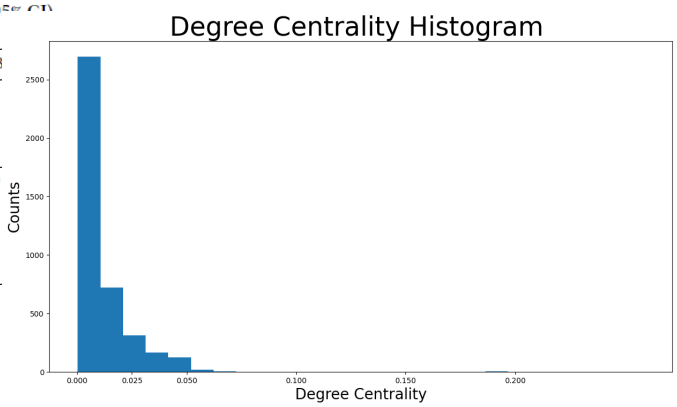

Fig. 5. Degree Centrality Histogram of a subset of Facebook dataset

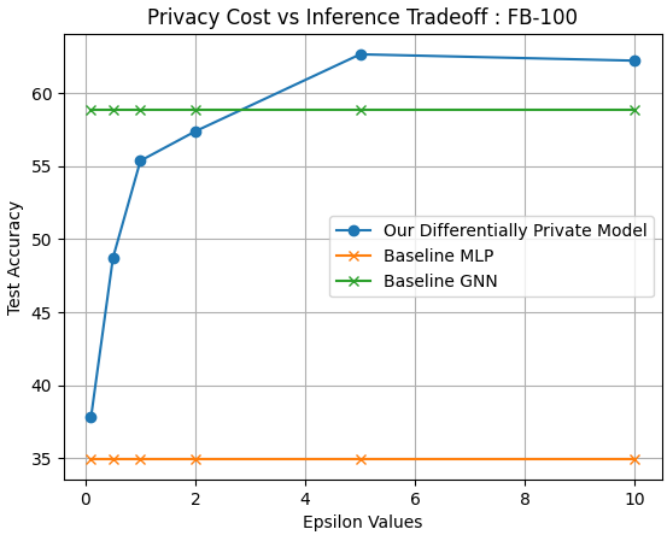

Fig. 6. Inference vs Privacy Trade-off for FB-100 Dataset

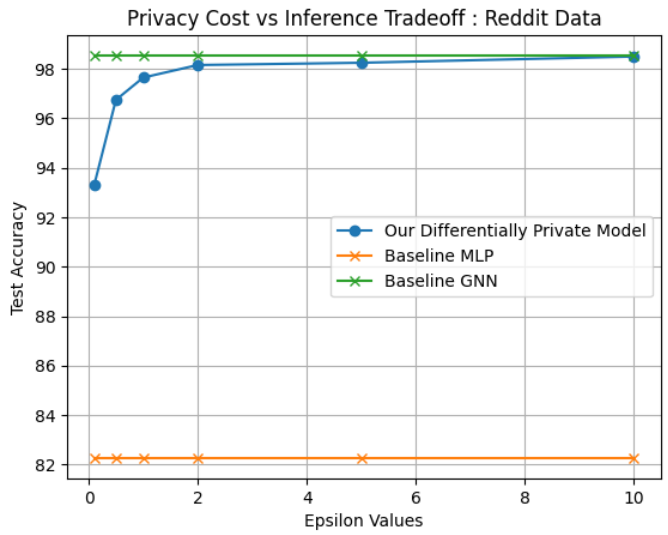

Fig. 7. Inference vs Privacy Trade-off for Reddit Dataset

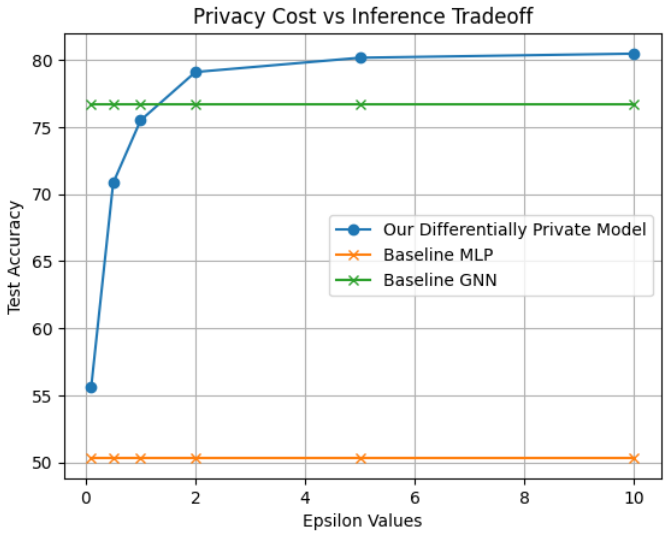

Fig. 8. Inference vs Privacy Trade-off for Facebook Dataset

partitioning algorithm to separately deal with the cases of nodes with high connectivity - implementing a noise perturbation function that can be called by these nodes in each iteration, or separately defining another approach to handle their embeddings can work as well.

Conclusion

In this era of digi-socio civilities, social networks encompass across lives, and differential privacy provides an intuitive yet powerful tool for analysing social networks. The concept of plausible deniability is made available to individual entrants of datasets, and in powerful abstractions like graph data structures, this deniability is being delivered via innovative techniques like Aggregation-Perturbation.

Neural Networks, which are especially vulnerable to attacks, despite of anonymizing individual data entries or weights, can be made differentially private by considering definitions of edge-adjacent graphs and node-adjacent graphs and obfuscating one, or more than one edges respectively to achieve plausible deniability. As is evident, the inference ability for such a distribution does not suffer at the expense of increased privacy - courtesies of innovative training paradigms such as ProGAP.

Bibliography

1. Cynthia Dwork. Differential privacy. pages 338–340, 2011. doi: 10.1007/978-1-4419-5906-5_752.
2. Christine Task and Chris Clifton. A guide to differential privacy theory in social network analysis. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ASONAM '12, page 411–417, USA, 2012. IEEE Computer Society. ISBN 9780769547992. doi: 10.1109/ASONAM.2012.73.
3. Savannah Thais, Paolo Calafiura, Grigorios Chachamis, Gage DeZoort, Javier Duarte, Sanmay Ganguly, Michael Kagan, Daniel Murnane, Mark S. Neubauer, and Kazuhiro Terao. Graph neural networks in particle physics: Implementations, innovations, and challenges, 2022.
4. Thomas Gaudelet, Ben Day, Arian R. Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy B. R. Hayter, Richard Vickers, Charles Roberts, Jian Tang, David Roblin, Tom L. Blundell, Michael M. Bronstein, and Jake P. Taylor-King. Utilising graph machine learning within drug discovery and development, 2021.
5. Yi Zhang, Yuying Zhao, Zhaoqing Li, Xueqi Cheng, Yu Wang, Olivera Kotevska, Philip S. Yu, and Tyler Derr. A survey on privacy in graph neural networks: Attacks, preservation, and applications, 2023.
6. Sina Sajadmanesh and Daniel Gatica-Perez. Progap: Progressive graph neural networks with differential privacy guarantees, 2023.
7. Sina Sajadmanesh, Ali Shahin Shamsabadi, Aurélien Bellet, and Daniel Gatica-Perez. Gap: Differentially private graph neural networks with aggregation perturbation, 2022.
8. William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.